RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2023.048

A sixth-order finite difference method for solving the generalized Burgers-Fisher and generalized Burgers-Huxley equations

Sheng-en Liu, Yongbin Ge*, Fang Tian

Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan 750021 China

*Corresponding author, e-mail: gybnxu@yeah.net

Received 10 Sep 2022, Accepted 29 Apr 2023
Available online 12 Aug 2023

ABSTRACT: The generalized Burgers-Huxley and generalized Burgers-Fisher equations are solved by using a new sixth-order finite difference method. Such equations are discretized in space by a five-point sixth-order finite difference scheme combined with a truncation error modification technique and in time by the sixth-order backward difference formula, which constitutes a finite difference scheme with the sixth-order accuracy in both space and time. Then, the Crank-Nicolson method combined with the Richardson extrapolation technique is employed to obtain the approximate solutions at the starting time steps which can keep the spatio-temporal sixth-order accuracy of the main scheme. The linear system formed by this scheme at each time step is efficiently solved by the Thomas algorithm. Finally, some numerical experiments are carried out to verify the accuracy and reliability of the proposed method.

KEYWORDS: generalized Burgers-Huxley equation, generalized Burgers-Fisher equation, sixth-order finite difference scheme, Thomas algorithm, Richardson extrapolation

MSC2020: 35K55

INTRODUCTION

Nonlinear partial differential equations are used to describe an enormous variety of phenomena in science and engineering. The generalized Burgers-Huxley equation (GBHE) [1] and the generalized Burgers-Fisher equation (GBFE) [2] are typical examples of such equations, and they have been used, for example, to describe the interaction between diffusion transport, convective effects, and reaction mechanisms [3].

In this article, we consider the following one-dimensional (1D) nonlinear partial differential equation:

$$\frac{\partial u}{\partial t}+\mu u^s\frac{\partial u}{\partial x}=\varepsilon\frac{\partial^2 u}{\partial x^2}+f(u),\quad (x,t)\in\Omega\times I,\tag{1}$$

with the initial condition

$$u(x,T_0)=\psi(x),\quad x\in\Omega,\tag{2}$$

and boundary conditions

$$u(a,t)=g(t),\ u(b,t)=l(t),\quad t\in I,\tag{3}$$

where $\Omega=[a,b]$ is the spatial domain and $I=(T_0,T]$ is the temporal interval. ε ($\varepsilon>0$) is the diffusion coefficient. Eq. (1) is called the GBHE if $f(u)=\rho(1-u^s)(u^s-\theta)$ and the GBFE if $f(u)=\rho(1-u^s)$, where $\mu,\ \rho,\ \theta,\ s\ (s>0)$ are parameters. In particular, when $\rho=0$ and $s=1$, Eq. (1) becomes the Burgers equation, which describes the propagation and reflection of waves in nonlinear dissipative systems [4]. u and $f(u)$ are sufficiently smooth functions which are required to have higher-order derivatives.

Many numerical approaches have been proposed to solve these equations. For instance, Sari et al [1, 5] proposed a series of high-order difference schemes for solving the GBHE and GBFE. Hammad and El-Azab [2] proposed a series of difference schemes combined with the collocation method which can achieve a $2N$-order ($N=2,3,4,\dots$) accuracy in space for solving the GBHE and GBFE. Zhang et al [6] obtained numerical solutions of the GBFE and GBHE by using the local discontinuous Galerkin method. Duan et al [7] developed a lattice Boltzmann model to solve the GBHE. Mittal and Tripathi [8] developed a scheme based on a collocation of modified cubic B-spline functions in space and the SSP-RK54 method in time for solving the GBHE and GBFE. Singh et al [9] solved the GBFE using a fourth-order B-spline collocation method. Ray and Gupta [10] used a variational iteration method for solving the GBHE. Zhang and Ge [11] proposed a fully implicit fourth-order scheme for solving 1D nonlinear convection-diffusion-reaction equations and also applied this scheme to solve the GBHE. Kumar and Ray [12] obtained numerical solutions of the GBHE and GBFE using the discontinuous Legendre wavelet Galerkin method. Mohanty and Sharma [13] proposed a high-resolution method based on off-step non-polynomial spline approximations to obtain numerical solutions of the GBFE. Sun and Zhu [14] obtained numerical solutions of the GBHE using the cubic B-spline quasi-interpolation method. Zibari et al [15] and Namjoo et al [16] used a nonstandard finite difference method to obtain numerical solutions of the Burgers-Huxley equation and the GBFE, respectively.

As can be seen, many high-precision finite difference methods have been developed to solve the GBHE and GBFE [1, 2, 5, 11]. It should be noted that a common characteristic of these difference schemes is that they involve the computation of the spatial first-, second-, and even higher-order derivatives, which inevitably increases both computational complexity and the storage requirement. In addition, with regard to discretization of the time derivative, in [2], only the low-order backward difference formulae were adopted to result in the low-order time accuracy; while in [1, 5], the explicit Runge-Kutta method was employed to present a conditionally stable difference scheme; Zhang and Ge [11] used the fourth-order backward difference formula to provide the fourth-order accuracy in time. Therefore, in this paper, we try to develop a novel sixth-order finite difference scheme to solve the GBHE and GBFE. The merit of this scheme is that it does not need to calculate any derivative, and when the time step is proportional to the space step, this scheme can obtain high-precision calculation results and reach the sixth-order accuracy in both space and time.

In this new approach, the spatial derivatives at the interior points of Eq. (1) are discretized by a new five-point sixth-order difference scheme, and the spatial derivatives at the neighbouring boundary points are discretized by a three-point fourth-order difference scheme. The time derivative is discretized by the sixth-order backward difference method, and Crank-Nicolson scheme combined with Richardson extrapolation technique are used to provide the approximate solutions at the starting time steps. It should be noted that after Gear [17] introduced the sixth-order difference method in his monograph in 1971, the application of such a method was very rare. Using the above methods, we will propose a spatio-temporal sixth-order accurate difference method to numerically solve the GBFE and GBHE.

SIXTH-ORDER FINITE DIFFERENCE SCHEME

In this section, we first introduce some notations. We divide the domain $[a,b]\times(T_0,T]$ into an $N\times M$ uniform mesh, with spatial step length $h=(b-a)/N$ and temporal step length $\tau=(T-T_0)/M$. u_i^n stands for the numerical solution at grid point (x_i,t_n), $x_i=a+ih$, $i=0,1,\dots,N$, $t_n=T_0+n\tau$, $n=0,1,\dots,M$.

First, the Taylor series of u_{i+1} and u_{i-1} at x_i are

$$u_{i\pm1}=u_i\pm h\left(\frac{\partial u}{\partial x}\right)_i+\frac{h^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_i\pm\frac{h^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_i+\frac{h^4}{24}\left(\frac{\partial^4 u}{\partial x^4}\right)_i\pm\frac{h^5}{120}\left(\frac{\partial^5 u}{\partial x^5}\right)_i+\frac{h^6}{720}\left(\frac{\partial^6 u}{\partial x^6}\right)_i\pm\frac{h^7}{5040}\left(\frac{\partial^7 u}{\partial x^7}\right)_i+O(h^8). \quad (4)$$

Similarly, the Taylor series of u_{i+2} and u_{i-2} at x_i are

$$u_{i\pm2}=u_i\pm 2h\left(\frac{\partial u}{\partial x}\right)_i+2h^2\left(\frac{\partial^2 u}{\partial x^2}\right)_i\pm\frac{8h^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_i+\frac{16h^4}{24}\left(\frac{\partial^4 u}{\partial x^4}\right)_i\pm\frac{32h^5}{120}\left(\frac{\partial^5 u}{\partial x^5}\right)_i+\frac{64h^6}{720}\left(\frac{\partial^6 u}{\partial x^6}\right)_i\pm\frac{128h^7}{5040}\left(\frac{\partial^7 u}{\partial x^7}\right)_i+O(h^8). \quad (5)$$

