Computational and Applied Mathematics (2023) 42:201
https://doi.org/10.1007/s40314-023-02345-3

Numerical solution of time dependent nonlinear partial differential equations using a novel block method coupled with compact finite difference schemes

Akansha Mehta[1,2] · Gurjinder Singh[2] · Higinio Ramos[3,4]

Received: 30 November 2022 / Revised: 6 April 2023 / Accepted: 2 May 2023 /
Published online: 2 June 2023

Abstract

In this paper, we have developed a novel three step second derivative block method and coupled it with fourth order standard compact finite difference schemes for solving time dependent nonlinear partial differential equations (PDEs) of physical relevance. Two well-known problems viz. the FitzHugh–Nagumo equation and the Burgers' equation have been considered as test problems to check the effectiveness of the proposed scheme. Firstly, we developed a novel block scheme and discussed its characteristics for solving initial-value systems, such as the one resulting from the discretization of the spatial derivatives that appear in the PDEs. Although many time integration techniques already exist to solve discretized PDEs, our goal is to develop a numerical scheme keeping in mind saving computational time while maintaining good accuracy. The proposed block scheme has been proved to be $\mathcal{A}$-stable and consistent. The method performs well for solving the stiff case of the FitzHugh–Nagumo equation, as well as for solving the Burgers equation at different values of viscosity and time. The numerical experiments reveal that the developed numerical scheme is computationally efficient.

Communicated by Fabricio Simeoni de Sousa.

✉ Gurjinder Singh
gurjinder11@gmail.com

Akansha Mehta
mehta.akansha49@gmail.com

Higinio Ramos
higra@usal.es

1 Department of Mathematical Sciences, I. K. Gujral Punjab Technical University Jalandhar, Main Campus, Kapurthala, Punjab 144603, India

2 Department of Applied Sciences, I. K. Gujral Punjab Technical University Jalandhar, Main Campus, Kapurthala, Punjab 144603, India

3 Scientific Computing Group, Universidad de Salamanca, Plaza de la Merced, 37008 Salamanca, Spain

4 Escuela Politécnica Superior de Zamora, Campus Viriato, 49022 Zamora, Spain

Keywords Nonlinear PDEs · Block methods · Compact finite difference schemes · Stability

Mathematics Subject Classification 65M06 · 65N35 · 35F50

1 Introduction

Many phenomena of physical importance are modelled by nonlinear PDEs. Since the analytical approach to obtain exact solutions of nonlinear PDEs is only applicable to a small class of problems, and in this case, one of the possible choices to tackle the given problem is the consideration of numerical approximations to the solution (Collatz 1966; Debnath 2012). Developing novel efficient numerical schemes or modifying existing ones to solve these problems is a permanent goal in this field. In this article, our work considers the approximate solution of one-dimensional time dependent initial-boundary value PDEs of the form

$$u_t = F(x, t, u, u_x, u_{xx}), \qquad \text{with} \qquad a \leq x \leq b, \qquad t \geq t_0, \tag{1}$$

along with the initial condition

$$u(x, t_0) = g(x),$$

and boundary conditions as:

$$u(a, t) = g_1(t), \quad u(b, t) = g_2(t),$$

where u, x and t represent the solution of the problem, and the space and time variables, respectively. The spatial semi-discretization of the above problem results into a system of first order ordinary differential equations in t as follows

$$\frac{dU}{dt} = f(t, U) \qquad \text{and} \qquad U(t_0) = U_0, \tag{2}$$

which can then be solved by any existing time integration method, for instance, Runge–Kutta methods or linear multi-step methods. In this article, we have considered two non-linear time dependent PDEs viz. the FitzHugh–Nagumo equation and the Burgers' equation, that have a great number of practical applications. The FitzHugh–Nagumo equation is a non-linear reaction diffusion equation arising in the field of science and technology, especially in neurophysiology and population growth models. The Burgers' equation is also a non-linear PDE arising mainly in the fields of turbulence modeling and shock theory. It involves both connective and diffusive effects. To obtain numerical solutions of FitzHugh–Nagumo equations, some researchers have considered finite difference methods as well as compact difference methods as in Agbavon and Appadu (2020), Akkoyunlu (2019). Moreover, the authors in Ramos et al. (2022) used a cubic B-spline approach to obtain numerical solutions of the FitzHugh–Nagumo equation. Similarly, for solving Burgers' equation, many different types of approaches have been considered in the literature, for instance, a predictor-corrector scheme (Zhang and Wang 2012), a high order compact finite difference scheme (Yang et al. 2019), a cubic B-splines collocation method (Mittal and Jain 2012), or a Crank–Nicolson technique (Kadalbajoo and Awasthi 2006). For over the last fifty years, compact finite differences schemes (CFDS) have been popularly used over the standard finite difference schemes (FDS) due to their small stencil size and better accuracy (Adam 1975; Li and Visbal 2006; Tyler 2007). But, so far, the performance of compact finite difference schemes have not been analyzed by combining them with block methods to compute numerical solutions of time dependent PDEs. Compact finite difference schemes are used over the standard finite difference schemes to optimize accuracy in approximating the spatial derivatives appearing in a given problem. Block methods are self starting methods firstly proposed by Milne (1953) for getting approximate solution of a system of first order ordinary differential equations. These methods provide an approximate solution for more than one point at a time and save computational time while maintaining accuracy (Lambert 1973). For more details on block methods and their implementations, one can consult (Lambert 1973; Ramos and Singh 2017; Ramos et al. 2022; Shampine and Watts 1969; Singh and Ramos 2018). Our objective here is to extend the applicability of block methods by coupling them with compact finite difference schemes, for solving time dependent PDEs.

2 Derivation of a three step block method

To approximate the solution of problem (2), we will firstly discretize the time domain $[t_0, t_k]$ into k steps of equal width $h = \frac{t_k - t_0}{k}$ with grid points as: $t_0 < t_1 < t_2 \cdots < t_k$. For the sake of simplicity, we will develop the method for a scalar problem $u' = f(t, u), u(t_0) = u_0$, although it could be applied using a componentwise implementation for solving a system like the one in (2). Consider a polynomial approximation to the exact solution of this problem on the interval $[t_n, t_{n+3}]$ as:

$$u(t) \approx p(t) = \sum_{n=0}^{6} a_n t^n. \tag{3}$$

The choice of the degree of the above polynomial must be a compromise between the accuracy that one wants to achieve and the complexity of the method that is obtained. In our case, the interpolatory and collocation conditions imposed below determine that the maximum degree of the approximation polynomial is six. Then, we have

$$u'(t) \approx p'(t) = \sum_{n=1}^{6} n a_n t^{n-1} \tag{4}$$

$$\text{and} \qquad u''(t) \approx p''(t) = \sum_{n=2}^{6} n(n-1) a_n t^{n-2}, \tag{5}$$

where $a_n \in \mathbf{R}$ are coefficients to be determined. Note that there are seven unknown coefficients to be determined in (3) and to get them, we impose the following interpolatory and collocation conditions

$$\begin{aligned} p(t_n) = u_n, \, p'(t_n) = f_n, \, p'(t_{n+1}) = f_{n+1}, \\ p'(t_{n+2}) = f_{n+2}, \, p'(t_{n+3}) = f_{n+3}, \\ p''(t_n) = f'_n, \, p''(t_{n+3}) = f'_{n+3}. \end{aligned}$$

