

RESEARCH

Open Access



On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term

K.M. Agbavon¹, A.R. Appadu^{1*} and M. Khumalo²

*Correspondence:

rao.appadu@up.ac.za;
biprao2@yahoo.com

¹Department of Mathematics and Applied Mathematics, University of Pretoria, Pretoria, South Africa
Full list of author information is available at the end of the article

Abstract

Li et al. (*SIAM J. Sci. Comput.* 20:719–738, 1998) used the moving mesh partial differential equation (MMPDE) to solve a scaled Fisher's equation and the initial condition consisting of an exponential function. The results obtained are not accurate because MMPDE is based on a familiar arc-length or curvature monitor function. Qiu and Sloan (*J. Comput. Phys.* 146:726–746, 1998) constructed a suitable monitor function called modified monitor function and used it with the moving mesh differential algebraic equation (MMDAE) method to solve the same problem of scaled Fisher's equation and obtained better results.

In this work, we use the forward in time central space (FTCS) scheme and the nonstandard finite difference (NSFD) scheme, and we find that the temporal step size must be very small to obtain accurate results. This causes the computational time to be long if the domain is large. We use two techniques to modify these two schemes either by introducing artificial viscosity or using the approach of Ruxun et al. (*Int. J. Numer. Methods Fluids* 31:523–533, 1999). These techniques are efficient and give accurate results with a larger temporal step size. We prove that these four methods are consistent for partial differential equations, and we also obtain the region of stability.

Keywords: Fisher's equation; Moving mesh method; FTCS; NSFD; Artificial viscosity

1 Introduction

Real-life problems are mainly modeled by partial differential equations (PDEs) with applications to engineering, physics, chemistry, ecology, biology, and other related fields of science. PDEs can be of different forms:

- (i) linear or nonlinear,
- (ii) homogeneous or nonhomogeneous,
- (iii) elliptic, hyperbolic, or parabolic.

PDEs have some specifications that give the information how smooth the solution is, how rapid information propagates, and what is the impact of initial and boundary conditions (which help to find if a particular approach is suitable to the problem being portrayed by the PDEs). Some examples of modeling real-life problems can be found in [4–7]. Indeed,

a new model and solution method for wave propagation in compressible two-phase flow problem were proposed by Zeidan et al. [4]. The model consists of six equations and is applicable for pure fluid and fluid mixtures. A modern shock-capturing method (total-variation diminishing (TVD)) slope limiter centered scheme (SLIC)) has been proposed to solve the problems considered in simple way and with good accuracy. The modeling of a two-phase gas–magma mixture was made using total-variation diminishing (TVD) slope limiter centered scheme (SLIC) and the model is based on a nonhomogeneous system of nonlinear hyperbolic conservation laws [5]. There is strong evidence that the model and the method used are accurate, robust, and conservative. The study of unsteady cavitation in liquid hydrogen flows was made in the context of compressible two-phase one-fluid inviscid solver by Goncalvés and Zeidan [6]. Three conservation laws for mixture mass, mixture momentum, and total energy alongside with gas volume fraction transport equation with thermodynamic effect were used. Minhajul et al. [7] investigated the interaction of weak shocks for widely used isentropic drift-flux equations of two-phase flows. The existence and uniqueness condition for elementary waves was obtained.

Our study is based on reaction–diffusion equations of the form of PDEs, which are mostly used in modeling transport of air, adsorption of pollutants in soil, diffusion of neutrons, food processing, modeling of biological and ecological systems, modeling of semiconductors, oil reservoir flow transport, among others [8]. Some tangible applications are modeling amazing patterns and phenomena such as tree–grass interactions in fire-prone savannas [9], pulse splitting and shedding (the Gray–Scott equation; see [10]). The Gray–Scott equation has some applications, namely, reaction and competition in excitable systems, autocatalysis, reaction between two chemical species with different diffusivities [11], modeling labyrinthine patterns [12], which are formed in models of catalytic reactions. There are only a few cases where analytical solutions to such reaction–diffusion equations exist, and therefore we need to construct accurate and efficient numerical methods.

In this work, our interest is in Fisher’s equation [13], which describes spontaneous growth and spread of a dominant gene. Fisher considers a population that is distributed linearly in an habitat (shore line) with uniform density. If the mutation happens at any point of the habitat, then the mutant gene is expected to increase at the risk of the allelomorphs previously occupying the same position. This occurrence will be first terminated in a neighborhood of the mutation and later in the adjacent portion of its range. Assuming the range to be long enough in comparison with the distance separating the locations of offspring from those of their parents, there will be from the origin a wave of increase in the gene frequency.

1.1 Background of Fisher’s equation

We consider Fisher’s equation [13]

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

where $x \in (-\infty, +\infty), t > 0$, and the boundary and initial conditions are

$$\lim_{x \rightarrow -\infty} u(x, t) = 1, \quad \lim_{x \rightarrow +\infty} u(x, t) = 0, \tag{2}$$

$$u(x, 0) = u_0(x). \tag{3}$$

This problem [13] was solved by Kolmogorov et al. [14] by introducing the concepts of traveling waves and the existence of wave speed c . Moreover, they showed that the propagation speed c of the waves is greater than two ($c \geq 2$) if the initial condition $u_0(x)$ is in the interval $[0, 1]$ and the type of solution is $u(x, t) = v(\xi)$, where $\xi = x - ct$ satisfying $u \in [0, 1]$ for all ξ . They also proved that such solutions do not exist for $c \in [0, 1)$. The studies in [15] showed that all positive initial data, $u_0(x)$, decaying at least exponentially as $x \rightarrow \infty$ evolves to a unique travelling wave. If

$$u_0(x) \sim e^{-\beta} \quad \text{as } x \rightarrow \infty, \tag{4}$$

then the solution u evolves a traveling wave speed, which is a function of β , where

$$c(\beta) = \begin{cases} \beta + \frac{1}{\beta}, & \beta \leq 1, \\ 2, & \beta \geq 1. \end{cases} \tag{5}$$

Furthermore, they proved that if the initial amplitude drops sufficiently quickly as x goes to infinity, then the propagation speed of the wave (which determines the behavior of initial condition) has the minimum value, $c = 2$.

The numerical implementation of Eq. (1) with boundary and initial conditions given, respectively, by (2) and (3) involving the traveling wave solution is challenging due to the dependence of sensitive solution on the initial data behavior at infinity. For instance, problem (1) with initial condition (3) (Cauchy problem) is replaced by an initial and boundary value problem on the finite spatial domain $[x_l, x_r]$. Moreover, Gazdag and Canosa [16] resolved this issue by imposing an asymptotic representation of the boundary condition (2) at $x = x_l, x = x_r$. They found that the solution draws toward a traveling wave of the minimum speed $c = 2$. They concluded that the demanding time to change to the minimum wave speed profile is linked to the right-hand cutoff point $x = x_r$. The same approach was done by Hagstrom et al. [15] with the wave speed greater than the minimum wave speed $c = 2$. They showed that the traveling wave solutions can be interpreted in finite domain by constructing accurately the asymptotic boundary conditions at $x = x_l$ and $x = x_r$. They obtained good results with $u(x_l, t) = 1$ and $u(x_r, t) = 0$ for $t \geq 0$.

Many authors like Canosa [17] and Hagstrom and Keller [15] have worked on the issue of stability and sensitivity of the solution to the boundary of traveling wave. For instance, the equilibrium solutions $u = 0$ and $u = 1$ of Eq. (1) are, respectively, unstable and stable to small perturbations. Moreover, they demonstrated that all traveling waves are stable to small perturbations of compact support but unstable to those of infinite support.

In 2005, Anguelov et al. [18] solved the same problem (Eq. (1)) by using a periodic initial condition with θ -nonstandard method. They concluded that their method is elementarily stable in the limit case of space-independent variable, stable with respect to the boundedness and positivity property, and finally stable with respect to the conservation of energy in the stationary case.