Summing the two equalities in (4) gives

$$u_{i+1}+u_{i-1}=2u_i+h^2\left(\frac{\partial^2 u}{\partial x^2}\right)_i+\frac{h^4}{12}\left(\frac{\partial^4 u}{\partial x^4}\right)_i+\frac{h^6}{360}\left(\frac{\partial^6 u}{\partial x^6}\right)_i+O(h^8). \quad (6)$$

Similarly, from (5), we obtain

$$u_{i+2}+u_{i-2}=2u_i+4h^2\left(\frac{\partial^2 u}{\partial x^2}\right)_i+\frac{4h^4}{3}\left(\frac{\partial^4 u}{\partial x^4}\right)_i+\frac{64h^6}{360}\left(\frac{\partial^6 u}{\partial x^6}\right)_i+O(h^8). \quad (7)$$

We multiply (6) by 64, subtract the result from (7), and, after some rearrangement, obtain

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i=\frac{1}{60h^2}[64(u_{i+1}+u_{i-1})-(u_{i+2}+u_{i-2})-126u_i]-\frac{h^2}{15}\left(\frac{\partial^4 u}{\partial x^4}\right)_i+O(h^6). \quad (8)$$

We introduce the notation $\tilde{D}_x^2 u_i=\frac{1}{60h^2}[64(u_{i+1}+u_{i-1})-(u_{i+2}+u_{i-2})-126u_i]$. It is obvious that

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i=\tilde{D}_x^2 u_i-\frac{h^2}{15}\left(\frac{\partial^4 u}{\partial x^4}\right)_i+O(h^6). \quad (9)$$

Next, subtracting the two equalities in (4), we obtain

$$u_{i+1}-u_{i-1}=2h\left(\frac{\partial u}{\partial x}\right)_i+\frac{h^3}{3}\left(\frac{\partial^3 u}{\partial x^3}\right)_i+\frac{h^5}{60}\left(\frac{\partial^5 u}{\partial x^5}\right)_i+\frac{h^7}{2520}\left(\frac{\partial^7 u}{\partial x^7}\right)_i+O(h^8). \quad (10)$$

Similarly, from (5), we have

$$u_{i+2}-u_{i-2}=4h\left(\frac{\partial u}{\partial x}\right)_i+\frac{8h^3}{3}\left(\frac{\partial^3 u}{\partial x^3}\right)_i+\frac{8h^5}{15}\left(\frac{\partial^5 u}{\partial x^5}\right)_i+\frac{16h^7}{315}\left(\frac{\partial^7 u}{\partial x^7}\right)_i+O(h^8). \quad (11)$$

We multiply (10) by 32, subtract the result from (11), and, after some rearrangement, obtain

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{1}{60h}[32(u_{i+1}-u_{i-1})-(u_{i+2}-u_{i-2})] - \frac{2h^2}{15}\left(\frac{\partial^3 u}{\partial x^3}\right)_i + O(h^6). \tag{12}$$

We introduce the notation $\tilde{D}_x u_i = \frac{1}{60h}[32(u_{i+1}-u_{i-1})-(u_{i+2}-u_{i-2})]$. Then, (12) can be rewritten as

$$\left(\frac{\partial u}{\partial x}\right)_i = \tilde{D}_x u_i - \frac{2h^2}{15}\left(\frac{\partial^3 u}{\partial x^3}\right)_i + O(h^6). \tag{13}$$

It is obvious that if the $\partial^4 u/\partial x^4$ term in (9) and the $\partial^3 u/\partial x^3$ term in (13) can be treated with a fourth-order approximation, we can obtain a sixth-order approximation for the second- and first-order derivatives. This is the principle underlying the construction of the sixth-order difference scheme in this paper. To do this, we consider the 1D steady equation that is transformed from Eq. (1):

$$-\varepsilon\frac{\partial^2 u}{\partial x^2} + p\frac{\partial u}{\partial x} = S, \tag{14}$$

in which $S = f - \partial u/\partial t$ and $p = \mu u^s$. We discretize the first- and second-order derivatives in Eq. (14) using (13) and (9), respectively, to obtain

$$-\varepsilon\tilde{D}_x^2 u_i + p_i\tilde{D}_x u_i - Q_i = S_i, \tag{15}$$

where Q_i is the truncation error at node x_i given by

$$Q_i = \frac{h^2}{15}\left(2p\frac{\partial^3 u}{\partial x^3} - \varepsilon\frac{\partial^4 u}{\partial x^4}\right)_i + O(h^6). \tag{16}$$

We take the derivative of both sides of Eq. (14) and then get, respectively,

$$\frac{\partial^3 u}{\partial x^3} = \frac{1}{\varepsilon}\left(p\frac{\partial^2 u}{\partial x^2} + \frac{\partial p}{\partial x}\frac{\partial u}{\partial x} - \frac{\partial S}{\partial x}\right), \tag{17}$$

$$\frac{\partial^4 u}{\partial x^4} = \frac{1}{\varepsilon}\left(p\frac{\partial^3 u}{\partial x^3} + 2\frac{\partial p}{\partial x}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 p}{\partial x^2}\frac{\partial u}{\partial x} - \frac{\partial^2 S}{\partial x^2}\right). \tag{18}$$

Substituting (17) and (18) into (16), we obtain the new truncation error expression

$$Q_i = \frac{h^2}{15}\left(\frac{p^2}{\varepsilon}\frac{\partial^2 u}{\partial x^2} + \frac{p}{\varepsilon}\frac{\partial p}{\partial x}\frac{\partial u}{\partial x} - 2\frac{\partial p}{\partial x}\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 p}{\partial x^2}\frac{\partial u}{\partial x} - \frac{p}{\varepsilon}\frac{\partial S}{\partial x} + \frac{\partial^2 S}{\partial x^2}\right)_i + O(h^6). \tag{19}$$

To obtain the sixth-order approximation of (19), it is necessary to adopt the fourth-order approximation of the first- and second-order derivatives, for which we adopt the following formulas [18, 19]:

$$\bar{D}_x u_i = \frac{8u_{i+1} - 8u_{i-1} - u_{i+2} + u_{i-2}}{12h}, \tag{20}$$

$$\bar{D}_x^2 u_i = \frac{16u_{i+1} + 16u_{i-1} - u_{i+2} - u_{i-2} - 30u_i}{12h^2}. \tag{21}$$

We can then discretize (19) as

$$Q_i = \frac{h^2}{15}\left(\frac{p_i^2}{\varepsilon}\bar{D}_x^2 u_i + \frac{p_i}{\varepsilon}\bar{D}_x p_i\bar{D}_x u_i - 2\bar{D}_x p_i\bar{D}_x^2 u_i - \bar{D}_x^2 p_i\bar{D}_x u_i - \frac{p_i}{\varepsilon}\bar{D}_x S_i + \bar{D}_x^2 S_i\right) + O(h^6). \tag{22}$$

Substituting (22) into (15), we obtain an approximation of sixth-order accuracy in space

$$-\varepsilon\tilde{D}_x^2 u_i + p_i\tilde{D}_x u_i - \frac{h^2}{15\varepsilon}(p_i^2\bar{D}_x^2 u_i + p_i\bar{D}_x p_i\bar{D}_x u_i - 2\varepsilon\bar{D}_x p_i\bar{D}_x^2 u_i - \varepsilon\bar{D}_x^2 p_i\bar{D}_x u_i) = S_i + \frac{h^2}{15}\bar{D}_x^2 S_i - \frac{h^2 p_i}{15\varepsilon}\bar{D}_x S_i + O(h^6), \quad i = 2, 3, \cdots, N{-}2. \tag{23}$$

As shown in Fig. 1, Eq. (23) is used as a spatial difference scheme for discrete points ($i = 2, 3, \dots, N{-}2$). For the spatial discretization, since the five-point template is used, the neighbouring boundary points are needed to be processed separately.