Here, u_{n+j}, f_{n+j} and f'_{n+j} are respectively approximation to $u(t_{n+j})$, $u'(t_{n+j})$ and $u''(t_{n+j})$. The above conditions result in the following system of equations that can be written in matrix form as

$$\begin{bmatrix} 1 & t_n & t_n^2 & t_n^3 & t_n^4 & t_n^5 & t_n^6 \\ 0 & 1 & 2t_n & 3t_n^2 & 4t_n^3 & 5t_n^4 & 6t_n^5 \\ 0 & 1 & 2t_{n+1} & 3t_{n+1}^2 & 4t_{n+1}^3 & 5t_{n+1}^4 & 6t_{n+1}^5 \\ 0 & 1 & 2t_{n+2} & 3t_{n+2}^2 & 4t_{n+2}^3 & 5t_{n+2}^4 & 6t_{n+2}^5 \\ 0 & 1 & 2t_{n+3} & 3t_{n+3}^2 & 4t_{n+3}^3 & 5t_{n+3}^4 & 6t_{n+3}^5 \\ 0 & 0 & 2 & 6t_n & 12t_n^2 & 20t_n^3 & 30t_n^4 \\ 0 & 0 & 2 & 6t_{n+3} & 12t_{n+3}^2 & 20t_{n+3}^3 & 30t_{n+3}^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} u_n \\ f_n \\ f_{n+1} \\ f_{n+2} \\ f_{n+3} \\ f'_n \\ f'_{n+3} \end{bmatrix}$$

The values of the seven unknowns appearing in the above system of equations have been obtained by using the *Mathematica* system. The values of the unknown coefficients are not displayed here as they are cumbersome expressions. By putting these values and changing the variable t to $t_n + mh$, the polynomial in (3) can be re-written as

$$p(t_n + mh) = b_0 u_n + h(b_1 f_n + b_2 f_{n+1} + b_3 f_{n+2} + b_4 f_{n+3}) + h^2(b_5 f'_n + b_6 f'_{n+3}), \quad (6)$$

where the coefficients $b_i's$ are continuous functions of the variable m. In order to get complete structure of the block method, the polynomial given in (6) has been evaluated for $m = 1, 2$ and 3. This results in a three step block method given by

$$\begin{aligned} u_{n+1} &= u_n + \frac{h}{6480}(3463 f_n + 3537 f_{n+1} - 783 f_{n+2} + 263 f_{n+3}) \\ &\quad + \frac{h^2}{6480}(582 f'_n - 102 f'_{n+3}) \\ u_{n+2} &= u_n + \frac{h}{405}(181 f_n + 459 f_{n+1} + 189 f_{n+2} - 19 f_{n+3}) + \frac{h^2}{405}(24 f'_n + 6 f'_{n+3}) \\ u_{n+3} &= u_n + \frac{h}{80}(39 f_n + 81 f_{n+1} + 81 f_{n+2} + 39 f_{n+3}) + \frac{h^2}{80}(6 f'_n - 6 f'_{n+3}) \end{aligned} \quad (7)$$

The method given above is a three-step second derivative block method that will produce approximate solutions of the initial-value problem at the points t_{n+1}, t_{n+2} and t_{n+3} simultaneously.

3 Characteristics of the block method

In this section, the basic characteristics of the method (7) have been analyzed.

3.1 Error analysis, order and consistency of the method

Firstly, consider a difference operator $\mathcal{L}_j$ related to the three step block method given by (7) as:

$$\begin{aligned} \mathcal{L}_j(u(t), h) &= u(t + jh) \\ &\quad - F_j[h, u(t), u'(t), u'(t+h), u'(t+2h), u'(t+3h), u''(t), u''(t+3h)] \end{aligned} \quad (8)$$

with $j = 1, 2, 3$ and F_j is the corresponding right hand side of each formula. Expanding the above expression by the usual Taylor series about the point t and combining the like terms in h, the local truncation errors of each formula given in (7), are obtained as

$$\mathcal{LTE}_1 = \frac{-97u^7(t)h^7}{100800} + O(h^8),$$
$$\mathcal{LTE}_2 = \frac{u^7(t)h^7}{6300} + O(h^8),$$
$$\mathcal{LTE}_3 = \frac{-9u^7(t)h^7}{11200} + O(h^8).$$

The above expressions for local truncation errors suggest that the method is of order six. Moreover, the proposed block method is consistent since it has an algebraic order ≥ 1 (Ramos and Singh 2017).

3.2 Zero-stability

The stability of any numerical method is an important aspect that characterizes a small change in the solution under a small perturbation in the initial conditions. The proposed block method given in (7) is said to be zero-stable if all the roots of its first characteristic polynomial have magnitude ≤ 1 and the roots with unit modulus, if any, must be simple. Now, as $h \to 0$, the proposed method takes the form:

$$IU_n - BU_{n-1} = 0,$$

where $U_n = (u_{n+1}, u_{n+2}, u_{n+3})^T$ and $U_{n-1} = (u_{n-2}, u_{n-1}, u_n)^T$,

$$B = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

and I being the identity matrix. The characteristic equation for the above method will be $|B - \lambda I| = \lambda^2(1 - \lambda) = 0$, having characteristic roots as $\{0, 0, 1\}$. Thus, the proposed method is zero-stable.

3.3 Linear stability analysis

The linear stability analysis of a numerical scheme can be carried out by applying it to the Dahlquist's test equation given by:

$$u' = \lambda u, \qquad Re(\lambda) < 0. \tag{9}$$

Any exact solution $ce^{\lambda t}$ to the above Eq. (9) will be damped out as t approaches to ∞. The behavior of the numerical solution should also mimic the nature of the true solution for the method to be stable. Now, applying the proposed block method to Eq. (9) and substituting $\lambda h = \bar{h}$, the resulting difference system can be written in matrix form as:

$$A \begin{bmatrix} u_{n+1} \\ u_{n+2} \\ u_{n+3} \end{bmatrix} = B \begin{bmatrix} u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix},$$

where, the matrix A is given by:

$$A = \begin{bmatrix} 1 - \frac{3537\bar{h}}{6480} & \frac{783\bar{h}}{6480} & \frac{102\bar{h}^2}{6480} - \frac{263\bar{h}}{6480} \\ \frac{-459\bar{h}}{405} & 1 - \frac{189\bar{h}}{405} & \frac{-6\bar{h}^2}{405} + \frac{19\bar{h}}{405} \\ -\frac{81\bar{h}}{80} & \frac{-81\bar{h}}{80} & 1 + \frac{6\bar{h}^2}{80} - \frac{39\bar{h}}{80} \end{bmatrix},$$

and the matrix B:

$$B = \begin{bmatrix} 0 & 0 & 1 + \frac{3463\bar{h}}{6480} + \frac{582\bar{h}^2}{6480} \\ 0 & 0 & 1 + \frac{181\bar{h}}{405} + \frac{24\bar{h}^2}{405} \\ 0 & 0 & 1 + \frac{39\bar{h}}{80} + \frac{6\bar{h}^2}{80} \end{bmatrix}.$$

So, the final form of the method applied to the test equation results in:

$$\begin{bmatrix} u_{n+1} \\ u_{n+2} \\ u_{n+3} \end{bmatrix} = M(\bar{h}) \begin{bmatrix} u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix}.$$

Here, the matrix

$$M(\bar{h}) = A^{-1}B$$

is the corresponding stability matrix. The eigenvalues of this stability matrix are $\{0, 0, P(\bar{h})\}$, where

$$P(\bar{h}) = \frac{120 + 180\bar{h} + 116\bar{h}^2 + 39\bar{h}^3 + 6\bar{h}^4}{120 - 180\bar{h} + 116\bar{h}^2 - 39\bar{h}^3 + 6\bar{h}^4}.$$

The region of absolute stability is defined as (see Hairer and Wanner 1996):

$$S = \{\bar{h} \in \mathbb{C} : |P(\bar{h})| < 1\}.$$

A method is said to be $\mathcal{A}$-stable if the left half of the complex plane is contained within S. Figure 1 below shows the stability region for the proposed block method, which is $\mathcal{A}$-stable.