2 Organization of paper

The paper is organized as follows. In Sect. 3, we describe in detail how the moving mesh method with monitor function is implemented. In Sect. 4, we describe numerical experiments considered as in [1, 2]. In Sects. 5 and 6, the forward in time central space (FTCS)

and FTCS- ϵ difference scheme are studied, and numerical results are displayed. Sections 7 and 8 are devoted to derivation and properties of NSFD and NSFD- ϵ schemes, and results are presented. In Sect. 9, we add artificial viscosity to both FTCS and NSFD and study properties of new schemes and present some results. In Sect. 10, we highlight the salient features of this paper. All simulations are performed using MATLAB R2014a software on an Intel core2 as CPU.

3 Moving mesh method

Li et al. [1] have considered the scaled Fisher equation

$$u_t = u_{xx} + \rho u(1 - u), \tag{6}$$

where $x \in (-\infty, +\infty)$, $t > 0$, and ρ is a positive large constant. The boundary condition and initial condition are given by Eqs. (2) and (3), respectively. The exact solution to this problem is

$$u(x, t) = \left[1 + \exp\left(\sqrt{\frac{\rho}{6}}x - \frac{5\rho}{6}t\right) \right]^{-2}, \tag{7}$$

with wave speed $c = 5\sqrt{\rho/6}$ and the minimum wave speed $c = 2\sqrt{\rho}$. Li et al. [1] used the method called the moving mesh partial differential equation (MMPDE). They obtained poor results when ρ was chosen to be 10^4 and concluded that MMPDE is not suitable for reaction–diffusion equation (in particular, Fisher’s equation) when the reaction term is much greater than the diffusion term with initial condition consisting of an exponential function. This is due to the fact that MMPDE is based on familiar arc-length or curvature monitor function and does not produce accurate results [19]. Qiu et al. [2] improved the results of Li et al. [1] by constructing a specific monitor function and used the method of moving mesh differential algebraic equation (MMDAE).

The technique of the MMPDE method has been utilized broadly over the last few years to find a solution to time-dependent partial differential equations (PDEs). The method consists of moving the mesh points as time change with motion designed to minimize some measurement in computational error [2].

We consider the variables ζ and t with ζ defined by a one-to-one coordinate transformation of the form

$$x = x(\zeta, t), \quad \zeta_i = 1 + \frac{2i}{N}, \quad i = 0, \dots, N, \tag{8}$$

where ζ_i are spaced nodes in the interval $[-1, 1]$ to the nodes $\{x_i\}_{i=0}^N$ in the interval $[x_l, x_r]$, with

$$x_l = x_0(t) < x_1(t) < \dots < x_N(t) = x_r, \quad \forall t \geq 0.$$

We can rewrite (6) in a semidiscrete form such that

$$\dot{u}_i - \dot{x}_i \frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} = \frac{2}{x_{i+1} - x_{i-1}} \left(\frac{u_{i+1} - u_i}{x_{i+1} - x_i} - \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \right) + \rho u_i(1 - u_i) \tag{9}$$

for $i = 1, 2, \dots, N - 1$ by using the Lagrangian form [19]

$$\dot{u} - \dot{x} \partial u_x = \partial u_{xx} + \rho u(1 - u). \tag{10}$$

Moreover \dot{u}, \dot{x} are the derivatives respect to t , independent of ζ , and $\{x_i\}_{i=0}^N$ and $\{u_i\}_{i=0}^N$ are the time-dependent vectors for approximations. To adjust the mesh to the solution as presented in [20], they introduced the equidistribution principle

$$\int_{x_l}^{x(\zeta, t)} M(s, t) ds = \zeta \int_{x_l}^{x(x_r)} M(s, t) ds, \tag{11}$$

where $M > 0$ indicates the monitor function that has to be equally distributed between the nodes x_l, x_r . Differentiation of Eq. (11) with respect to ζ gives

$$\partial_\zeta [M(x(\zeta, t)) \partial_\zeta x(\zeta, t)] = 0. \tag{12}$$

Furthermore, Eq. (12) has been used in [20] to derive a collection of moving meshes, and the most accurate of this collection is [21]

$$\partial_\zeta \zeta \dot{x} = -\frac{1}{\tau} \partial_\zeta (M \partial_\zeta x), \tag{13}$$

denoted by MMPDE6 with small positive parameter $\tau \ll 1$. Under the condition that the discretization has been done on the grid ζ_i and using the second-order central differences leads to a semidiscrete form of moving mesh equation:

$$\dot{x}_{i-1} - 2\dot{x}_i + \dot{x}_{i+1} = -\frac{1}{\tau} [M'_{i+1/2}(x_{i+1} - x_i) - M'_{i+1/2}(x_i - x_{i-1})] \tag{14}$$

for $i = 1, 2, \dots, N - 1$ with $M'_{i+\frac{1}{2}}$ being a smoothed monitor function given in [19, 20] by

$$M'_{i+\frac{1}{2}} = \frac{\sum_{k=i-p}^{i+p} M_{k+1/2}^2 (\frac{q}{q+1})^{|k-i|}}{\sum_{k=i-p}^{i+p} (\frac{q}{q+1})^{|k-i|}}, \tag{15}$$

where q is a positive real number, and p is a nonnegative integer. Furthermore, setting

$$M'_{i+1/2}(x_{i+1} - x_i) - M'_{i+1/2}(x_i - x_{i-1}) = 0 \tag{16}$$

in Eq. (14) leads to the moving mesh differential-algebraic equation (MMDAE) developed by Mulholland et al. [19]. This method combines systems (9) and (16). The difference between MMPDE6 and MMDAE is that MMPDE6 accommodates the parameter τ , which shows the time used to attain equidistribution from some initial state, whereas MMDAE enforces the approximate equidistribution condition (16) at each moment of time in the time discretization.

Each problem has its own choice of a monitor function. This makes the choice of a monitor function an open question. Following [2, 19, 20], the monitor function (arc-length) is defined by

$$M(x, t) = \sqrt{1 + \alpha^2 (\partial_x u)^2} \tag{17}$$

with its discrete approximation $M_{i+1/2}$ being

$$M_{i+\frac{1}{2}} = \sqrt{1 + \alpha^2 \left(\frac{u_{i+1} - u_i}{x_{i+1} - x_i} \right)^2}, \tag{18}$$

where the parameter α measures the amplitude to which the solution slope has control over the mesh location.

It has been shown in [2] that moving mesh based on the arc-length and curvature monitor function is not convenient for the computational solution of Eq. (6). Indeed, firstly, the computed solution at $t = 2.5 \times 10^{-3}$ is susceptible to the choice of τ (at values 10^{-3} , 10^{-5} , 10^{-7} with α fixed at 2 and $x_l = -0.2$, $x_r = 0.8$) in Eq. (13) in the moving mesh using the arc-length monitor function. Secondly, in the common monitor function utilized, the first derivative in (17) is substituted by the second derivative, and we have

$$M(x, t) = (1 + \alpha^2 (\partial_{xx} u)^2)^{1/4} \tag{19}$$

with its discrete approximation

$$M_{i+\frac{1}{2}}^4 = 1 + \alpha^2 \left[\frac{1}{x_{i+1} - x_i} \left(\frac{u_{i+2} - u_i}{x_{i+2} - x_i} - \frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} \right) \right]^2. \tag{20}$$

The results show the same sensitivity as in the case of arc-length monitor function. There is oscillation of the solution at the front of wave. In the quest for obtaining the accurate result, Qiu et al. [2] introduced the modified monitor function.