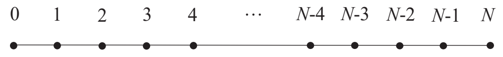

Fig. 1 Spatial discrete diagram.

At present, there are two commonly used methods for dealing with neighbouring boundary points. One is to establish two ghost points outside the computational domain [20], and the other is to use a lower-order scheme with small templates [21]. The latter is adopted in this paper. Considering the neighbouring boundary points ($i = 1,\ N{-}1$) of Eq. (14), we adopt the same derivation process of the above sixth-order scheme to obtain a three-point fourth-order compact difference scheme

$$-\varepsilon D_x^2 u_i + p_i D_x u_i - \frac{h^2}{12\varepsilon}(p_i^2 D_x^2 u_i + p_i D_x p_i D_x u_i - 2\varepsilon D_x p_i D_x^2 u_i - \varepsilon D_x^2 p_i D_x u_i) = S_i + \frac{h^2}{12}D_x^2 S_i - \frac{h^2 p_i}{12\varepsilon}D_x S_i + O(h^4), \quad i = 1, N-1, \tag{24}$$

where $D_x u_i = \frac{u_{i+1}-u_{i-1}}{2h}$ and $D_x^2 u_i = \frac{u_{i+1}-2u_i+u_{i-1}}{h^2}$. The schemes (23)–(24) form the spatial difference scheme of Eq. (14) as follows

$$\begin{cases} -\varepsilon\tilde{D}_x^2u_i+p_i\tilde{D}_xu_i-\frac{h^2}{15\varepsilon}(p_i^2\bar{D}_x^2u_i+p_i\bar{D}_xp_i\bar{D}_xu_i \\ -2\varepsilon\bar{D}_xp_i\bar{D}_x^2u_i-\varepsilon\bar{D}_x^2p_i\bar{D}_xu_i)=S_i+\frac{h^2}{15}\bar{D}_x^2S_i \\ -\frac{h^2p_i}{15\varepsilon}\bar{D}_xS_i+O(h^6), \qquad i=2,3,\cdots,N-2, \\ -\varepsilon D_x^2u_i+p_iD_xu_i-\frac{h^2}{12\varepsilon}(p_i^2D_x^2u_i+p_iD_xp_iD_xu_i \\ -2\varepsilon D_xp_iD_x^2u_i-\varepsilon D_x^2p_iD_xu_i)=S_i+\frac{h^2}{12}D_x^2S_i \\ -\frac{h^2p_i}{12\varepsilon}D_xS_i+O(h^4), \qquad i=1,N-1. \end{cases} \tag{25}$$

Remark 1 Scheme (25) adopts a three-point fourth-order scheme with lower precision than interior points for processing neighbouring boundary points, which can achieve the sixth-order accuracy in space. In [21, 22], it has been discussed and proved that the local low-order accuracy would not affect the global high-order accuracy.

To facilitate the derivation, the scheme (25) will be written in the following form, and the overall difference scheme of the space is denoted by $\mathscr{A}_xu_i=\mathscr{L}_xS_i$ as follows

$$\begin{cases} A_x^6u_i=L_x^6S_i, \quad i=2,3,\cdots,N-2 \\ A_x^4u_i=L_x^4S_i, \quad i=1,N-1 \end{cases}$$

$$\Longrightarrow \mathscr{A}_xu_i=\mathscr{L}_xS_i, \quad i=1,2,\cdots,N-1. \tag{26}$$

where

$$\begin{aligned} A_x^6&=-\varepsilon\tilde{D}_x^2+p_i\tilde{D}_x-\frac{h^2}{15\varepsilon}(p_i^2\bar{D}_x^2+p_i\bar{D}_xp_i\bar{D}_x \\ &\quad-2\varepsilon\bar{D}_xp_i\bar{D}_x^2-\varepsilon\bar{D}_x^2p_i\bar{D}_x), \\ L_x^6&=1+\frac{h^2}{15}(\bar{D}_x^2-\frac{p_i}{\varepsilon}\bar{D}_x), \\ A_x^4&=-\varepsilon D_x^2+p_iD_x-\frac{h^2}{12\varepsilon}(p_i^2D_x^2+p_iD_xp_iD_x \\ &\quad-2\varepsilon\bar{D}_xp_iD_x^2-\varepsilon D_x^2p_iD_x), \\ L_x^4&=1+\frac{h^2}{12}(D_x^2-\frac{p_i}{\varepsilon}D_x). \end{aligned}$$

In Appendix A, we show the coefficients of the spatial difference scheme.

Next, we discretize the temporal variable. Substituting $S=f-\partial u/\partial t$ into (26), we obtain

$$\mathscr{A}_xu_i=\mathscr{L}_x\Big(f-\frac{\partial u}{\partial t}\Big)_i. \tag{27}$$

We consider the value of (27) at the n-th time step

$$\mathscr{L}_x\left(\frac{\partial u}{\partial t}\right)_i^n=\mathscr{L}_xf_i^n-\mathscr{A}_xu_i^n. \tag{28}$$

The temporal derivative $(\partial u/\partial t)^n$ is discretized with the sixth-order backward difference method [17]

$$\begin{aligned} \left(\frac{\partial u}{\partial t}\right)_i^n=\frac{49}{20\tau}u_i^n-\frac{6}{\tau}u_i^{n-1}+\frac{15}{2\tau}u_i^{n-2}-\frac{20}{3\tau}u_i^{n-3} \\ +\frac{15}{4\tau}u_i^{n-4}-\frac{6}{5\tau}u_i^{n-5}+\frac{1}{6\tau}u_i^{n-6}. \end{aligned} \tag{29}$$

Then, (29) is substituted into (28) to obtain

$$\begin{aligned} \mathscr{L}_x\frac{49}{20}u_i^n+\tau\mathscr{A}_xu_i^n=\mathscr{L}_x(6u_i^{n-1}-\frac{15}{2}u_i^{n-2}+\frac{20}{3}u_i^{n-3} \\ -\frac{15}{4}u_i^{n-4}+\frac{6}{5}u_i^{n-5}-\frac{1}{6}u_i^{n-6})_{i,j}+\tau\mathscr{L}_xf_i^n, \\ i=1,2,\cdots N-1, \quad n=6,7,\cdots,M. \end{aligned} \tag{30}$$

Eq. (30) is a difference scheme with the sixth-order accuracy in both time and space. Since (30) is a seven-step scheme, besides the n-th and initial time step, we also need a startup scheme to calculate the approximation solutions at the other five steps. Next, we will construct a scheme for solving these five startup steps.

We consider the value of (27) at the $(n-\frac{1}{2})$-th time step

$$\mathscr{L}_x\left(\frac{\partial u}{\partial t}\right)_i^{n-\frac{1}{2}}=\mathscr{L}_xf_i^{n-\frac{1}{2}}-\mathscr{A}_xu_i^{n-\frac{1}{2}}. \tag{31}$$

The temporal derivative $(\partial u/\partial t)_i^{n-\frac{1}{2}}$ in (31) is discretized with the Crank–Nicolson method

$$\left(\frac{\partial u}{\partial t}\right)_i^{n-\frac{1}{2}}=\frac{u_i^n-u_i^{n-1}}{\tau}+O(\tau^2), \tag{32}$$

and the other terms in (31) are taken as weighted averages of the respective values at the $(n-1)$-th and n-th time steps, i.e.,

$$f_i^{n-\frac{1}{2}}=\frac{f_i^n+f_i^{n-1}}{2}+O(\tau^2), u_i^{n-\frac{1}{2}}=\frac{u_i^n+u_i^{n-1}}{2}+O(\tau^2). \tag{33}$$