4 Compact finite difference scheme for the spatial derivatives

To get a semi-discretization of a given PDE, the spatial derivatives present in the PDE will be approximated using standard fourth order compact finite difference schemes (for details on compact finite difference schemes, [see Lele (1992), Li and Chen (2008), Li and Visbal (2006), Tyler (2007) and references therein].) To do that, firstly discretize the space interval $[a, b]$ into N equal parts as follows:

$$a = x_1 < x_2 < \cdots < x_N < x_{N+1} = b$$

with a uniform mesh-size $h_x = x_{i+1} - x_i$. Now, consider a fourth order compact discretization of tri-diagonal nature (Li and Chen 2008) for approximating first order spatial derivatives of $u(x, t)$ at the interior nodes as

$$\frac{1}{4}u'_{i-1} + u'_i + \frac{1}{4}u'_{i+1} = \frac{3}{4h_x}(u_{i+1} - u_{i-1}); \qquad i = 2, 3, ..., N,$$

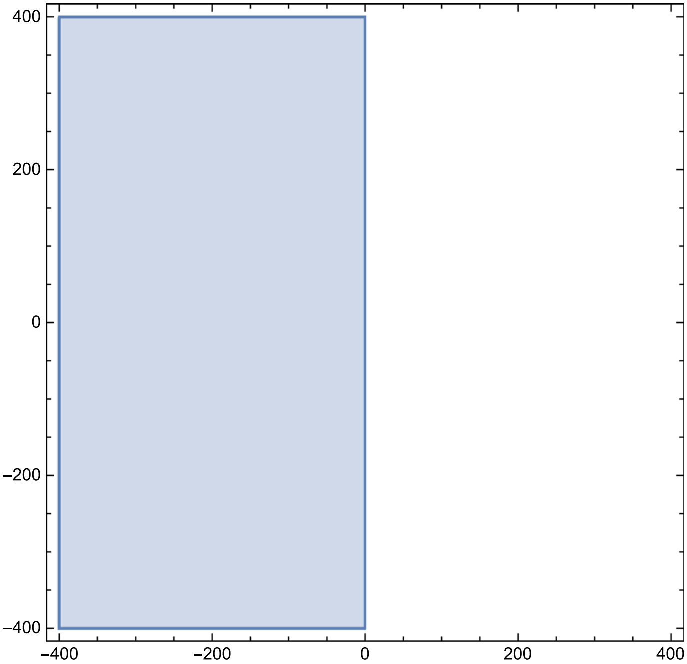

Fig. 1 Stability region of the proposed method

and then to approximate the first order derivatives at the boundary points, consider one sided boundary schemes of the same order as that of the scheme for interior nodes as follows

for $i = 1$

$$u_1' + 3u_2' = \frac{-17}{6h_x}u_1 + \frac{3}{2h_x}(u_2 + u_3) - \frac{1}{6h_x}u_4,$$

for $i = N + 1$,

$$3u_N' + u_{N+1}' = \frac{17}{6h_x}u_{N+1} - \frac{3}{2h_x}(u_N + u_{N-1}) + \frac{1}{6h_x}u_{N-2}.$$

For more details on one sided boundary schemes, one can consult (Lele 1992). The complete system of equations obtained above from the fourth order compact discretization of first order derivatives can be written in matrix form as:

$$F_1 U' = F_2 U, \tag{10}$$

where the matrices F_1 and F_2 are written as

$$F_1 = \begin{bmatrix} 1 & 3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1/4 & 1 & 1/4 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1 & 1/4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1/4 & 1 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1/4 & 1 & 1/4 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 3 & 1 \end{bmatrix}_{(N+1)\times(N+1)}$$

and

$$F_2 = \frac{1}{2h_x}\begin{bmatrix} \frac{-17}{3} & 3 & 3 & \frac{-1}{3} & \dots & 0 & 0 & 0 & 0 \\ \frac{-3}{2} & 0 & \frac{3}{2} & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \frac{-3}{2} & 0 & \frac{3}{2} & 0 & \dots & & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & 0 & \dots & 0 & \frac{-3}{2} & 0 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{3} & -3 & -3 & \frac{17}{3} \end{bmatrix}_{(N+1)\times(N+1)}$$

Here,

$$U' = [u'_1, u'_2, u'_3, ..., u'_N, u'_{N+1}]^T \qquad \text{and} \qquad U = [u_1, u_2, u_3, ..., u_N, u_{N+1}].$$

Clearly, we can approximate the first order spatial derivatives at each grid point by considering the matrix multiplication $F_1^{-1}F_2(U)$, resulting in vector U'.

In a similar manner, the matrix system approximating the second order spatial derivatives using the standard fourth order compact finite difference scheme as in Li and Chen (2008) can be written as:

$$F_3U'' = F_4U, \tag{11}$$

where the matrices F_3 and F_4 are given as

$$F_3 = \begin{bmatrix} 1 & 11 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1/10 & 1 & 1/10 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1/10 & 1 & 1/10 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1/10 & 1 & 1/10 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1/10 & 1 & 1/10 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 11 & 1 \end{bmatrix}_{(N+1)\times(N+1)}$$

and

$$F_4 = \frac{1}{h_x^2}\begin{bmatrix} 13 & -27 & 15 & -1 & \dots & 0 & 0 & 0 & 0 \\ \frac{6}{5} & \frac{-12}{5} & \frac{6}{5} & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \frac{6}{5} & \frac{-12}{5} & \frac{6}{5} & 0 & \dots & & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & 0 & \dots & 0 & \frac{6}{5} & \frac{-12}{5} & \frac{6}{5} \\ 0 & 0 & 0 & 0 & \dots & -1 & 15 & -27 & 13 \end{bmatrix}_{(N+1)\times(N+1)}$$

Here,

$$U'' = [u''_1, u''_2, u''_3, ..., u''_N, u''_{N+1}]^T$$

is the vector approximating the second derivatives at all the grid points which can be obtained by computing $F_3^{-1}F_4(U)$.

Using the above discretizations for the first- and second-order derivatives appearing in (1), we get a semi-discretized form of that PDE as a system of first order ODEs. Our idea is to use the above described compact finite difference schemes for approximating the spatial derivatives and then solve the obtained system of ODEs using the proposed block method (we will name this strategy as CFDBM). Since the proposed block method obtained is of implicit nature, therefore we need the solution values of the previous block as the starting values to approximate the solution at next block. So, an iterative strategy as decsribed in Amat and Busquier (2017), Petkovic et al. (2013) has to be used. For this purpose, we have used the FindRoot command in Mathematica system.