3.1 Modified monitor function

The modified monitor function is constructed to give a great nodal density and hence a better accuracy at the wave front. It has been shown by Hagstrom and Keller [15] and Gazdag and Canosa [16] that the difficulties that occur in simulating numerically the traveling waves for Fisher’s equation come from the front of the wave. This is why significant care should be taken in formulation of boundary conditions at $x = x_r$. Furthermore, the results in [16] showed that the numerical solution of all traveling wave are stable to small disturbances with compact support and unstable with infinite extent especially to truncation errors inserted at the wave front. Consequently, it is an origin of inaccuracy of truncations errors rather than similar truncations errors introduced at the back of the wave. In this regard, the modified monitor function is

$$M(x, t) = [1 + \alpha^2(1 - u)^2 + \beta^2(a - u)^2(u_{xx})^2]^{1/2}, \tag{21}$$

where α , β , and a are real specific carefully chosen parameters. The expressions $(1 - u)^2$ and $(a - u)^2$ are designed to give more influence of the curvature region at the front of the wave than that of the corresponding curvature region at the back of the wave.

With $\alpha = 1.5$, $\beta = 0.1$, $a = 1.015$, and $t = 2.5 \times 10^{-3}$ in the computations of MMDAE and the modified monitor function given by Eq. (21), the maximum pointwise error 9.25×10^{-3} is much smaller than the corresponding error $O(1)$ using the arc-length monitor function. With the method of MMPDE6, the situation is less improved than with the method of MMDAE, which is also far better than the method of arc-length monitor function. Indeed,

Table 1 Computation of L_1 and L_∞ errors using MMPDE and MMDAE methods with $\rho = 10^4, \alpha = 1.5, \beta = 0.1, a = 1.015, \tau = 10^{-7}, N = 50$, at time $t = 2.5 \times 10^{-3}$

Methods	L_1 error	L_∞ error	CPU
MMPDE	$O(1)$	4.29×10^{-2}	ext
MMDAE	9.25×10^{-3}	$k \times 10^{-2}$	0.86ext

with the parameter $\tau = 10^{-7}$ and time $t = 2.5 \times 10^{-3}$, the L_∞ error is 4.29×10^{-2} . For the values greater than $N = 50$ with $\tau = 10^{-7}$ and time $t = 2.5 \times 10^{-3}$, the error is not diminished. Whenever the reduction is applied to the value of τ , we have the reduction in the L_∞ error. L_1 and L_∞ errors for MMPDE and MMDAE are displayed in Table 1.

4 Numerical experiments

We consider two problems. Firstly, we consider the same problem as in Qiu et al. [2], which involves solving the following:

Problem 1

$$u_t = u_{xx} + 10^4 u(1 - u)$$

for $x \in [-0.2, 0.8]$ with boundary conditions $\lim_{x \rightarrow -\infty} u(x, t) = 1$ and $\lim_{x \rightarrow +\infty} u(x, t) = 0$ and time 2.5×10^{-3} .

The initial condition is

$$u(x, 0) = \left[1 + \exp\left(\sqrt{\frac{\rho}{6}} x\right) \right]^{-2}. \tag{22}$$

Secondly, we consider a slight modification of Problem 1. We use a larger domain with the same boundary and initial conditions and the same propagation time.

Problem 2

$$u_t = u_{xx} + 10^4 u(1 - u)$$

for $x \in [-10, 90]$ with boundary conditions $\lim_{x \rightarrow -\infty} u(x, t) = 1$ and $\lim_{x \rightarrow +\infty} u(x, t) = 0$ and time 2.5×10^{-3} .

The initial condition is

$$u(x, 0) = \left[1 + \exp\left(\sqrt{\frac{\rho}{6}} x\right) \right]^{-2}. \tag{23}$$

In the next sections, we present the numerical methods used and study the properties.

5 Forward in time central space (FTCS)

The forward in time central space (FTCS) scheme, when used to discretize Eq. (6), gives [22]

$$\frac{u_m^{n+1} - u_m^n}{k} = \frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{h^2} + \rho u_m^n (1 - u_m^n). \tag{24}$$

A single expression for the FTCS scheme is

$$u_m^{n+1} = (1 - 2R)u_m^n + k\rho u_m^n(1 - u_m^n) + R(u_{m+1}^n + u_{m-1}^n), \tag{25}$$

where $R = \frac{k}{h^2}$. The time-step size and spatial mesh are denoted by k and h , respectively.

5.1 Stability

Equation (6) is nonlinear, and hence Fourier series stability analysis cannot be applied directly. We need to freeze the coefficients before applying von Neumann stability analysis [23]. Taha and Ablowitz [24] obtained the stability of a method proposed by Zabusky and Krustal [25] for Korteweg–de Vries (KdV) equation using the method of freezing coefficients and von Neumann stability analysis. The scheme derived by Zabusky and Kuskal for the KdV equation $u_t + 6uu_x + u_{xxx} = 0$ is

$$\begin{aligned} &\frac{u_m^{n+1} - u_m^{n-1}}{2k} + 6\left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3}\right)\left(\frac{u_{m+1}^n - u_{m-1}^n}{2h}\right) \\ &+ \frac{1}{2h^3}(u_{m+2}^n - 2u_{m+1}^n + 2u_{m-1}^n - u_{m-2}^n) = 0. \end{aligned} \tag{26}$$

To obtain stability, Taha and Ablowitz [24] express uu_x as $u_{\max}u_x$ and use the ansatz $u_m^n = \xi^n e^{lmw}$, where w is the phase angle. They obtain the following equation:

$$\frac{\xi - \xi^{-1}}{2k} + \frac{6|u_{\max}|}{h} I \sin(w) + \frac{1}{2h^3}(e^{2lw} - 2e^{lw} + 2e^{-lw} - e^{-2lw}) = 0,$$

which can be rewritten as

$$\xi = \xi^{-1} - \frac{12k|u_{\max}|}{h} I \sin(w) - \frac{k}{h^3}(e^{2lw} - 2e^{lw} + 2e^{-lw} - e^{-2lw}).$$

The linear stability requirement is

$$\frac{k}{h} \left| \frac{1}{h^2} - 2|u_{\max}| \right| \leq \frac{2}{3\sqrt{3}}. \tag{27}$$

Appadu et al. [26] used the method of freezing coefficient and von Neumann stability analysis to obtain the region of stability of some schemes for Eq. (6). We use the same idea to obtain the stability region of the FTCS scheme. We rewrite Eq. (25) as

$$u_m^{n+1} = \left(1 - \frac{2k}{h^2}\right)u_m^n + \frac{k}{h^2}(u_{m+1}^n + u_{m-1}^n) + k\rho u_m^n - (k\rho u_m^n)|u_{\max}|, \tag{28}$$

where u_{\max} is a frozen coefficient. It follows by using Fourier series analysis for Eq. (28) that the amplification factor is given by

$$\xi = 1 + \frac{2k}{h^2}(\cos(w) - 1) + k\rho(1 - |u_{\max}|). \tag{29}$$

In our numerical experiment, $u_{\max} = 1$ and $\rho = 10^4$. Hence we obtain

$$\xi = 1 - \frac{4k}{h^2} \sin^2\left(\frac{w}{2}\right). \tag{30}$$

For stability, we must have $|\xi| \leq 1$ for $w \in [-\pi, \pi]$, and therefore

$$-1 \leq 1 - \frac{4k}{h^2} \sin^2\left(\frac{w}{2}\right) \leq 1, \tag{31}$$

which leads to

$$k \leq \frac{h^2}{2}. \tag{32}$$

For $h = 0.01$, we obtain $k \leq 5 \times 10^{-5}$. The time of the experiment is $T_{\max} = 2.5 \times 10^{-3}$, and for the stability, the temporal step size is less than or equal to 5×10^{-5} or $T_{\max}/50$.

For the accuracy order of FTCS, we use the Taylor series expansion about point (n, m) of (25):

$$\begin{aligned} & u + ku_t + \frac{k^2}{2}u_{tt} + \frac{k^3}{6}u_{ttt} + O(k^4) \\ &= \left(1 - \frac{2k}{h^2} + k\rho\right)u - k\rho u^2 \\ & \quad + \frac{k}{h^2}\left(u + hu_x + \frac{h^2}{2}u_{xx} + \frac{h^3}{6}u_{xxx} + \frac{h^4}{24}u_{xxxx} + O(h^5)\right) \\ & \quad + \frac{k}{h^2}\left(u - hu_x + \frac{h^2}{2}u_{xx} - \frac{h^3}{6}u_{xxx} + \frac{h^4}{24}u_{xxxx} + O(h^5)\right), \end{aligned} \tag{33}$$

which gives

$$u_t - u_{xx} - \rho u(1 - u) = -\frac{k}{2}u_{tt} - \frac{k^2}{6}u_{ttt} + \frac{h^2}{12}u_{xxxx} + O(k^4) + O(h^5). \tag{34}$$

The FTCS scheme has the first-order accuracy in time and the second-order accuracy in space.