Substituting (32) and (33) into (31), we get the following startup step scheme

$$\begin{aligned} \Big(\mathscr{L}_x+\frac{\tau}{2}\mathscr{A}_x\Big)u_i^n=\Big(\mathscr{L}_x-\frac{\tau}{2}\mathscr{A}_x\Big)u_i^{n-1}+\frac{\tau}{2}\mathscr{L}_x(f_i^n+f_i^{n-1}), \\ i=1,2,\cdots,N-1, \quad n=1,2,3,4,5, \end{aligned} \tag{34}$$

which can achieve the sixth-order accuracy in space, but only has second-order accuracy in time. In order to be consistent with the scheme (30) with the sixth-order accuracy in time, we use the Richardson extrapolation formula [23]

$$u_i^n(h,\tau)=\frac{64}{45}u_i^{4n}\Big(h,\frac{\tau}{4}\Big)-\frac{4}{9}u_i^{2n}\Big(h,\frac{\tau}{2}\Big)+\frac{1}{45}u_i^n(h,\tau). \tag{35}$$

to improve the time accuracy from second-order to sixth-order. Combining (30) with (34), the difference scheme with the sixth-order accuracy in time and space is obtained as follows

$$\begin{cases} \left(\mathscr{L}_x+\frac{\tau}{2}\mathscr{A}_x\right)u_i^n=\left(\mathscr{L}_x-\frac{\tau}{2}\mathscr{A}_x\right)u_i^{n-1}+\frac{\tau}{2}\mathscr{L}_x(f_i^n+f_i^{n-1}), \\ \qquad i=1,2,\cdots,N-1, \quad n=1,2,3,4,5. \\ \left(\mathscr{L}_x\frac{49}{20}+\tau\mathscr{A}_x\right)u_i^n=\mathscr{L}_x(6u_i^{n-1}-\frac{15}{2}u_i^{n-2}+\frac{20}{3}u_i^{n-3} \\ \qquad -\frac{15}{4}u_i^{n-4}+\frac{6}{5}u_i^{n-5}-\frac{1}{6}u_i^{n-6})+\tau\mathscr{L}_x f_i^n, \\ \qquad i=1,2,\cdots,N-1, \quad n=6,7,\cdots,M. \end{cases} \tag{36}$$

Scheme (36) is a seven-step implicit difference scheme for solving the Eqs. (1)–(3). From the derivation process, it can achieve the sixth-order accuracy in both time and space.

NUMERICAL ALGORITHM

In order to make the proposed scheme (36) clear, we present the practical implementation of such a time-stepping method as Algorithm 1, namely

Algorithm 1

```
Step 1: Give the initial function values $u_i^0=\psi_i$ by (2);
Step 2: Do $n=1,2,\ldots,5$
            Give the boundary function values $u_0^n=g^n$, $u_N^n=l^n$ by (3);
            Set arbitrary initial values $u_i^{n,(0)}$;
            do $l=1,2,\ldots$
              (1) Set $p(u_i^{n,(l)})\leftarrow p(u_i^{n,(l-1)})$, $f(u_i^{n,(l)})\leftarrow f(u_i^{n,(l-1)})$;
              (2) Compute $u_i^{n,(l)}$ by scheme (34) with the Thomas algorithm;
              (3) Till $\max_{1\leqslant i\leqslant N-1}|u_i^{n,(l)}-u_i^{n,(l-1)}|<\sigma$;
            end do
            Set $u_i^n\leftarrow u_i^{n,(l)}$;
        end do
Step 3: Set time step $\tau/2$ and $\tau/4$, repeat Step 2 to obtain $u_i^{2n}$ and $u_i^{4n}$;
        Compute the extrapolated numerical solution $u_i^n$ by using (35);
Step 4: Do $n=6,7,\ldots,M$
            Give the boundary function values $u_0^n=g^n$, $u_N^n=l^n$ by (3);
            Set arbitrary initial values $u_i^{n,(0)}$;
            do $l=1,2,\ldots$
              (1) Set $p(u_i^{n,(l)})\leftarrow p(u_i^{n,(l-1)})$, $f(u_i^{n,(l)})\leftarrow f(u_i^{n,(l-1)})$;
              (2) Compute $u_i^{n,(l)}$ by scheme (30) with the Thomas algorithm;
              (3) Till $\max_{1\leqslant i\leqslant N-1}|u_i^{n,(l)}-u_i^{n,(l-1)}|<\sigma$;
            end do
            Set $u_i^n\leftarrow u_i^{n,(l)}$;
        end do
```

Note: l is the iteration number; $\sigma=10^{-12}$.

Since the coefficient matrix of the dsicretized linear system formed at each time step is pentadiagonal and such a linear system can be solved by the Thomas algorithm [24] whose computational complexity reaches $O(11N)$, where N is the order of the matrix. In conclusion, the complexity of the proposed scheme (36) is still not high, even if the nonlinear part is solved by iterative algorithm.

NUMERICAL EXPERIMENTS

To verify the accuracy and reliability of the proposed scheme, three examples are computed and the numerical results are compared with those in the literature. We implement all computations coded with Fortran 90 language and run on a personal computer with 6 GB Intel i5-2320 CPU @ 3 GHz.

The definitions of the L_∞, L_2 norm errors and convergence order are as follows:

$$L_\infty=\max_{0\leqslant i\leqslant N}|u_i^M-u(x_i,t_M)|,$$

$$L_2=\sqrt{h\sum_{i=0}^{N}[u_i^M-u(x_i,t_M)]^2},$$

$$\text{Order}=\frac{\log[L_\infty(h_1)/L_\infty(h_2)]}{\log(h_1/h_2)},$$

where $u(x_i,t_M)$ and u_i^M are, respectively, the exact solution and the numerical solution at the point (x_i,t_M).

The Burgers equation

Firstly, we consider the Burgers equation [25,26]

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\varepsilon\frac{\partial^2 u}{\partial x^2}, \quad x\in[a,b], \quad t\in(1,T],$$

with the initial and boundary conditions

$$\begin{aligned} u(x,1)&=\frac{x}{1+\mathrm{e}^{\left[\frac{1}{4\varepsilon}(x^2-\frac{1}{4})\right]}}, \\ u(a,t)&=\frac{\frac{a}{t}}{1+\left(\frac{t}{t_0}\right)^{\frac{1}{2}}\mathrm{e}^{\frac{a^2}{4\varepsilon t}}}, \\ u(b,t)&=\frac{\frac{b}{t}}{1+\left(\frac{t}{t_0}\right)^{\frac{1}{2}}\mathrm{e}^{\frac{b^2}{4\varepsilon t}}}, \quad t_0=\mathrm{e}^{\frac{1}{8\varepsilon}}. \end{aligned} \tag{37}$$

The exact solution is

$$u(x,t)=\frac{\frac{x}{t}}{1+\left(\frac{t}{t_0}\right)^{\frac{1}{2}}\mathrm{e}^{\frac{x^2}{4\varepsilon t}}}, \quad t_0=\mathrm{e}^{\frac{1}{8\varepsilon}}.$$

Table 1 L_∞ and L_2 norm errors and convergence orders for the Burgers equation in the case $T=2$, $\tau=h$, $a=0$, $b=1.2$.

	$\varepsilon=0.05$			$\varepsilon=0.005$		
N	L_∞	L_2	Order	L_∞	L_2	Order
30	1.141(-8)	5.843(-9)		3.396(-3)	8.947(-3)	
60	1.776(-10)	8.545(-11)	6.01	8.534(-5)	2.100(-5)	5.31
90	1.524(-11)	7.324(-12)	6.06	8.073(-6)	1.967(-6)	5.82
120	2.670(-12)	1.285(-13)	6.05	1.462(-6)	3.577(-7)	5.94

Table 1 shows the L_∞ and L_2 norm errors and convergence orders for the Burgers equation under different mesh numbers N in the case $T=2$, $\tau=h$, $a=0$, $b=1.2$, $\varepsilon=0.05$ and $\varepsilon=0.005$. As can be seen, with increasing mesh number, the calculation results using the proposed scheme can achieve the sixth-order accuracy in space, which shows the global sixth-order accuracy in space is not significantly affected by the local lower-order scheme for neighbouring boundary points.