5 Test problems

In this section, we will use the CFDBM described above to solve two well-known nonlinear problems, viz., the FitzHugh–Nagumo equation and the Burgers' equation. Furthermore, the stability of the resulting differential systems has also been addressed.

5.1 FitzHugh–Nagumo equation

The FitzHugh–Nagumo equation is a well-known reaction-diffusion equation of non-linear nature physically significant in the fields of genetics, biology, heat and mass transfer, nuclear reactor theory and many other branches (Ramos et al. 2022). Its mathematical model can be written as

$$u_t = u_{xx} + u(1-u)(u-\nu), \tag{11a}$$

along with the initial condition

$$u(x,0) = g(x); \qquad a \le x \le b, \tag{11b}$$

and two boundary conditions as:

$$u(a,t) = g_1(t) = u_1(t) \quad \text{and} \quad u(b,t) = g_2(t) = u_{N+1}(t), \qquad t \ge 0. \tag{11c}$$

Here, ν represents a parameter which monitors the overall dynamics of the equation. As a first step, the spatial derivatives appearing in (11a) will be approximated using the fourth order compact finite difference schemes discussed in Lele (1992), Tyler (2007). Then, we will have $u_{xx} \approx F_3^{-1}F_4(U)$. After semi-discretization, the system of first order ODEs obtained from (11a)–(11c) can be written as

$$\begin{bmatrix} u_1' \\ u_2' \\ \dots \\ u_N' \\ u_{N+1}' \end{bmatrix} = (F_3^{-1}F_4 - \nu I)\begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_N \\ u_{N+1} \end{bmatrix} + (1+\nu)\begin{bmatrix} u_1^2 \\ u_2^2 \\ \dots \\ u_N^2 \\ u_{N+1}^2 \end{bmatrix} - \begin{bmatrix} u_1^3 \\ u_2^3 \\ \dots \\ u_N^3 \\ u_{N+1}^3 \end{bmatrix}.$$

So, the ODEs obtained from above system can be compactly written as:

$$U' = AU + B, \tag{12}$$

where $A = (F_3^{-1}F_4 - \nu I)$ is a $(N+1) \times (N+1)$ matrix and B is a $(N+1) \times 1$ row vector containing the remaining non-linear terms.

5.2 Burgers' equation

The Burgers' equation is a well-known non-linear PDE having its applications in fluid dynamics, statistical physics and many other areas. Its one-dimensional form can be written as:

$$u_t + uu_x = \nu u_{xx}, \tag{12a}$$

along with the initial condition

$$u(x,0) = g(x); \qquad a \le x \le b, \tag{12b}$$

and two boundary conditions as:

$$u(a,t) = g_1(t) = u_1(t) \quad \text{and} \quad u(b,t) = g_2(t) = u_{N+1}(t), \quad t \ge 0. \tag{12c}$$

Here, x represents the space variable, t represents the time variable, ν is the kinematic viscosity and u gives the velocity of the fluid. Similarly, as in the FitzHugh–Nagumo equation, we will have $u_x = F_1^{-1}F_2(U)$ and $u_{xx} = F_3^{-1}F_4(U)$. After semi-discretization, the system of first order ODEs obtained from (12a)–(12c) can be written as

$$\begin{bmatrix} u_1' \\ u_2' \\ \dots \\ u_N' \\ u_{N+1}' \end{bmatrix} = \nu(F_3^{-1}F_4)\begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_N \\ u_{N+1} \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_N \\ u_{N+1} \end{bmatrix} \circ (F_1^{-1}F_2)\begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_N \\ u_{N+1} \end{bmatrix}.$$

Here, $''\circ''$ represents the Hadamard (element-wise) product of two matrices of same dimensions. The above system can be written in a more compact form as

$$U' = AU + B, \tag{13}$$

where $A = \nu(F_3^{-1}F_4)$ is a $(N+1)\times(N+1)$ matrix and B contains the remaining non-linear part.

5.3 Stability of the differential system

In the above subsections, we have discussed the semi-discretization process considering the standard fourth order compact finite difference schemes and obtained a first order differential system in the time independent variable, on which the proposed block method can be applied to obtain a numerical solution. This section addresses the stability of the differential system obtained in both cases, using a similar approach as in Ramos et al. (2022). The implementation of compact finite difference schemes to the considered PDEs results in a system of ODEs of the form

$$U' = AU + B, \tag{14}$$

where, A is a $(N+1)\times(N+1)$ matrix and B is a $(N+1)\times 1$ vector containing non-homogeneous parts. The matrices for the considered PDEs can be written as

$$A = (A_2 - \nu I)$$

for the FitzHugh–Nagumo equation, and

$$A = \nu A_2$$

for the Burgers' equation where the matrix A_2 is given by: $A_2 = F_3^{-1}F_4$.

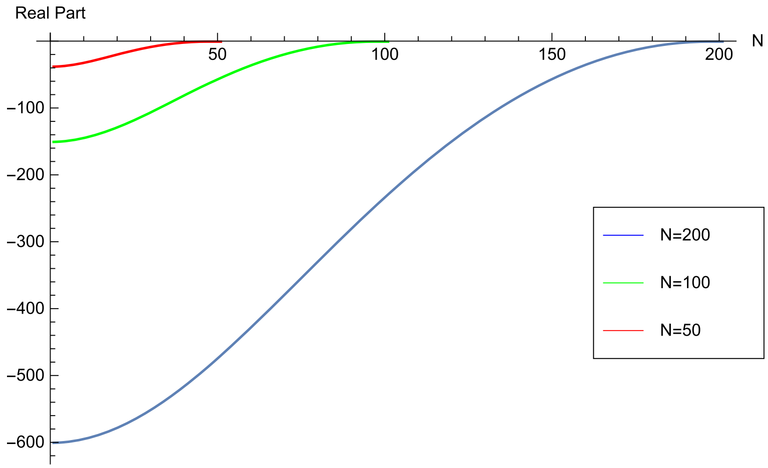

Fig. 2 Real parts of eigenvalues v/s N for FitzHugh–Nagumo system taking $\nu = 0.75$

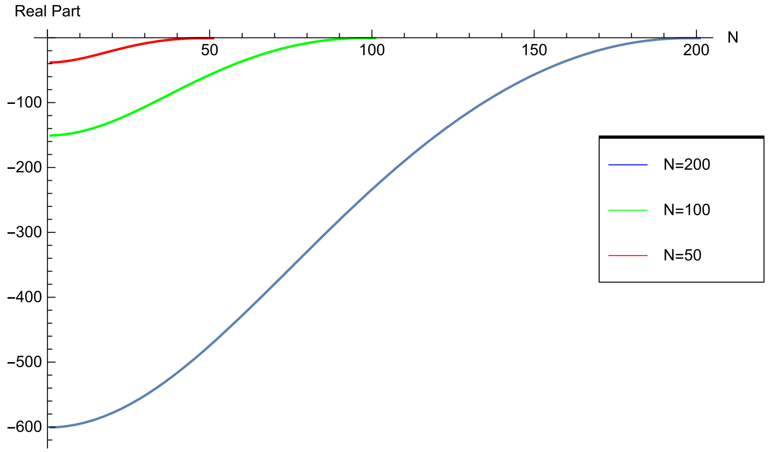

Fig. 3 Real parts of eigenvalues v/s N for Burgers' system taking $\nu = 0.01$

To investigate the stability of the differential system given by (14), we will linearize the non-linear terms of the considered PDEs by assuming the constant value of $u(x, t) = U_i^j$ for (x, t). The stability of the resulting linear differential system will imply the stability of the non-linear differential system as discussed in Ramos et al. (2022). The eigenvalues of matrix A will determine the stability characteristics. The differential system is said to be stable if all the eigenvalues have either negative or zero real part. This fact has been verified for the two considered differential systems for different values of N (number of spatial grid points).