5.2 Numerical results using FTCS

We tabulate the L_1 and L_∞ errors and display CPU times when Problems 1 and 2 are solved using FTCS at some different values of time-step size with spatial step size $h = 0.01$. The errors are displayed in Tables 2 and 3.

We observe from Tables 2 and 3 that the L_1 and L_∞ errors are almost the same with different computational times, which was expected since as we increase the length of the domain, the computational time increases. As we decrease the time-step size, the L_1 and L_∞ errors initially decrease and reach minimum when $k \simeq T_{\max}/1300$, and then the errors increase again. For k close to $T_{\max}/50$, the dispersion error is quite large. Comparing Tables 2 and 3 to Table 1, we notice that the L_1 and L_∞ errors from the FTCS method at an optimal temporal step size are quite smaller than the L_1 and L_∞ errors from MMPDE and MMDAE methods. Some plots of u against x are depicted in Fig. 1 using $h = 0.01$ and some different values of k .

6 FTCS- ϵ scheme

In this section, we modify the FTCS scheme to obtain FTCS- ϵ . Ruxun et al. [3] have designed a new scheme by modifying Lax–Wendroff (LW) scheme used to discretize the

Table 2 L_1 and L_∞ errors and CPU time at some different values of time-step size k for Problem 1 with $\rho = 10^4$ at time 2.5×10^{-3} with spatial mesh size $h = 0.01$ using FTCS scheme, where $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/52$	2.8968×10^{-1}	1.5200	0.648
$T_{\max}/100$	6.5144×10^{-2}	6.9877×10^{-1}	0.719
$T_{\max}/200$	3.2980×10^{-2}	3.9927×10^{-1}	0.672
$T_{\max}/300$	2.0727×10^{-2}	2.5613×10^{-1}	0.693
$T_{\max}/400$	1.4262×10^{-2}	1.7830×10^{-1}	0.708
$T_{\max}/500$	1.0268×10^{-2}	1.2943×10^{-1}	0.730
$T_{\max}/600$	7.5557×10^{-3}	9.5605×10^{-2}	0.770
$T_{\max}/700$	5.5929×10^{-3}	7.1038×10^{-2}	0.791
$T_{\max}/800$	4.1068×10^{-3}	5.2473×10^{-2}	0.816
$T_{\max}/900$	2.9425×10^{-3}	3.7988×10^{-2}	0.866
$T_{\max}/1000$	2.0059×10^{-3}	2.6389×10^{-2}	0.904
$T_{\max}/1100$	1.2404×10^{-3}	1.6901×10^{-2}	0.932
$T_{\max}/1200$	6.2983×10^{-4}	9.0013×10^{-3}	1.233
$T_{\max}/1300$	1.9347×10^{-4}	2.3240×10^{-3}	1.055
$T_{\max}/1400$	4.2509×10^{-4}	4.3458×10^{-3}	1.141
$T_{\max}/1500$	8.3336×10^{-4}	8.7691×10^{-3}	1.245
$T_{\max}/1600$	1.1913×10^{-3}	1.3138×10^{-2}	1.366

Table 3 L_1 and L_∞ errors and CPU time at some different values of time-step size k for Problem 2 with $\rho = 10^4$ at time 2.5×10^{-3} with spatial mesh size $h = 0.01$ using FTCS scheme, where $T_{\max} = 2.5 \times 10^{-3}$

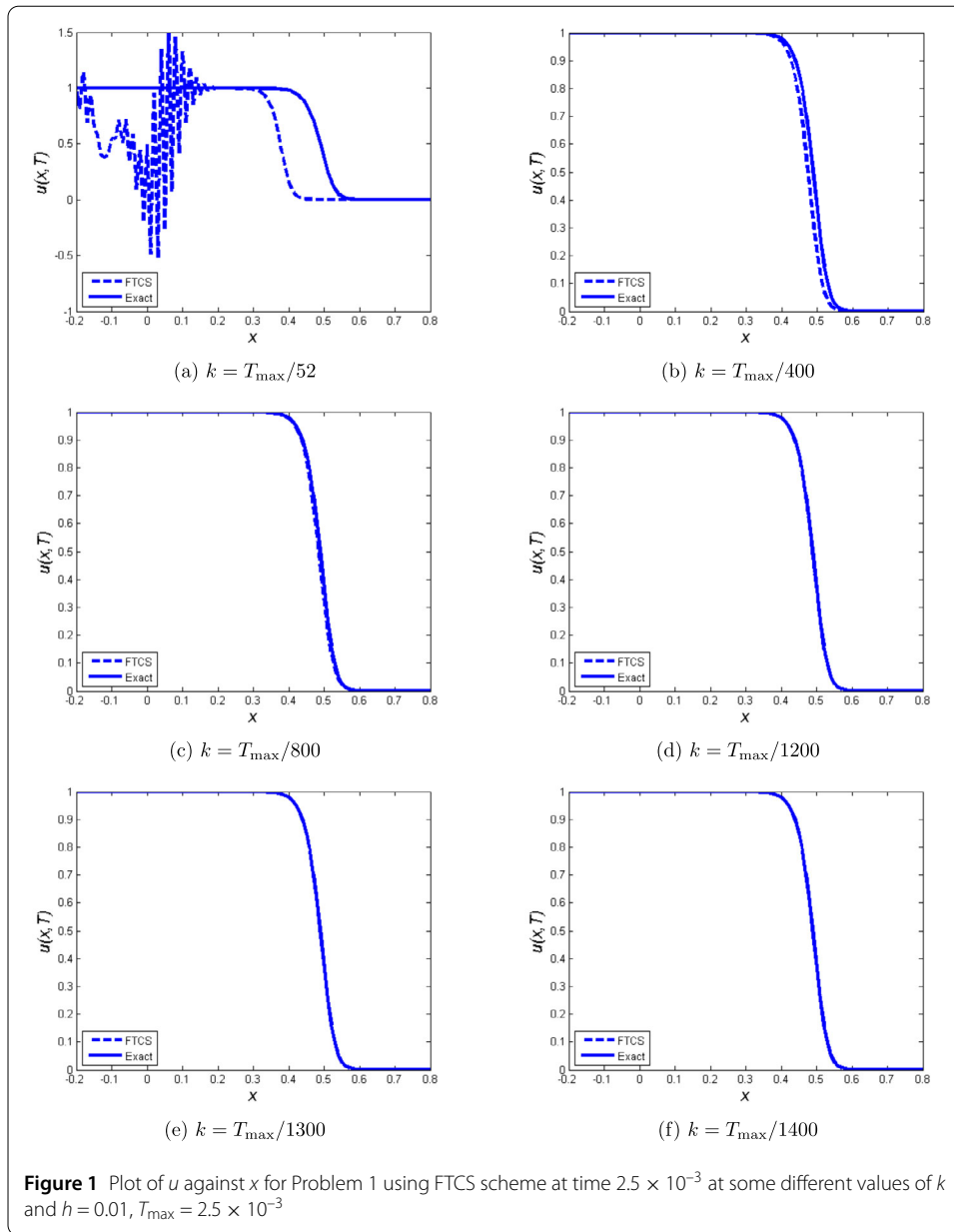
Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/52$	2.8667×10^{-1}	1.5040	1.368
$T_{\max}/100$	6.5144×10^{-2}	6.9877×10^{-1}	2.183
$T_{\max}/200$	3.2980×10^{-2}	3.9927×10^{-1}	4.532
$T_{\max}/300$	2.0727×10^{-2}	2.5613×10^{-1}	7.689
$T_{\max}/400$	1.4262×10^{-2}	1.7830×10^{-1}	11.509
$T_{\max}/500$	1.0268×10^{-2}	1.2943×10^{-1}	16.099
$T_{\max}/600$	7.5557×10^{-3}	9.5605×10^{-2}	22.270
$T_{\max}/700$	5.5929×10^{-3}	7.1038×10^{-2}	28.343
$T_{\max}/800$	4.1068×10^{-3}	5.2473×10^{-2}	35.383
$T_{\max}/900$	2.9425×10^{-3}	3.7988×10^{-2}	43.181
$T_{\max}/1000$	2.0059×10^{-3}	2.6389×10^{-2}	51.644
$T_{\max}/1100$	1.2404×10^{-3}	1.6901×10^{-2}	61.532
$T_{\max}/1200$	6.2983×10^{-4}	9.0013×10^{-3}	71.195
$T_{\max}/1300$	1.9347×10^{-4}	2.3240×10^{-3}	82.456
$T_{\max}/1400$	4.2509×10^{-4}	4.3458×10^{-3}	94.146
$T_{\max}/1500$	8.3336×10^{-4}	8.7691×10^{-3}	107.725
$T_{\max}/1600$	1.1913×10^{-3}	1.3138×10^{-2}	118.352