Table 2 L_∞ and L_2 norm errors for the Burgers equation in the case $T = 3.6$, $\tau = 0.001$, $\varepsilon = 0.005$, $a = 0$, $b = 1$.

	mCBCM [27]		mCBCT4 [26]		HOC [25]		Present	
N	L_∞	L_2	L_∞	L_2	L_∞	L_2	L_∞	L_2
40	1.00 (-3)	3.10 (-4)	8.29 (-5)	2.54 (-5)	1.49 (-3)	4.44 (-4)	1.99 (-4)	5.84 (-5)
60	4.70 (-4)	1.50 (-4)	2.82 (-5)	6.47 (-6)	3.07 (-4)	9.20 (-5)	1.93 (-5)	5.69 (-6)
80	2.70 (-4)	8.00 (-5)	3.34 (-5)	5.08 (-6)	9.96 (-5)	2.98 (-5)	3.57 (-6)	1.05 (-6)
100	1.70 (-4)	2.30 (-5)	3.69 (-5)	5.16 (-6)	4.16 (-5)	1.25 (-5)	9.52 (-7)	2.79 (-7)
120	1.20 (-4)	4.00 (-5)	3.94 (-5)	5.29 (-6)	2.04 (-5)	6.18 (-6)	3.21 (-7)	9.41 (-8)

Table 2 shows the L_∞ and L_2 norm errors under different mesh numbers N in the case $T = 3.6$, $\tau = 0.001$, $\varepsilon = 0.005$, $a = 0$, $b = 1$. It is clear that with increasing mesh number, the calculation results using the proposed scheme exhibit the higher precision than those from the HOC [25], mCBCT4 [26], and mCBCM [27] schemes.

Table 3 L_∞ and L_2 norm errors for the Burgers equation in the case $\varepsilon = 0.005$, $a = 0$, $b = 1.2$.

	HOC [25]		mCBCT4 [26]		MCB-DQM [28]		Present	
(h, τ)	(0.005,0.001)		(0.01,0.001)		(0.01,0.001)		(0.01, 0.01)	
T	L_∞	L_2	L_∞	L_2	L_∞	L_2	L_∞	L_2
1.7	2.56 (-6)	8.04 (-7)	6.64 (-6)	1.53 (-6)	7.77 (-6)	1.91 (-6)	1.68 (-6)	3.68 (-7)
2.5	3.15 (-6)	9.72 (-7)	2.62 (-6)	6.59 (-7)	2.75 (-6)	7.78 (-6)	1.24 (-6)	3.41 (-7)
3.0	3.12 (-6)	9.90 (-7)	1.63 (-6)	4.49 (-7)	1.70 (-6)	5.60 (-7)	1.11 (-6)	3.27 (-7)
3.5	3.02 (-6)	9.87 (-7)	4.09 (-5)	5.82 (-6)	4.36 (-5)	6.12 (-6)	1.01 (-6)	3.15 (-7)

Table 3 shows the L_∞ and L_2 norm errors for different computing times T in the case $\varepsilon = 0.005$, $a = 0$, $b = 1.2$. It can be seen that the proposed scheme with $h = 0.01$ and $\tau = 0.01$ is more accurate than the mCBCT4 [26] and MCB-DQM [28] schemes with $h = 0.01$ and $\tau = 0.001$, and even more accurate than the HOC scheme [25] with smaller mesh sizes $h = 0.005$ and $\tau = 0.001$. These results well reflect the advantages of the high-order accuracy scheme in the calculation. Fig. 2 shows exact and numerical solutions and absolute errors of the numerical solutions at $T = 1.7$, 2.5, and 3.5 in the case $a = 0$, $b = 1.2$, $\varepsilon = 0.005$, $\tau = 0.01$, $h = 0.01$. Fig. 3 depicts a three-dimensional (3D) plot of the numerical solutions and the absolute error of the numerical solutions at $t \in (1, T]$ in the case $a = 0$, $b = 1.2$, $\varepsilon = 0.005$, $\tau = 0.025$, $h = 0.02$, $T = 3.0$.

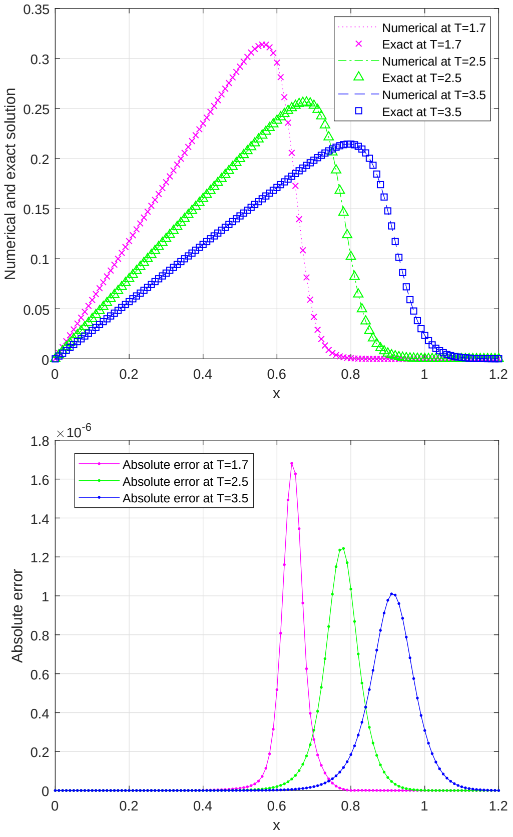

Fig. 2 Numerical results for the Burgers equation at $T = 1.7$, 2.5, and 3.5 in the case $a = 0$, $b = 1.2$, $\varepsilon = 0.005$, $\tau = 0.01$, $h = 0.01$.

The generalized Burgers-Fisher equation

Next, we consider the GBFE [2, 8] as follows

$$\frac{\partial u}{\partial t} + \mu u^s \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \rho u(1-u^s),\ x \in [a, b],\ t \in (0, T],$$

with initial and boundary conditions

$$u(x, 0) = \left[\frac{1}{2} + \frac{1}{2}\tanh(\varphi_1 x)\right]^{1/s},$$
$$u(a, t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh[\varphi_1(a - \varphi_2 t)]\right\}^{1/s},$$
$$u(b, t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh[\varphi_1(b - \varphi_2 t)]\right\}^{1/s},$$

where

$$\varphi_1 = \frac{-\mu s}{2(1+s)}, \quad \varphi_2 = \frac{\mu}{1+s} + \frac{\rho(1+s)}{\mu}.$$

The exact solution is

$$u(x, t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh[\varphi_1(x - \varphi_2 t)]\right\}^{1/s}.$$

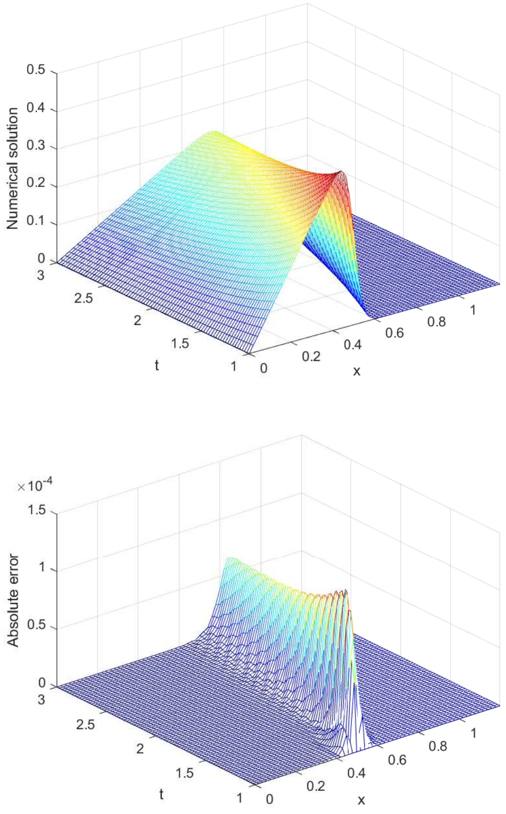

Fig. 3 3D numerical results for the Burgers equation at $t \in (1, T]$ in the case $a = 0$, $b = 1.2$, $\varepsilon = 0.005$, $\tau = 0.025$, $h = 0.02$, $T = 3.0$.