In Figs. 2 and 3, the real parts of the eigenvalues obtained from the differential systems of the discretized PDEs have been plotted against N. These plots show that the real parts of the eigenvalues so obtained are all negative, thereby making the differential system stable in both cases.

6 Numerical experiments

This section addresses the performance of the proposed CFDBM, by considering some numerical experiments. Comparisons with other methods in the existing literature have been carried out. We have used Wolfram Mathematica 11.0 on a personal computer with 1.70 GHz Intel i3 processor. The computations of L_∞ and L_{rms} errors are calculated by using the usual formulas (Erdogan et al. 2020; Jain et al. 2016)

$$L_\infty = \max_{1\le i\le N+1} |e_i| \quad ;$$
$$L_{rms} = \left(\sum_{i=1}^{N+1} \frac{e_i^2}{N+1}\right)^{1/2},$$

where

$$e_i = u(x_i, t) - U(x_i, t),$$

for a specific value of time t, and $u(x_i, t)$, $U(x_i, t)$ represent the analytical and numerical solutions at point (x_i, t) respectively.

6.1 Non-linear FitzHugh–Nagumo equation

6.1.1 Example 1

Consider the test problem given by (11a) taking $\nu = 0.75$ along with the initial condition as:

$$u(x, 0) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{1}{2\sqrt{2}}x\right), \qquad -10 \le x \le 10.$$

The analytical solution of this problem is given as:

$$u(x, t) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{1}{2\sqrt{2}}x - \frac{(2\nu-1)}{4}t\right),$$

and the boundary conditions are obtained from the exact solution.

In Table 1, we have compared the L_∞ error norm for various values of N at time $t = 0.2$ using CFDBM with some existing data from Akkoyunlu (2019). We have achieved almost same or even better accuracy in the numerical approximation only in a single application with the proposed scheme whereas the scheme given in Akkoyunlu (2019) has attained similar accuracies after 20 time steps. Therefore, smaller errors have been obtained in fewer iterations using CFDBM, thus saving computational effort.

It is clear that for this problem CFDBM performs better than the technique in Akkoyunlu (2019). Further, to justify the superior performance of CFDBM over some existing techniques, we have compared the results for this problem using CFDBM with the results from Ahmad et al. (2019), Jiwari et al. (2014) and Ramos et al. (2022) for $N = 100$ and $\nu = 0.01$ at different values of time. The number of time iterations used with CFDBM is the same as used by Ramos et al. with OHBCM in Ramos et al. (2022). Tables 2 and 3 clearly show that CFDBM is the most efficient among all the compared schemes by providing much smaller L_∞ and L_{rms} errors. It is clear that CFDBM performs better than all other compared techniques for all values of time.

Table 1 Comparison of L_∞ error for Example 1 at $t = 0.2$

N	L_∞ (CFDBM)	L_∞ (method in Akkoyunlu (2019))
12	1.7169×10^{-4}	3.9857×10^{-4}
24	1.7119×10^{-5}	2.3475×10^{-5}
48	1.0244×10^{-6}	8.3749×10^{-6}
64	3.2892×10^{-7}	5.9363×10^{-6}
Number of iterations	1	20

6.1.2 Example 2

Consider the test problem given by (11a) taking $\nu = 0.5$ along with the initial condition given in Inan et al. (2020) as:

$$u(x,0) = \frac{1}{1+\exp(\frac{-x}{\sqrt{2}})}, \qquad 0 \le x \le 1.$$

The analytical solution of this problem is:

$$u(x,t) = \frac{1}{1+\exp(\frac{-s}{\sqrt{2}})}, \quad t > 0,$$

where $s = x + ct$ and $c = \sqrt{2}\left(\frac{1}{2} - \nu\right)$.

In Table 4, absolute errors computed using CFDBM for some of the grid points from the domain have been compared with some existing data from Inan et al. (2020). The errors obtained using CFDBM are smaller than those provided by the schemes in Inan et al. (2020) named ExpFDM and ANM.

6.2 A stiff case of the FitzHugh–Nagumo equation

Consider the stiff case of test problem of type (11a) as given in Agbavon and Appadu (2020) written below:

$$u_t = u_{xx} + \beta u(1-u)(u-\gamma), \tag{15}$$

where $\gamma \in (0,1)$ characterizes the overall dynamics of the equation and $\beta > 0$ represents the natural growth rate. Consider the problem along with the initial condition given as:

$$u(x,0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\sqrt{\beta}}{2\sqrt{2}}x\right), \qquad -10 \le x \le 10.$$

The analytical solution of this problem is given as:

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\sqrt{\beta}}{2\sqrt{2}}(x-ct)\right), \tag{16}$$

where, $c = -\sqrt{\frac{\beta}{2}}(2\gamma - 1)$.

Here, we have considered a stiff case of the FitzHugh–Nagumo equation. It has been observed that for large values of β (say $\beta = 5$ to 10), the problem becomes stiff and is difficult to solve it with some numerical schemes like the one in Agbavon and Appadu (2020). Firstly,

Table 2 Comparison of L_∞ error for Example 1 for $\nu = 0.01$ and $N = 100$

t	L_∞ error Ahmad et al. (2019)	L_∞ error Jiwari et al. (2014)	L_∞ error OHBCM (Ramos et al. 2022)	L_∞ error CFDBM
0.2	1.8896×10^{-5}	4.7416×10^{-5}	1.8876×10^{-5}	5.4996×10^{-8}
0.5	4.1554×10^{-5}	1.2312×10^{-4}	4.1519×10^{-5}	1.0653×10^{-7}
1	6.9891×10^{-5}	2.6261×10^{-4}	6.9734×10^{-5}	1.5822×10^{-7}
1.5	9.1687×10^{-5}	4.2096×10^{-4}	9.1180×10^{-5}	1.9190×10^{-7}
2.0	1.0969×10^{-4}	5.9999×10^{-4}	1.0854×10^{-4}	2.1710×10^{-7}
3.0	1.3942×10^{-4}	1.0324×10^{-3}	1.3651×10^{-4}	2.5517×10^{-7}
5.0	1.8964×10^{-4}	2.3050×10^{-3}	1.8000×10^{-4}	3.1252×10^{-7}