linear advection equation

$$u_t + cu_x = 0, \quad c > 0. \tag{35}$$

We briefly describe their approach. To solve Eq. (35), a simple explicit consistent scheme can be constructed:

$$u_m^{n+1} = a_1 u_{m+1}^n + a_0 u_m^n + a_{-1} u_{m-1}^n. \tag{36}$$



The Taylor series expansion of (36) gives

$$\left(1 + \Delta t D_t + \frac{\Delta t^2}{2!} D_t^2 + \dots\right) u_i^n = \left[(a_1 + a_0 + a_{-1}) + \Delta x (a_1 - a_{-1}) D_x + \frac{\Delta x^2}{2!} (a_1 + a_{-1}) D_x^2 + \frac{\Delta x^3}{3!} (a_1 - a_{-1}) D_x^3 + \dots + u_i^n \right],$$

where $D_t = \partial/\partial t$ and $D_x = \partial/\partial x$. Ruxun et al. [3] arrive at the following theorem.

Theorem 6.1 Assume that the solution $u(x, t)$ of Eq. (35) is smooth enough and the scheme given by (36) is consistent with the original partial differential Eq. (35) and that the spatial

mesh size h is small enough. Then

$$\begin{aligned} a_1 + a_0 + a_{-1} &= 1, \\ a_1 - a_{-1} &= c \frac{k}{h}. \end{aligned} \tag{37}$$

The Lax–Wendroff (LW) scheme discretizing Eq. (35) is given by

$$u_m^{n+1} = \frac{1}{2}(r^2 - r)u_{m+1}^n + (1 - r^2)u_m^n + \frac{1}{2}(r^2 + r)u_{m-1}^n, \tag{38}$$

where $r = c \frac{k}{h}$. By the CFL condition the scheme is stable if $0 < r \leq 1$.

Clearly, LW is not monotonic and is not a positive scheme. A simple approach to construct a monotonic scheme is to reform the LW scheme. Ruxun et al. [3] constructed the LW- ϵ scheme given by

$$u_m^{n+1} = \left(\frac{r^2 - r}{2} + \epsilon_1 \right) u_{m+1}^n + (1 - r + \epsilon_2) u_m^n + \left(\frac{r^2 + r}{2} + \epsilon_3 \right) u_{m-1}^n. \tag{39}$$

For consistency, we set $\epsilon_1 + \epsilon_2 + \epsilon_3 = 0$ and $\epsilon_1 - \epsilon_3 = 0$. Hence $\epsilon_2 = -2\epsilon_1$. We let $\epsilon_2 = -\epsilon$. Therefore LW- ϵ scheme is

$$u_m^{n+1} = \left(\frac{r^2 - r}{2} + \frac{\epsilon}{2} \right) u_{m+1}^n + (1 - r - \epsilon) u_m^n + \left(\frac{r^2 + r}{2} + \frac{\epsilon}{2} \right) u_{m-1}^n \tag{40}$$

with $0 \leq \epsilon \ll 1$. By working with dissipation and dispersion remainders, they found that $\epsilon = 1/4$ gives rise to a positive monotonic scheme, which still has the second-order accuracy.

We attempt to derive the FTCS- ϵ scheme by adding numerical dissipation to the scheme to reduce numerical dispersion in the profile. We propose the following scheme:

$$\begin{aligned} u_m^{n+1} &= \left(1 - \frac{2k}{h^2} + k\rho \right) u_m^n + \frac{k}{h^2} u_{m+1}^n + \frac{k}{h^2} u_{m-1}^n - k\rho (u_m^n)^2 \\ &\quad + \epsilon_1 u_{m-1}^n + \epsilon_2 u_m^n + \epsilon_3 u_{m+1}^n. \end{aligned} \tag{41}$$

The Taylor series expansion about point (n, m) gives

$$\begin{aligned} &u + ku_t + \frac{k^2}{2} u_{tt} + \frac{k^3}{6} u_{ttt} + O(k^4) \\ &= \left(1 - \frac{2k}{h^2} + k\rho \right) u - k\rho u^2 \\ &\quad + \frac{k}{h^2} \left(u + hu_x + \frac{h^2}{2} u_{xx} + \frac{h^3}{6} u_{xxx} + O(h^4) \right) \\ &\quad + \frac{k}{h^2} \left(u - hu_x + \frac{h^2}{2} u_{xx} - \frac{h^3}{6} u_{xxx} + O(h^4) \right) \\ &\quad + \epsilon_1 \left(u - hu_x + \frac{h^2}{2} u_{xx} - \frac{h^3}{6} u_{xxx} + O(h^4) \right) \end{aligned}$$

$$\begin{aligned}
 & + \epsilon_3 \left(u + hu_x + \frac{h^2}{2}u_{xx} + \frac{h^3}{6}u_{xxx} + O(h^4) \right) \\
 & + \epsilon_2 u,
 \end{aligned} \tag{42}$$

which simplifies to

$$\begin{aligned}
 & u + ku_t + \frac{k^2}{2}u_{tt} + \frac{k^3}{6}u_{ttt} + O(k^4) \\
 & = (1 + k\rho + \epsilon_1 + \epsilon_2 + \epsilon_3)u + (-\epsilon_1 + \epsilon_3)hu_x - k\rho u^2 \\
 & + \left(k + \frac{\epsilon_1}{2}h^2 + \frac{\epsilon_3}{2}h^2 \right)u_{xx} + \left(-\frac{\epsilon_1}{6} + \frac{\epsilon_3}{6} \right)h^3u_{xxx} + O(h^4).
 \end{aligned} \tag{43}$$

On rearranging, we get

$$\begin{aligned}
 & u - (1 + \epsilon_1 + \epsilon_2 + \epsilon_3)u + ku_t - h(\epsilon_3 - \epsilon_1)u_x - ku_{xx} - k\rho u + k\rho u^2 \\
 & = -\frac{k^2}{2}u_{tt} - \frac{k^3}{6}u_{ttt} + \frac{h^3}{6}(\epsilon_3 - \epsilon_1)u_{xxx} + \frac{h^2}{2}(\epsilon_1 + \epsilon_3)u_{xx} + O(h^4) + O(k^4).
 \end{aligned} \tag{44}$$

We recall that we are solving Eq. (6). Therefore, for consistency, we must have $\epsilon_1 + \epsilon_2 + \epsilon_3 = 0$ and $\epsilon_3 - \epsilon_1 = 0$. We thus have $\epsilon_3 = \epsilon_1 = \epsilon$ and $\epsilon_2 = -2\epsilon_1 = -2\epsilon$. Hence the FTCS- ϵ scheme is given by

$$u_m^{n+1} = u_m^n + \frac{k}{h^2}(u_{m+1}^n - 2u_m^n + u_{m-1}^n) + k\rho u_m^n - k\rho(u_m^n)^2 + \epsilon(u_{m+1}^n - 2u_m^n + u_{m-1}^n), \tag{45}$$

and the scheme is consistent.