Table 4 L_∞ and L_2 norm errors and convergence orders in space and time for the GBFE in the case $T = 5$, $\mu = \rho = s = 1$, $a = -10$, $b = 20$.

	$\tau = 0.001$				$h = 0.01$		
N	L_∞	L_2	Order	M	L_∞	L_2	Order
30	1.39(-6)	2.79(-6)		30	4.69(-6)	7.91(-6)	
60	2.38(-8)	4.56(-8)	5.87	60	8.34(-8)	1.31(-7)	5.82
120	3.85(-10)	7.19(-10)	5.95	120	1.39(-9)	2.12(-9)	5.91
240	6.02(-12)	1.12(-11)	6.00	240	2.62(-11)	3.87(-11)	5.73
480	1.06(-13)	3.68(-13)	5.83	480	8.00(-13)	1.69(-12)	5.03

Table 5 Absolute errors for the GBFE in the case $N = 10$, $\tau = 0.0001$, $\mu = 0.1$, $\rho = -0.0025$, $a = 0$, $b = 1$.

		$x = 0.1$		$x = 0.5$		$x = 0.9$	
	T	[2]	Present	[2]	Present	[2]	Present
$s = 2$	0.1	1.766(-5)	7.105(-15)	4.492(-5)	3.175(-14)	1.754(-5)	2.021(-14)
	0.2	2.311(-5)	3.619(-14)	6.253(-5)	6.384(-14)	2.298(-5)	2.021(-14)
	0.3	2.514(-5)	1.643(-14)	6.910(-5)	6.861(-14)	2.501(-5)	2.565(-14)
	0.4	2.590(-5)	2.043(-14)	7.155(-5)	5.418(-14)	2.577(-5)	2.653(-14)
	0.5	2.618(-5)	1.854(-14)	7.246(-5)	5.285(-14)	2.605(-5)	1.699(-14)
$s = 4$	0.1	1.262(-5)	4.996(-15)	3.214(-5)	1.721(-14)	1.257(-5)	1.110(-15)
	0.2	1.653(-5)	5.551(-16)	4.476(-5)	5.440(-15)	1.647(-5)	9.104(-15)
	0.3	1.798(-5)	2.220(-15)	4.947(-5)	7.883(-15)	1.792(-5)	9.992(-16)
	0.4	1.852(-5)	1.443(-15)	5.122(-5)	2.232(-14)	1.846(-5)	8.882(-16)
	0.5	1.872(-5)	4.441(-15)	5.187(-5)	4.552(-15)	1.866(-5)	5.551(-15)
$s = 8$	0.1	7.659(-5)	7.550(-15)	1.951(-5)	2.354(-14)	7.636(-5)	1.055(-14)
	0.2	1.003(-5)	4.663(-15)	2.719(-5)	2.087(-14)	1.001(-5)	2.220(-14)
	0.3	1.091(-5)	2.076(-14)	3.004(-5)	6.328(-14)	1.089(-5)	3.331(-15)
	0.4	1.124(-5)	1.643(-14)	3.111(-5)	3.297(-14)	1.122(-5)	1.366(-14)
	0.5	1.136(-5)	7.328(-15)	3.150(-5)	2.076(-14)	1.134(-5)	9.326(-15)

Table 6 L_∞ norm errors for the GBFE in the case $N = 10$, $\tau = 0.0001$, $s = 1$, $a = 0$, $b = 1$.

		$\mu = 0.001$			$\mu = 0.0001$		
T	ρ	[6]	[8]	Present	[6]	[8]	Present
0.1	1	2.900(-13)	2.427(-12)	2.332(-15)	2.882(-13)	2.425(-12)	1.610(-14)
	10	3.318(-13)	1.283(-13)	3.109(-14)	3.330(-13)	1.283(-13)	4.308(-14)
	100	2.423(-13)	1.250(-12)	1.044(-13)	2.448(-13)	1.250(-12)	1.009(-13)
0.5	1	4.656(-13)	2.235(-12)	4.241(-14)	4.583(-13)	2.233(-12)	6.617(-14)
	10	7.049(-13)	1.250(-13)	3.075(-14)	6.981(-13)	1.250(-13)	4.818(-14)
	100	2.394(-13)	1.250(-12)	2.498(-14)	5.195(-13)	1.250(-12)	4.308(-14)

Table 4 shows the L_∞ and L_2 norm errors and convergence orders using the proposed scheme in the case $T = 5$, $\mu = \rho = s = 1$, $a = -10$, $b = 20$. The results show that the proposed method has the sixth-order spatial accuracy with temporal step size $\tau = 0.001$ and the sixth-order temporal accuracy with $h = 0.01$, which are consistent with the theoretical result. Table 5 shows the comparisons of the present results with those of [2] for the GBFE. Here, we use the same parameters as in [2], namely, $N = 10$, $\tau = 0.0001$, $a = 0$, and $b = 1$. The absolute errors ($|u_i^M - u(x_i, t_M)|$) at different points x at different calculation times T are shown for $\mu = 0.1$, $\rho = -0.0025$, and $s = 2$, 4 and 8 in Table 5. It is easy to see that the results from the proposed scheme are more accurate than those obtained in [2]. Table 6 shows the L_∞ norm errors using the proposed scheme and comparison with the results in [6, 8] in the case $N = 10$, $\tau = 0.0001$, $s = 1$, $a = 0$, $b = 1$. It can be clearly seen that the results from this scheme are one to two orders of magnitude better than those computed in [6, 8] with different values of μ and ρ for different computation times T.

The generalized Burgers-Huxley equation

Finally, we consider the GBHE [2],

$$\frac{\partial u}{\partial t} + \mu u^s \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \rho u(1-u^s)(u^s - \theta), \qquad x \in [a,b],\ t \in (0,T],$$

with initial and boundary conditions

$$u(x,0) = \left[\frac{\theta}{2} + \frac{\theta}{2}\tanh(\phi_1 x)\right]^{1/s},$$

$$u(a,t) = \left\{\frac{\theta}{2} + \frac{\theta}{2}\tanh[\phi_1(a - \phi_2 t)]\right\}^{1/s},$$

$$u(b,t) = \left\{\frac{\theta}{2} + \frac{\theta}{2}\tanh[\phi_1(b - \phi_2 t)]\right\}^{1/s},$$

where

$$\phi_1 = \frac{-\mu s + s\sqrt{\mu^2 + 4\rho(1+s)}}{4(1+s)}\theta,$$

$$\phi_2 = \frac{\mu\theta}{s+1} - \frac{(1+s-\theta)[-\mu + \sqrt{\mu^2 + 4\rho(1+s)}]}{2(1+s)}.$$

The exact solution is

$$u(x,t) = \left\{\frac{\theta}{2} + \frac{\theta}{2}\tanh[\phi_1(x - \phi_2 t)]\right\}^{1/s}.$$

Table 7 L_∞ and L_2 norm errors and convergence orders for the GBHE in the case $T = 5$, $\tau = 0.5h$, $\mu = \rho = s = 1$, $\theta = 2$, $a = -10$, $b = 20$.