Table 3 Comparison of L_{rms} error for Example 1 for $\nu = 0.01$ and $N = 100$

t	L_{rms} error Ahmad et al. (2019)	L_{rms} error Jiwari et al. (2014)	L_{rms} error OHBCM (Ramos et al. 2022)	L_{rms} error CFDBM
0.2	2.1960×10^{-7}	1.5880×10^{-5}	7.4559×10^{-6}	1.8498×10^{-8}
0.5	1.5696×10^{-6}	3.8433×10^{-5}	1.6411×10^{-5}	3.6476×10^{-8}
1	7.1449×10^{-6}	8.1870×10^{-5}	2.7433×10^{-5}	5.3790×10^{-8}
1.5	1.7262×10^{-5}	1.3387×10^{-4}	3.5345×10^{-5}	6.3739×10^{-8}
2.0	3.1857×10^{-5}	1.9433×10^{-4}	4.1285×10^{-5}	7.0226×10^{-8}
3.0	7.2878×10^{-5}	3.4320×10^{-4}	4.9731×10^{-5}	7.8748×10^{-8}
5.0	1.8803×10^{-3}	7.8638×10^{-4}	6.1345×10^{-5}	9.1738×10^{-8}

Table 4 Comparison of absolute errors for Example 2 at time $t = 0.04$ for $N = 10$

x	CFDBM	ExpFDM (Inan et al. 2020)	ANM (Inan et al. 2020)
0.2	1.72×10^{-8}	3.00×10^{-6}	2.00×10^{-7}
0.4	4.63×10^{-9}	1.00×10^{-5}	5.00×10^{-7}
0.6	1.63×10^{-9}	2.00×10^{-5}	7.00×10^{-7}
0.8	7.26×10^{-9}	4.00×10^{-5}	6.00×10^{-7}
Number of iterations	1	8	8

in Table 5, we have compared the numerical results obtained using CFDBM with various finite difference schemes from Agbavon and Appadu (2020) for different values of β and have obtained comparatively more accurate results using a larger time stepsize than the other compared schemes as shown in the table.

In this problem, our objective is not to show the comparison between errors for small values of β, but to highlight the fact that for larger values of β (say 5–10), the proposed scheme still works well. Table 6 lists the L_∞ error for different values of N at time $t = 0.5$ and for two large values of $\beta = 5$ and 10, taking $h = 0.16667$. It has been observed that even for the larger values of β, the proposed scheme has resulted in considerably smaller errors. Hence, the proposed scheme is a good alternative to handle such stiff problems.

6.3 Dynamical consistency of the proposed numerical scheme for the FitzHugh–Nagumo equation

The dynamical consistency of a numerical scheme refers to the replicated behavior of a numerical solution as that of the analytical solution. Approximate solutions are expected to physically behave in the same way that exact solutions do. It has been observed that the analytical solution given by (16) of the FitzHugh–Nagumo equation (15) has a non-negative and bounded solution (Agbavon and Appadu 2020). In this section, we have checked the boundedness and non-negativity of the approximate solutions obtained using CFDBM, by solving the problem for different values of $\gamma = 0.2, 0.4, 0.6, 0.8, 0.99$ at time $t = 0.5$ with $N = 100$ in Fig. 4.

6.4 Non-linear Burgers' equation

6.4.1 Example 1

Consider the test problem given by (12a), along with the initial condition considered as

$$u(x, 0) = \sin(\pi x); \qquad 0 \le x \le 1,$$

and boundary conditions are as

$$u(0, t) = u(1, t) = 0; \qquad t \ge 0.$$

The exact solution of the problem is given

Table 5 Comparison of L_∞ error with Agbavon and Appadu (2020) at time $t = 0.5$ for $\gamma = 0.2$ and $N = 100$

β	Scheme 1	Scheme 2	Scheme 3	Scheme 4	CFDBM
0.5	2.6767×10^{-3}	1.1164×10^{-5}	4.4063×10^{-6}	4.1322×10^{-6}	1.6401×10^{-8}
1	7.4715×10^{-3}	5.0919×10^{-5}	2.5881×10^{-5}	2.4227×10^{-5}	1.0738×10^{-7}
2	2.0671×10^{-2}	2.3146×10^{-4}	1.6408×10^{-4}	1.5759×10^{-4}	6.4360×10^{-7}
Time step size (h)	0.00125	0.00125	0.00125	0.00125	0.16667

Table 6 Error norms at time $t = 0.5$, $\gamma = 0.2$ and for large values of β

β	N	L_∞ error
5	20	3.8486×10^{-3}
	40	2.3864×10^{-4}
	100	5.9992×10^{-6}
10	20	1.6529×10^{-2}
	40	1.3899×10^{-3}
	100	3.3252×10^{-5}

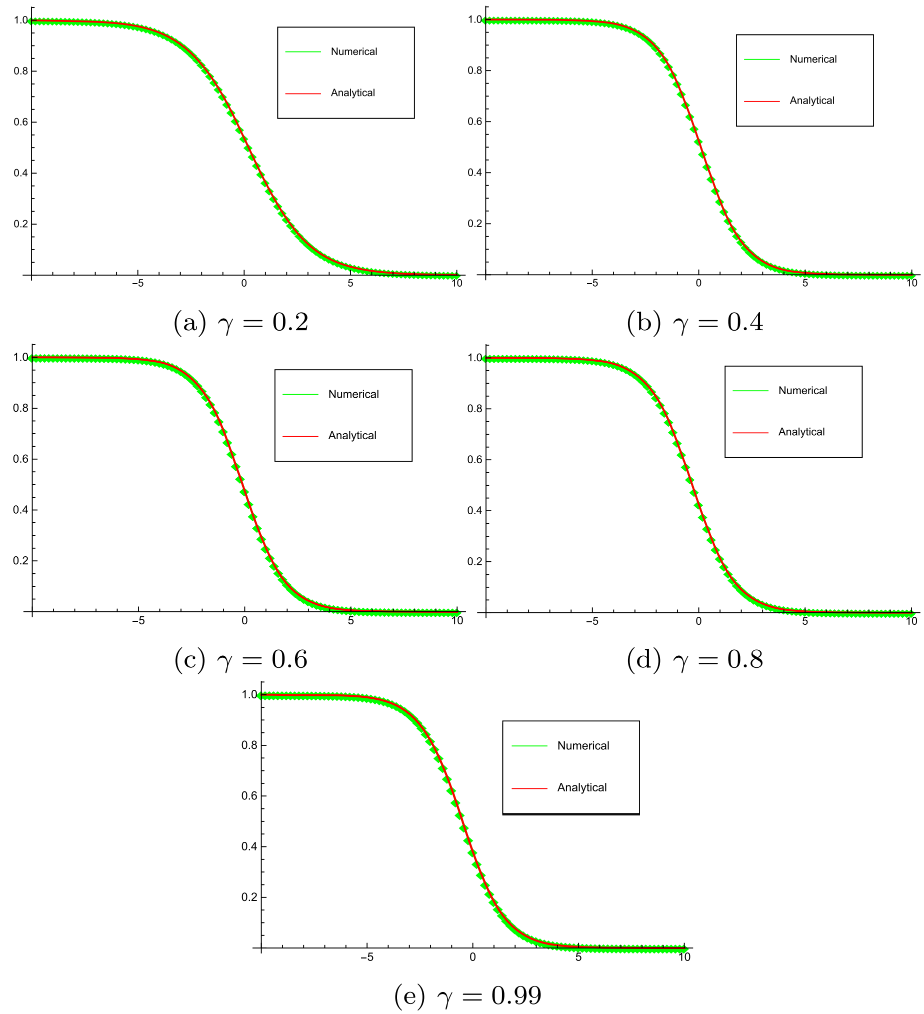

Fig. 4 Graphical representations of numerical and analytical solution of problem (15) for various values of γ

$$u(x,t) = 2\pi\nu \frac{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\nu t) n \sin(n\pi x)}{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\nu t) n \cos(n\pi x)},$$

where

$$c_0 = \int_0^1 \exp\left(\frac{-1}{2\pi\nu}(1-\cos(\pi x))\right) dx,$$
$$c_n = 2\int_0^1 \exp\left(\frac{-1}{2\pi\nu}(1-\cos(\pi x))\right) \cos(n\pi x) dx,$$
$$(n = 1, 2, 3, ...).$$