For order of accuracy, we consider Eq. (44), replace $\epsilon_3 = \epsilon_1 = \epsilon$ and $\epsilon_2 = -2\epsilon_1 = -2\epsilon$, and we have

$$ku_t - ku_{xx} - k\rho u + k\rho u^2 = -\frac{k^2}{2}u_{tt} - \frac{k^3}{6}u_{ttt} + h^2\epsilon u_{xx} + O(h^4) + O(k^4). \tag{46}$$

Dividing by k gives

$$u_t - u_{xx} - \rho u + \rho u^2 = -\frac{k}{2}u_{tt} - \frac{k^2}{6}u_{ttt} + \frac{h^2}{k}\epsilon u_{xx} + \frac{h^2}{12}u_{xxx} + O\left(\frac{h^4}{k}\right) + O(k^3). \tag{47}$$

The FTCS- ϵ scheme has the first-order accuracy both in time and in space.

For stability analysis, we apply Fourier series analysis, and we obtain the amplification factor

$$\xi = 1 - \frac{2k}{h^2} + \frac{2k}{h^2} \cos(w) + k\rho - k\rho u_{\max} + \epsilon(2 \cos(w) - 2). \tag{48}$$

We choose $u_{\max} = 1$, $\rho = 10^4$, and therefore,

$$\xi = 1 - 4 \sin^2\left(\frac{w}{2}\right) \left(\epsilon + \frac{k}{h^2} \right). \tag{49}$$

Table 4 L_1 and L_∞ errors and CPU time at some different values of time-step size k for Problem 1 and spatial mesh size $h = 0.01$, $\epsilon = 0.01$ using FTCS- ϵ , where $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/53$	2.8025×10^{-1}	1.5265	0.764
$T_{\max}/100$	5.9959×10^{-2}	6.6244×10^{-1}	0.783
$T_{\max}/200$	2.1273×10^{-2}	2.6296×10^{-1}	0.796
$T_{\max}/300$	2.4401×10^{-3}	3.1737×10^{-2}	0.928
$T_{\max}/310$	9.7448×10^{-4}	1.3521×10^{-2}	0.850
$T_{\max}/315$	3.2771×10^{-4}	4.7040×10^{-3}	0.799
$T_{\max}/320$	4.6757×10^{-4}	4.6218×10^{-3}	0.814
$T_{\max}/325$	1.1657×10^{-3}	1.2769×10^{-2}	0.902
$T_{\max}/350$	4.5061×10^{-3}	5.3608×10^{-2}	0.821
$T_{\max}/375$	7.6305×10^{-3}	9.1590×10^{-2}	0.927
$T_{\max}/400$	1.0576×10^{-2}	1.2682×10^{-1}	0.927
$T_{\max}/500$	2.1076×10^{-2}	2.4515×10^{-1}	0.925
$T_{\max}/600$	3.0241×10^{-2}	3.4502×10^{-1}	0.868

Table 5 L_1 and L_∞ errors and CPU time at some different values of time-step size k and spatial mesh size $h = 0.01$, $\epsilon = 0.01$ of Problem 2 using FTCS- ϵ , where $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/53$	2.8025×10^{-1}	1.5265	1.488
$T_{\max}/100$	5.9959×10^{-2}	6.6244×10^{-1}	2.265
$T_{\max}/200$	2.1273×10^{-2}	2.6296×10^{-1}	4.704
$T_{\max}/300$	2.4401×10^{-3}	3.1737×10^{-2}	7.819
$T_{\max}/310$	9.7448×10^{-4}	1.3521×10^{-2}	8.154
$T_{\max}/315$	3.2771×10^{-4}	4.7040×10^{-3}	8.429
$T_{\max}/320$	4.6757×10^{-4}	4.6218×10^{-3}	8.537
$T_{\max}/325$	1.1657×10^{-3}	1.2769×10^{-2}	8.732
$T_{\max}/350$	4.5061×10^{-3}	5.3608×10^{-2}	9.779
$T_{\max}/375$	7.6305×10^{-3}	9.1590×10^{-2}	10.678
$T_{\max}/400$	1.0576×10^{-2}	1.2682×10^{-1}	11.692
$T_{\max}/500$	2.1076×10^{-2}	2.4515×10^{-1}	16.319
$T_{\max}/600$	3.0241×10^{-2}	3.4502×10^{-1}	21.644

For stability, we take $|\xi| \leq 1$, which gives

$$2\left(\epsilon + \frac{k}{h^2}\right) \sin^2\left(\frac{w}{2}\right) \leq 1 \tag{50}$$

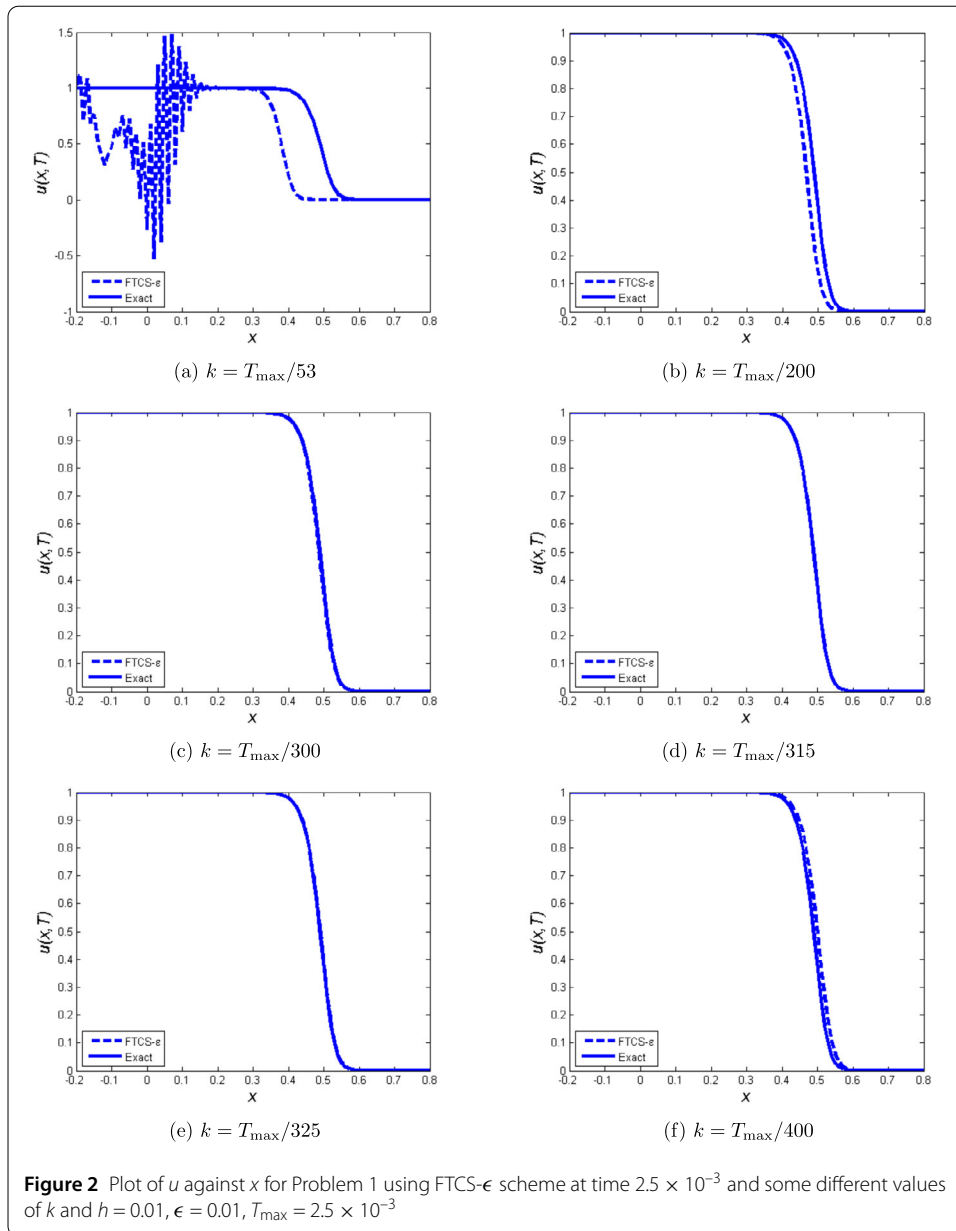
and finally yields

$$\frac{k}{h^2} \leq \frac{1}{2} - \epsilon. \tag{51}$$

For $h = 0.01$ and $\epsilon = 0.01$, we have the stability region given by $k \leq 4.90 \times 10^{-5}$.