	[11]			Present		
N	L_∞	L_2	Order	L_∞	L_2	Order
80	1.284(-4)	1.495(-4)		1.993(-5)	2.426(-5)	
160	8.614(-6)	1.017(-5)	3.90	3.635(-7)	4.373(-7)	5.78
320	5.572(-7)	6.683(-7)	3.95	6.061(-9)	7.281(-9)	5.91
640	3.499(-8)	4.236(-8)	3.99	9.640(-11)	1.154(-11)	5.97

Table 7 shows the L_∞, L_2 norm errors and convergence orders under different mesh numbers N in the case $T = 5$, $\tau = 0.5h$, $\mu = \rho = s = 1$, $\theta = 2$, $a = -10$, $b = 20$. It is clear that the calculation results using the proposed scheme can reach the sixth-order in space, and the calculation results using the proposed scheme exhibit the higher precision than those results got in [11] with increasing mesh number. Table 8 shows the L_∞, L_2 norm errors and convergence orders under different mesh numbers M in the case $T = 10$, $N = 4096$, $\mu = \rho = s = 1$, $\theta = 2$, $a = -10$, $b = 20$. It can also be clearly seen that the proposed method can achieve the sixth-order accuracy in time and the calculated results are much more accurate than those obtained in [11].

Table 8 L_∞ and L_2 norm errors and convergence orders for the GBHE in the case $T = 10$, $N = 4096$, $\mu = \rho = s = 1$, $\theta = 2$, $a = -10$, $b = 20$.

	[11]			Present		
M	L_∞	L_2	Order	L_∞	L_2	Order
64	1.607(-4)	2.395(-4)		4.603(-6)	6.262(-6)	
128	1.088(-5)	1.626(-5)	3.88	9.081(-8)	1.265(-7)	5.66
256	6.984(-7)	1.045(-6)	3.96	1.448(-9)	2.011(-9)	5.97
512	4.414(-8)	6.607(-8)	3.98	6.009(-12)	1.016(-11)	7.91

Fig. 4 shows numerical and exact solutions and the absolute errors of the numerical solutions at $T = 1$, 5, and 10 in the case $N = 100$, $\tau = 0.01$, $\mu = \rho = 1$, $\theta = 2$, $s = 1$, $a = -10$, $b = 20$. Fig. 5 shows 3D views of the numerical solution and its absolute error in the case $N = 100$, $\tau = 0.025$, $T = 2$, $\mu = \rho = 1$, $\theta = 2$, $s = 1$, $a = -10$, $b = 20$. It is easy to see that the proposed scheme obtains very accurate solutions for this problem.

CONCLUSION

A sixth-order finite difference method has been developed for numerical solution of the GBFE and GBHE. For the space derivatives, the five-point sixth-order difference scheme is used to discretize the interior points of the space, and the three-point fourth-order scheme is used to discretize the neighbouring boundary points of the space. For the time derivative, a sixth-order backward difference formula is used to form a seven-step difference scheme, and then the Crank-Nicolson scheme with the Richardson extrapolation technique are used to provide the solutions with the sixth-order accuracy at the first five starting time steps. The proposed scheme has the sixth-order accuracy in both space and time. And then the resulting linear system at each time step is solved by the efficient Thomas algorithm. Some numerical examples are conducted to demonstrate the high accuracy of the proposed scheme and its superiority to other numerical methods in the literature.

The presented method can be extended to solve other multi-dimensional nonlinear partial differential equations. And our ongoing work and the relevant results will be reported in the near future.

Appendix A: The coefficients of the spatial difference scheme

The following is the spatially discrete form of the scheme (26). The discrete expressions of the operators $A_x^6 u_i$ and $L_x^6 S_i$ at point x_i and the corresponding

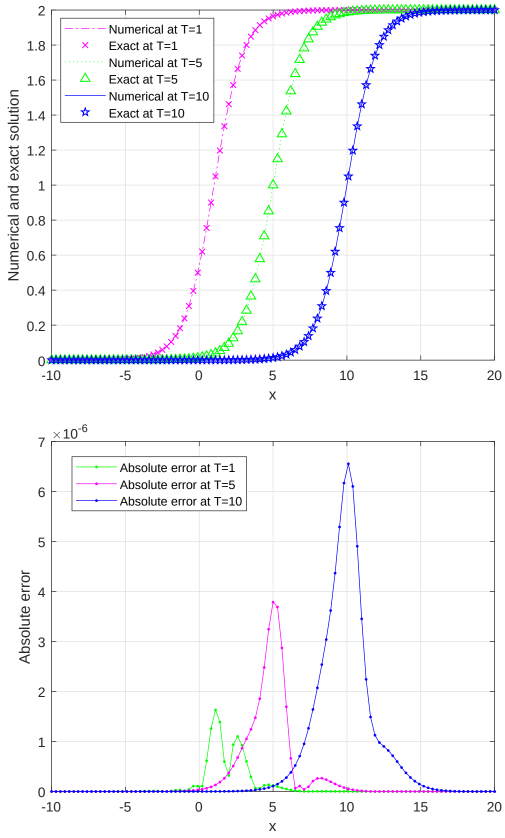

Fig. 4 Numerical results for the GBHE at $T = 1$, 5, and 10 in the case$\mu = \rho = 1$,$\theta = 2$, $s = 1$, $a = -10$, $b = 20$, $\tau = 0.01$, $N = 100$.

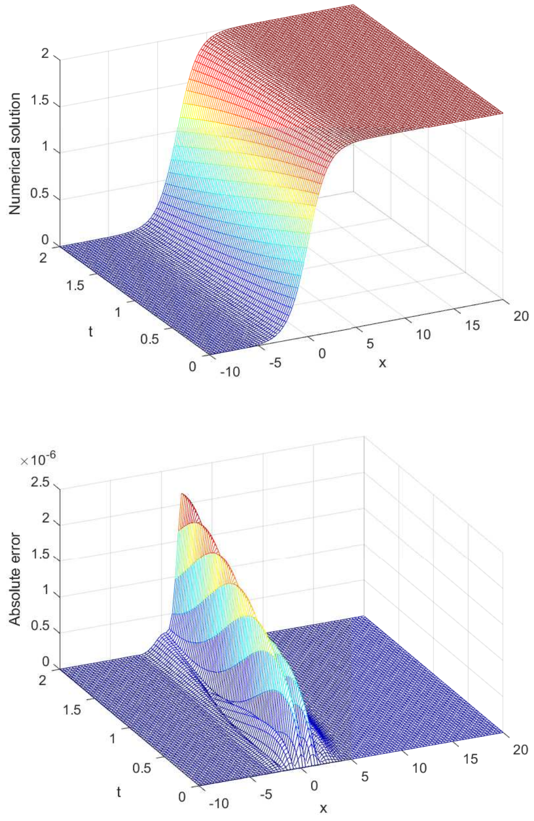

Fig. 5 3D numerical results for the GBHE at $t \in (0, T]$ in the case $T = 2$, $\mu = \rho = 1$, $\theta = 2$, $s = 1$, $a = -10$, $b = 20$, $\tau = 0.025$, $N = 100$.