For this example, the performance of CFDBM has been compared with some of the existing techniques. In Table 7, the data has been listed after solving the problem for different values of time t, taking $N = 80$ as in Ramos et al. (2022). The approximate solutions computed are compared with the approximate solutions provided by Kadalbajoo and Awasthi (2006), Kutulay et al. (2004), Jiwari (2015), Seydaoglu (2018), Ramos et al. (2022) for $\nu = 0.01$ and $N = 80$ at different grid points and various values of time. We have used the same number of time iterations as is used by Ramos et al. (2022), whereas the other methods used larger number of time iterations.

A similar kind of comparison has been made in Table 8 for different values of time t, taking $\nu = 0.1$. The approximate solutions computed using CFDBM are compared with the approximate solutions computed by Kadalbajoo and Awasthi (2006), Kutulay et al. (2004), Jiwari (2015), Özis et al. (2003) for $N = 80$. The last row of the tables list the absolute value of the maximum error of each of the column, showing that the proposed method (CFDBM) is the most competent among all the compared methods.

6.4.2 Example 2

Consider one more test problem on Burgers' equation given by (12a) as in Kadalbajoo and Awasthi (2006), along with the initial condition considered as

$$u(x,0) = 4x(1-x); \qquad 0 \le x \le 1.$$

and boundary conditions as

$$u(0,t) = u(1,t) = 0; \qquad t \ge 0.$$

The exact solution of the problem is given by

$$u(x,t) = 2\pi\nu \frac{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\nu t) n \sin(n\pi x)}{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\nu t) n \cos(n\pi x)},$$

where

$$c_0 = \int_0^1 \exp\left(\frac{-1}{3\nu}(3x^2 - 2x^3)\right) dx,$$
$$c_n = 2\int_0^1 \exp\left(\frac{-1}{3\nu}(3x^2 - 2x^3)\right) \cos(n\pi x) dx,$$
$$(n = 1, 2, 3, ...).$$

Table 7 Comparison of Numerical and Exact solution for Example 1 at different times for $\nu = 0.01$

x	t	Kadalbajoo and Awasthi (2006)	Kutulay et al. (2004)	Jiwari (2015)	Seydaoglu (2018)	OHBCM (Ramos et al. 2022)	CFDBM	Exact
0.25	0.4	0.34229	0.34819	0.34191	0.34187	0.34192	0.34191	0.34191
	0.6	0.26902	0.27536	0.26896	0.26894	0.26897	0.26896	0.26896
	1	0.18817	0.19375	0.18820	0.18818	0.18819	0.18819	0.18819
	3	0.07511	0.07754	0.07511	0.07511	0.07511	0.07511	0.07511
0.5	0.4	0.66797	0.66543	0.66069	0.66065	0.66071	0.66071	0.66071
	0.6	0.53211	0.53525	0.52942	0.52937	0.52942	0.52941	0.52942
	1	0.37500	0.38047	0.37443	0.37439	0.37442	0.37442	0.37442
	3	0.15018	0.15362	0.15019	0.15017	0.15018	0.15017	0.15018
0.75	0.4	0.93680	0.91201	0.91203	0.91032	0.91027	0.91026	0.91026
	0.6	0.77724	0.77132	0.76723	0.76721	0.76725	0.76724	0.76724
	1	0.55833	0.56157	0.55606	0.55601	0.55605	0.55605	0.55605
	3	0.22485	0.22874	0.22486	0.22485	0.22483	0.22481	0.22481
Max Abs Error		0.02654	0.0064	0.00005	0.00006	0.00002	0.00001	

Table 8 Comparison of numerical and exact solution for Example 1 at different times for $\nu = 0.1$

x	t	Kadalbajoo and Awasthi (2006)	Kutulay et al. (2004)	Jiwari (2015)	Özis et al. (2003)	CFDBM	Exact
0.25	0.4	0.30881	0.31215	0.30889	0.31420	0.30889	0.30889
	0.6	0.24069	0.24360	0.24075	0.24368	0.24074	0.24074
	1	0.16254	0.16473	0.16258	0.16391	0.16256	0.16256
	3	0.02720	0.02771	0.02720	0.02742	0.02720	0.02720
0.5	0.4	0.56955	0.57293	0.56963	0.57629	0.56963	0.56963
	0.6	0.44714	0.45088	0.44724	0.45164	0.44720	0.44721
	1	0.29188	0.29532	0.29195	0.29437	0.29192	0.29192
	3	0.04021	0.04097	0.04021	0.04051	0.04020	0.04021
0.75	0.4	0.62540	0.63038	0.62537	0.62603	0.62544	0.62544
	0.6	0.48715	0.49268	0.48718	0.49039	0.48721	0.48721
	1	0.28744	0.29204	0.28747	0.29019	0.28747	0.28747
	3	0.02978	0.03038	0.02977	0.03000	0.02977	0.02977
Max Abs error		0.00008	0.00547	0.00007	0.00531	0.00001	

Table 9 Comparison of numerical and exact solution for Example 2 at different times for $\nu = 0.01$

x	t	Kadalbajoo and Awasthi (2006)	Kutulay et al. (2004)	Jiwari (2015)	Seydaoglu (2018)	OHBCM (Ramos et al. 2022)	CFDBM	Exact
0.25	0.4	0.36273	0.36911	0.36225	0.36215	0.36226	0.36225	0.36226
	0.6	0.28212	0.28905	0.28204	0.28196	0.26203	0.26203	0.28204
	1	0.19467	0.20069	0.19469	0.19465	0.19469	0.19469	0.19469
	3	0.07613	0.07865	0.07613	0.07613	0.07613	0.07613	0.07613
0.5	0.4	0.69186	0.68818	0.68364	0.68357	0.68368	0.68367	0.68368
	0.6	0.55125	0.55425	0.54831	0.54821	0.54832	0.54831	0.54832
	1	0.38627	0.39206	0.38568	0.38561	0.38567	0.38567	0.38568
	3	0.15218	0.15576	0.15219	0.15217	0.15218	0.15218	0.15217
0.75	0.4	0.94940	0.92194	0.92044	0.92051	0.92051	0.92050	0.92050
	0.6	0.79399	0.78676	0.78297	0.78292	0.78300	0.78299	0.78299
	1	0.57170	0.57491	0.56932	0.56924	0.56932	0.56931	0.56932
	3	0.22778	0.23183	0.22779	0.22777	0.22776	0.22774	0.22774
Max Abs Error		0.0289	0.00701	0.00006	0.00011	0.00002	0.00001	