We tabulate the L_1 and L_∞ errors and display CPU times for Problems 1 and 2 using FTCS- ϵ at some different values of time-step size with spatial step size $h = 0.01$ and $\epsilon = 0.01$. The errors are displayed in Tables 4 and 5. The time is $T_{\max} = 2.5 \times 10^{-3}$, and the time-step size is less than or equal to 4.90×10^{-5} . Plots of u vs x at time $T_{\max} = 2.5 \times 10^{-3}$ are displayed in Fig. 2.

We observe from Tables 4 and 5 that the L_1 and L_∞ errors are the same and the CPU times are different. As we increase the length of the domain, it is obvious that CPU time must increase. As we decrease the time-step size, the L_1 and L_∞ errors initially decrease



and reach the minimum when $k \simeq T_{\max}/315$, and then the errors increase again. For k close to $T_{\max}/100$, the dispersion error is quite large.

Comparing Tables 2 and 3 to Table 1, we notice that the L_1 and L_∞ errors from the FTCS- ϵ method at an optimal step size are quite smaller than those obtained using the MMPDE and MMDAE methods.

7 Nonstandard finite difference schemes (NSFD)

The nonstandard finite difference scheme (NSFD) has been widely used this past three decades. The concept of NSFD method has been introduced by Ronald Mickens [27] to approximate solutions of several partial differential models. The derivations are generally based on the notion of dynamical consistency [28]. The concept of dynamical consistency performs an important role in the construction of discrete models, which provide a sig-

nificant difficulty in the computation of numerical solutions. This difficulty is numerical instabilities. Indeed, numerical instabilities are solutions to discrete equations that do not link to any solutions of the original differential equations [29]. The dynamical consistency is defined with respect to peculiar properties of a physical system, which vary mostly from one system to another. These properties must preserve positivity, boundedness, monotonicity of the solutions, correct number and stability of fixed-points, and other special solutions. The importance of dynamical consistency is to be taken as general standard used to restrain the possible forms for constructing an NSFD scheme [28].

7.1 Construction of NSFD schemes

The NSFD schemes are especially based upon two fundamental principles:

- (1) Replacing the denominator of the discrete derivative by a more general function.
- (2) Nonlocal representation of both linear and nonlinear terms:

$$x \rightarrow 2x_k - x_{k+1}, \tag{52}$$

$$x^3 \rightarrow \left(\frac{x_{k+1} + x_{k-1}}{2} \right) x_k^2, \tag{53}$$

$$x^3 \rightarrow 2x_k^3 - x_k^2 x_{k+1}, \tag{54}$$

$$x^2 \rightarrow \left(\frac{x_{k+1} + x_k + x_{k-1}}{3} \right) x_k. \tag{55}$$

The selection of the functions $\phi(k)$ for time derivatives has no general rule. Nevertheless, particular forms for precise equation can be easily found. The common functions usually used in [29] are

$$\phi(k) = \frac{1 - e^{-\lambda k}}{\lambda}, \tag{56}$$

where λ is some parameter emerging in the differential equation. For partial differential equation, a suitable generalization can be made. For instance, in the nonlocalization, the nonlinear terms [29] are

$$[u(x, t)]^2 \rightarrow u_{m-1}^k u_m^{k+1}, \tag{57}$$

where $x \rightarrow x_m = (\Delta x)m$ and $t \rightarrow t_k = (\Delta t)k$. The utility and strength of NSFD procedures are that they do not need any a priori knowledge of the exact solutions to the differential equation. They come from the enforcement of certain physical system necessities on the discrete model equations as found by dynamical consistency. In conclusion, the lack of dynamical consistency leads to numerical instability. This practically appears for some values of the parameters or step-sizes [28].

Diverse explicit NSFD schemes have been suggested for Fisher’s equation with respect to their performances [18, 29, 30]. These performances are the stability of fixed points, positivity, boundedness of solutions, and so on. Following the idea of Mickens [31], a non-standard finite difference scheme for Eq. (6) is

$$\frac{u_m^{n+1} - u_m^n}{\phi(\Delta t)} - \frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{[\psi(\Delta h)]^2} = \rho u_m^n - \rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right) u_m^{n+1}, \tag{58}$$

where the simple choice was made for the two denominator functions,

$$\phi(\Delta t) = \Delta t = k; \quad \psi(\Delta x) = (\Delta x)^2 = h^2, \tag{59}$$

and where nonlocal representation was used for the u^2 terms;

$$u^2 \rightarrow \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right) u_m^{n+1}. \tag{60}$$

A single expression of the scheme is

$$u_m^{n+1} = \frac{(1 + k\rho - \frac{2k}{h^2})u_m^n + \frac{k}{h^2}(u_{m+1}^n + u_{m-1}^n)}{1 + k\rho(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3})}. \tag{61}$$

7.2 Positivity and boundedness: relation between time and space step-sizes

In this subsection, we study the positivity and boundedness properties of NSFD.

From the initial data, if $u(x, 0) = f(x)$ such that $0 \leq f(x) \leq 1$, then we have $0 \leq u(x, t) \leq 1$ [30]. If the quantity u_m^{n+1} from Eq. (61) is required to satisfy the positivity condition ($u_m^{n+1} \geq 0$) if $u_m^n \geq 0$, then we must have

$$\Gamma = 1 + k\rho - 2R \geq 0, \quad R = \frac{k}{h^2}. \tag{62}$$

It follows that

$$1 - \Gamma = 2R - k\rho. \tag{63}$$

We have

$$0 \leq 2R - k\rho \leq 1, \tag{64}$$

which gives

$$0 \leq k \left[\frac{2}{h^2} - \rho \right] \leq 1. \tag{65}$$

It follows that

$$k \leq \frac{h^2}{2} \left[\frac{1 - \Gamma}{1 - \frac{\rho h^2}{2}} \right] \quad \text{and} \quad 0 \leq \Gamma < 1, \tag{66}$$

which is the condition required for positivity [30].

For stability, the amplification factor ξ is given by

$$\xi = \frac{1 + k\rho - \frac{2k}{h^2}(1 - \cos(w))}{1 + k\rho u_{\max}} = \frac{1 + k\rho - \frac{4k}{h^2} \sin^2(\frac{w}{2})}{1 + k\rho u_{\max}}. \tag{67}$$

We require $|\xi| \leq 1$. Since $u_{\max} = 1$, we have

$$\frac{2k}{h^2} \sin^2\left(\frac{w}{2}\right) \leq 1 + k\rho. \tag{68}$$

For $w \in [-\pi, \pi]$, we have

$$\frac{2k}{h^2} \leq 1 + k\rho \Rightarrow k \leq \frac{h^2}{2} \left[\frac{2R - k\rho}{1 - \frac{\rho h^2}{2}} \right], \tag{69}$$

which yields, using Eq. (63),

$$k \leq \frac{h^2}{2} \left[\frac{1 - \Gamma}{1 - \frac{\rho h^2}{2}} \right] \text{ and } 0 \leq \Gamma < 1. \tag{70}$$

Note that we get the same inequality between h and h for stability and positive definiteness to be satisfied. For $h = 0.01$ and $\rho = 10^4$, we have $k \leq 10^{-4}$ or $k \leq \frac{T_{\max}}{25}$.