coefficients are written out as follows

$$A_x^6 u_i = A_0^6 u_{i-2} + A_1^6 u_{i-1} + A_2^6 u_i + A_3^6 u_{i+1} + A_4^6 u_{i+2},\quad i = 2, 3, \dots, N-2.$$

where

$$A_0^6 = \frac{\varepsilon}{60h^2} + \frac{p_i^2}{180\varepsilon} - \frac{hp_i\bar{D}_x p_i}{180\varepsilon} + \frac{p_i}{60h} - \frac{\bar{D}_x p_i}{90} + \frac{h\bar{D}_x^2 p_i}{180},$$

$$A_1^6 = -\frac{16\varepsilon}{15h^2} - \frac{4p_i^2}{45\varepsilon} + \frac{2hp_i\bar{D}_x p_i}{45\varepsilon} - \frac{8p_i}{15h} + \frac{8\bar{D}_x p_i}{45} - \frac{2h\bar{D}_x^2 p_i}{45},$$

$$A_2^6 = \frac{21\varepsilon}{10h^2} + \frac{p_i^2}{6\varepsilon} - \frac{\bar{D}_x p_i}{3},$$

$$A_3^6 = -\frac{16\varepsilon}{15h^2} - \frac{4p_i^2}{45\varepsilon} - \frac{2hp_i\bar{D}_x p_i}{45\varepsilon} + \frac{8p_i}{15h} + \frac{8\bar{D}_x p_i}{45} + \frac{2h\bar{D}_x^2 p_i}{45},$$

$$A_4^6 = \frac{\varepsilon}{60h^2} + \frac{p_i^2}{180\varepsilon} + \frac{hp_i\bar{D}_x p_i}{180\varepsilon} - \frac{p_i}{60h} - \frac{\bar{D}_x p_i}{90} - \frac{h\bar{D}_x^2 p_i}{180},$$

and

$$L_x^6 S_i = \left(-\frac{1}{180} - \frac{hp_i}{180\varepsilon}\right)S_{i-2} + \left(\frac{4}{45} + \frac{2hp_i}{45\varepsilon}\right)S_{i-1} + \left(\frac{5}{6}\right)S_i + \left(\frac{4}{45} - \frac{2hp_i}{45\varepsilon}\right)S_{i+1} + \left(\frac{hp_i}{180\varepsilon} - \frac{1}{180}\right)S_{i+2},\quad i = 2, 3, \dots, N-2.$$

Similarly, the discrete expressions of the operators $A_x^4 u_i$ and $L_x^4 S_i$ at point x_i and the corresponding coefficients are

$$A_x^4 u_i = A_0^4 u_{i-1} + A_1^4 u_i + A_2^4 u_{i+1},\qquad i = 1, N-1.$$

where

$$A_0^4 = -\frac{\varepsilon}{h^2} - \frac{p_i^2}{12\varepsilon} + \frac{hp_i D_x p_i}{24\varepsilon} - \frac{p_i}{2h} + \frac{D_x p_i}{6} - \frac{hD_x^2 p_i}{24},$$

$$A_1^4 = \frac{2\varepsilon}{h^2} + \frac{p_i^2}{6\varepsilon} - \frac{D_x p_i}{3},$$

$$A_2^4 = -\frac{\varepsilon}{h^2} - \frac{p_i^2}{12\varepsilon} - \frac{hp_i D_x p_i}{24\varepsilon} + \frac{p_i}{2h} + \frac{D_x p_i}{6} + \frac{hD_x^2 p_i}{24},$$

and

$$L_x^4 S_i = \left(\tfrac{1}{12} + \tfrac{hp_i}{24\varepsilon}\right)S_{i-1} + \left(\tfrac{5}{6}\right)S_i + \left(\tfrac{hp_i}{24\varepsilon}\right)S_{i+1},\ i = 1, N-1.$$

Acknowledgements: This work is partially supported by National Natural Science Foundation of China (12161067, 11961054, 11902170), National Natural Science Foundation of Ningxia (2022AAC02023, 2020AAC03059), and National Youth Top-notch Talent Support Program of Ningxia.

REFERENCES

1. Sari M, Gürarslan G, Zeytinoğlu A (2011) High-order finite difference schemes for numerical solutions of the generalized Burgers-Huxley Equation. *Numer Methods Partial Differ Equations* **27**, 1313–1326.
2. Hammad DA, El-Azab MS (2015) 2*N* order compact finite difference scheme with collocation method for solving the generalized Burger's-Huxley and Burger's-Fisher equations. *Appl Math Comput* **258**, 296–311.
3. Satsuma J, Ablowitz M, Fuchssteiner B, Kruskal M (1987) *Topics in Soliton Theory and Exactly Solvable Nonlinear Equations*, World Scientific, Singapore.
4. Whitham GB (1974) *Linear and Nonlinear Waves*, Wiley, New York, NY.
5. Sari M, Grarslan G, Dag İ (2010) A compact finite difference method for the solution of the generalized Burgers-Fisher equation. *Numer Methods Partial Differ Equations* **26**, 125–134.
6. Zhang R, Yu X, Zhao G (2012) The local discontinuous Galerkin method for Burgers-Huxley and Burgers-Fisher equations. *Appl Math Comput* **218**, 8773–8778.
7. Duan Y, Kong L, Zhang R (2012) A lattice Boltzmann model for the generalized Burgers-Huxley equation. *Phys A* **391**, 625–632.
8. Mittal RC, Tripathi A (2015) Numerical solutions of generalized Burgers-Fisher and generalized Burgers-Huxley equations using collocation of cubic B-splines. *Int J Comput Math* **92**, 1053–1077.
9. Singh A, Dahiya S, Singh SP (2020) A fourth-order B-spline collocation method for nonlinear Burgers-Fisher equation. *Math Sci* **14**, 75–85.
10. Ray SS, Gupta AK (2014) Comparative analysis of variational iteration method and Haar wavelet method for the numerical solutions of Burgers-Huxley and Huxley equations. *J Math Chem* **52**, 1066–1080.
11. Zhang L, Ge Y (2021) Numerical solution of nonlinear advection diffusion reaction equation using high-order compact difference method. *Appl Numer Math* **166**, 127–145.
12. Kumar S, Ray SS (2021) Numerical treatment for Burgers-Fisher and generalized Burgers-Fisher equations. *Math Sci* **15**, 21–28.
13. Mohanty RK, Sharma S (2020) A high-resolution method based on off-step non-polynomial spline approximations for the solution of Burgers-Fisher and coupled nonlinear Burgers' equations. *Eng Comput* **37**, 2785–2818.
14. Sun L, Zhu C (2020) Cubic B-spline quasi-interpolation and an application to numerical solution of generalized Burgers-Huxley equation. *Adv Mech Eng* **12**, 1–8.
15. Zibaei S, Zeinadini M, Namjoo M (2016) Numerical solutions of Burgers-Huxley equation by exact finite difference and NSFD schemes. *J Difference Equ Appl* **22**, 1098–1113.
16. Namjoo M, Zeinadini M, Zibaei S (2018) Nonstandard finite-difference scheme to approximate the generalized Burgers-Fisher equation. *Math Methods Appl Sci* **41**, 8212–8228.
17. Gear C (1971) *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ.
18. Abramowitz M, Stegun IA (1970) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., Mineola, NY.
19. Shih TM (1984) *Numerical Heat Transfer*, Hemisphere, New York, NY.
20. Feng HS, Zhao S (2020) FFT-based high order central difference schemes for three-dimensional Poisson's equation with various types of boundary conditions. *J Comput Phys* **410**, 190391.
21. Wang Z, Ge Y, Sun H, Sun T (2022) Sixth-order quasi-compact difference schemes for 2D and 3D Helmholtz equations. *Appl Math Comput* **431**, 127347.
22. Svärd M, Nordström J (2014) Review of summation-by-parts schemes for initial-boundary-value problems. *J Comput Phys* **268**, 17–38.
23. Zhou H, Wu Y, Tian W (2012) Extrapolation algorithm of compact ADI approximation for two-dimensional parabolic equation. *Appl Math Comput* **219**, 2875–2884.
24. Xu X, Ma D (2013) The solution of penta-diagonal and ninefold-diagonal linear equations by Thomas method. *J Yangtze Univ (Nat Sci Ed)* **10**, 5–9+4.
25. Yang X, Ge Y, Zhang L (2019) A class of high-order compact difference schemes for solving the Burgers equations. *Appl Math Comput* **358**, 394–417.
26. Singh BK, Gupta M (2021) A new efficient fourth order collocation scheme for solving Burgers equation. *Appl Math Comput* **399**, 126011.
27. Mittal RC, Jain RK (2012) Numerical solutions of nonlinear Burgers equation with modified cubic B-splines collocation method. *Appl Math Comput* **218**, 7839–7855.
28. Arora G, Singh BK (2013) Numerical solution of Burgers equation with modified cubic B-spline differential quadrature method. *Appl Math Comput* **224**, 166–177.