Table 10 Comparison of numerical and exact solution for Example 2 at different times for $\nu = 0.1$

x	t	Kadalbajoo and Awasthi (2006)	Kutulay et al. (2004)	Jiwari (2015)	Özis et al. (2003)	CFDBM	Exact
0.25	0.4	0.31743	0.32091	0.31752	0.32678	0.31752	0.31752
	0.6	0.24609	0.24910	0.24615	0.25118	0.24614	0.24614
	1	0.16558	0.16782	0.16561	0.16780	0.16560	0.16560
	3	0.02776	0.02828	0.02776	0.02814	0.02776	0.02775
0.5	0.4	0.58446	0.58788	0.58454	0.59660	0.58454	0.58454
	0.6	0.45791	0.46174	0.45800	0.46580	0.45798	0.45798
	1	0.29831	0.30183	0.29838	0.30255	0.29834	0.29834
	3	0.04107	0.04185	0.04107	0.04161	0.04106	0.04106
0.75	0.4	0.64558	0.65054	0.64556	0.64691	0.64562	0.64562
	0.6	0.50261	0.50825	0.50265	0.50858	0.50268	0.50268
	1	0.29582	0.30057	0.29585	0.30067	0.29586	0.29586
	3	0.03044	0.03106	0.03044	0.03084	0.03044	0.03044
Max Abs Error		0.00015	0.00557	0.00006	0.01206	0.00001	

The numerical results computed using CFDBM have been compared with some of the existing data, similarly as done with the previous example. In Tables 9 and 10, the data for the considered example has been listed after solving it using CFDBM with $N = 80$ at $\nu = 0.01$ and $\nu = 0.1$ respectively and the results are then compared with some of the existing techniques. CFDBM has shown superior performance than the other techniques.

7 Conclusions

The proposed block method in combination with the standard fourth order compact finite difference scheme is a new interesting approach to effectively solve considered type of non-linear PDEs. The proposed block method is consistent and $\mathcal{A}$-stable. The differential system obtained after implementation of compact finite differences to considered problems also turns to be stable for various values of step sizes. Two test problems have been solved for different parameters and it has been observed that the computed results are in good agreement with the exact solutions. Moreover, the method has given accurate results for a stiff case of the FitzHugh–Nagumo problem, where the corresponding compared technique has not worked. The computed results have also been compared with some of the existing data and it has been observed that better accuracy can be achieved in a very few time steps in comparison to some compared existing schemes. Overall, the combined proposed method is a novel, accurate, consistent and computationally time saving numerical scheme and can be considered as a good alternative to solve time dependent non-linear PDEs of the considered type. In future, its applicablility could be extended for solving second order differential systems arising in compact discretizations of second order time dependent PDEs and may be formulated in a variable step-size mode.

Acknowledgements We would like to thank the anonymous reviewers for their valuable comments that have greatly contributed to improve the quality of the manuscript. Akansha Mehta would like to thank I. K. Gujral Punjab Technical University Jalandhar, Punjab (India) for providing research facilities for the present work.

Data Availability No data is associated with this article.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Adam Y (1975) A Hermitian finite difference method for the solution of parabolic equations. J Comput Phys 1:393–406

Agbavon KM, Appadu AR (2020) Construction and analysis of some nonstandard finite difference methods for the Fitz Hugh–Nagumo equation. Numer Methods Partial Differ Equ 36(5):1–25

Ahmad I, Ahsan M, Din ZU, Masood A, Kumam P (2019) An efficient local formulation for time-dependent PDEs. Mathematics 7(3):216

Akkoyunlu C (2019) Compact finite differences method for Fitz Hugh–Nagumo equation. Univ J Math Appl 2(4):180–187

Amat S, Busquier S (2017) After notes on Chebyshev's iterative method. Appl Math Nonlinear Sci 2(1):1–2

Collatz L (1966) The numerical treatment of differential equations. Springer, Berlin

Debnath L (2012) Nonlinear partial differential equations for scientists and engineers. Birkhauser, Basel

Erdogan L, Sakar MG, Saldir O (2020) A finite difference method on layer-adapted mesh for singularly perturbed delay differential equations. Appl Math Nonlinear Sci 5(1):425–436

Hairer E, Wanner G (1996) Solving ordinary differential equation-II: stiff and differential-algebraic problems. Springer, Berlin
Inan B, Ali KK, Saha A, Ak T (2020) Analytical and numerical solutions of the Fitz Hugh–Nagumo equation and their multistability behavior. Numer Methods Partial Differ Equ 37:7–23
Jain MK, Iyengar SRK, Jain RK (2016) Computational methods for partial differential equations. New Age International Publishers, New Delhi
Jiwari R (2015) A Hybrid numerical scheme for the numerical solution of the Burgers' equation. Comput Phys Commun 188:59–67
Jiwari R, Gupta R, Kumar V (2014) Polynomial differential quadrature method for numerical solutions of the generalized Fitz Hugh–Nagumo equation with time-dependent coefficients. Ain Shams Eng J 5(4):1343–1350
Kadalbajoo MK, Awasthi A (2006) A numerical method based on Crank–Nicolson scheme for Burgers' equation. Appl Math Comput 182:1430–1442
Kutulay S, Esan A, Dag I (2004) Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method. J Comput Appl Math 167(1):21–33
Lambert JD (1973) Computational methods in ordinary differential equations. Introductory mathematics for scientists and engineers. Wiley, New York
Lele SK (1992) Compact finite difference schemes with spectral-like resolution. J Comput Phys 103(1):16–42
Li J, Chen Y (2008) Computational partial differential equations using MATLAB. A Chapman and Hall Book, London
Li J, Visbal MR (2006) High-order compact schemes for nonlinear dispersive waves. J Sci Comput 26:1–23
Milne WE (1953) Numerical solution of differential equations. Wiley, New York
Mittal RC, Jain RK (2012) Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. Appl Math Comput 218:7839–7855
Özis T, Aksan EN, Özdes A (2003) A finite element approach for solution of Burgers' equation. Appl Math Comput 139(1):417–428
Petkovic MS, Neta B, Petkovic LD, Dzunic J (2013) Multipoint methods for solving nonlinear equations. Elsevier, Amsterdam
Ramos H, Singh G (2017) A tenth order A—stable two-step hybrid block method for solving initial value problems of ODEs. Appl Math Comput 310:75–88
Ramos H, Kaur A, Kanwar V (2022) Using a cubic B-spline method in conjunction with a one-step optimized hybrid block approach to solve nonlinear partial differential equations. Comput Appl Math 41(34):1–28
Seydaoglu M (2018) An accurate approximation algorithm for Burgers' equation in the presence of small viscosity. J Comput Appl Math 344:473–481
Shampine LF, Watts HA (1969) Block implicit one-step methods. Math Comput 23:731–740
Singh G, Ramos H (2018) An optimized two-step hybrid block method formulated in variable step-size mode for integrating y"=f(x, y, y') numerically. Numer Math Theor Methods Appl 12(2):640–660
Tyler GJ (2007) Analysis and implementation of high-order compact finite difference schemes. MSc. thesis, Brigham Young University Provo
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
Zhang P, Wang J (2012) A predictor corrector compact finite difference scheme for Burgers' equation. Appl Math Comput 219(3):892–898