We next study the boundedness. We assume that $0 \leq u_m^n \leq 1$. We rewrite Eq. (61) as

$$u_m^{n+1} = \frac{\Gamma u_m^n + R(u_{m+1}^n + u_{m-1}^n)}{1 + (\frac{\rho k}{3})(u_{m+1}^n + u_m^n + u_{m-1}^n)}. \tag{71}$$

From [30], Eq. (71) takes the symmetric form if $\Gamma = R$. It follows that

$$k = \frac{h^2}{3} \left[\frac{1}{1 - \frac{\rho h^2}{3}} \right]. \tag{72}$$

Hence

$$R = \frac{k}{h^2} = \left(\frac{1}{3} \right) \left[\frac{1}{1 - \frac{\rho h^2}{3}} \right]. \tag{73}$$

Taking into account the symmetric condition (73), Eq. (71) can be rewritten as

$$u_m^{n+1} = \frac{R(u_m^n + u_{m+1}^n + u_{m-1}^n)}{1 + (\frac{\rho k}{3})(u_{m+1}^n + u_m^n + u_{m-1}^n)}. \tag{74}$$

We know that $0 \leq u_m^n \leq 1$. It follows that

$$0 \leq \frac{u_m^n + u_{m+1}^n + u_{m-1}^n}{3} \leq 1. \tag{75}$$

Multiplying Eq. (75) by $1 - \frac{\rho h^2}{3}$ and dividing by $1 - \frac{\rho h^2}{3}$, we have

$$\left[1 - \frac{\rho h^2}{3} \right] \frac{u_m^n + u_{m+1}^n + u_{m-1}^n}{3 \left[1 - \frac{\rho h^2}{3} \right]} \leq 1, \tag{76}$$

which can be rewritten

$$\frac{u_m^n + u_{m+1}^n + u_{m-1}^n}{3 \left[1 - \frac{\rho h^2}{3} \right]} - \left[\frac{\rho h^2}{3} \right] \frac{u_m^n + u_{m+1}^n + u_{m-1}^n}{3 \left[1 - \frac{\rho h^2}{3} \right]} \leq 1. \tag{77}$$

It follows that

$$\frac{u_m^n + u_{m+1}^n + u_{m-1}^n}{3 \left[1 - \frac{\rho h^2}{3} \right]} \leq 1 + \frac{h^2}{3} \left[\frac{1}{1 - \frac{\rho h^2}{3}} \right] \left(\frac{\rho}{3} \right) (u_m^n + u_{m+1}^n + u_{m-1}^n). \tag{78}$$

Table 6 L_1 and L_∞ errors and CPU time at some different values of time-step size k for Problem 1 at time 2.5×10^{-3} with spatial mesh size $h = 0.01$ using NSFD scheme, where $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/25$	1.9125×10^{-1}	9.8876×10^{-1}	0.683
$T_{\max}/50$	1.2304×10^{-1}	9.2384×10^{-1}	0.743
$T_{\max}/100$	7.0554×10^{-2}	7.0761×10^{-1}	0.689
$T_{\max}/200$	3.6165×10^{-2}	4.1770×10^{-1}	0.724
$T_{\max}/300$	2.3031×10^{-2}	2.7850×10^{-1}	0.738
$T_{\max}/400$	1.6095×10^{-2}	1.9599×10^{-1}	0.758
$T_{\max}/500$	1.1808×10^{-2}	1.4689×10^{-1}	0.831
$T_{\max}/600$	8.8945×10^{-3}	1.1268×10^{-1}	0.814
$T_{\max}/700$	6.7864×10^{-3}	8.7538×10^{-2}	0.854
$T_{\max}/800$	5.1909×10^{-3}	6.8368×10^{-2}	0.900
$T_{\max}/900$	3.9428×10^{-3}	5.3310×10^{-2}	0.960
$T_{\max}/1000$	2.9430×10^{-3}	4.1187×10^{-2}	0.972
$T_{\max}/1100$	2.1324×10^{-3}	3.1229×10^{-2}	0.939
$T_{\max}/1200$	1.4815×10^{-3}	2.2908×10^{-2}	0.995
$T_{\max}/1300$	9.7728×10^{-4}	1.5854×10^{-2}	1.034
$T_{\max}/1400$	6.3200×10^{-4}	9.8018×10^{-3}	1.077
$T_{\max}/1500$	4.5216×10^{-4}	4.5522×10^{-3}	1.175
$T_{\max}/1600$	5.0673×10^{-4}	5.3751×10^{-3}	1.210
$T_{\max}/1700$	8.4460×10^{-4}	7.6933×10^{-3}	1.263
$T_{\max}/1800$	1.1475×10^{-3}	1.0160×10^{-2}	1.298

Using Eqs. (73) and (72), we have

$$R(u_m^n + u_{m+1}^n + u_{m-1}^n) \leq 1 + \frac{k\rho}{3}(u_m^n + u_{m+1}^n + u_{m-1}^n), \tag{79}$$

which gives

$$0 \leq u_m^{n+1} = \frac{R(u_m^n + u_{m+1}^n + u_{m-1}^n)}{1 + (\frac{k\rho}{3})(u_{m+1}^n + u_m^n + u_{m-1}^n)} \leq 1. \tag{80}$$

Hence the boundedness of u_m^{n+1} .

We tabulate the L_1 and L_∞ errors and CPU time when Problems 1 and 2 are solved using the NSFD scheme at some different values of time-step size k , with spatial step size $h = 0.01$. The errors are displayed in Tables 6 and 7. As the time-step size is reduced, the errors initially decrease, and optimal k is approximately equal to $T_{\max}/1500$. On further decreasing k , the errors start to increase. Plots of u vs x at some values of time-step size are displayed in Fig. 3.

8 NSFD- ϵ schemes

We construct the NSFD- ϵ by adding the expression $\epsilon_1 u_{m+1}^n + \epsilon_2 u_m^n + \epsilon_3 u_{m-1}^n$ to u_m^{n+1} to Eq. (61). We thus have

$$u_m^{n+1} = \frac{(1 + k\rho - \frac{2k}{h^2})u_m^n + \frac{k}{h^2}(u_{m+1}^n + u_{m-1}^n)}{1 + k\rho(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3})} + \epsilon_1 u_{m+1}^n + \epsilon_2 u_m^n + \epsilon_3 u_{m-1}^n. \tag{81}$$

Using the Taylor series expansion about (n, m) , we have

$$u + ku_t + \frac{k^2}{2}u_{tt} + \frac{k^3}{6}u_{ttt} + O(k^4)$$

Table 7 L_1 and L_∞ errors and CPU time at some different values of time-step size k for Problem 2 at time 2.5×10^{-3} with spatial mesh size $h = 0.01$ using NSFD scheme, where $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/25$	1.9125×10^{-1}	9.8876×10^{-1}	1.130
$T_{\max}/50$	1.2304×10^{-1}	9.2384×10^{-1}	1.470
$T_{\max}/100$	7.0554×10^{-2}	7.0761×10^{-1}	2.280
$T_{\max}/200$	3.6165×10^{-2}	4.1770×10^{-1}	4.844
$T_{\max}/300$	2.3031×10^{-2}	2.7850×10^{-1}	7.950
$T_{\max}/400$	1.6095×10^{-2}	1.9599×10^{-1}	12.219
$T_{\max}/500$	1.1808×10^{-2}	1.4689×10^{-1}	17.052
$T_{\max}/600$	8.8945×10^{-3}	1.1268×10^{-1}	22.408
$T_{\max}/700$	6.7864×10^{-3}	8.7538×10^{-2}	28.688
$T_{\max}/800$	5.1909×10^{-3}	6.8368×10^{-2}	35.804
$T_{\max}/900$	3.9428×10^{-3}	5.3310×10^{-2}	44.176
$T_{\max}/1000$	2.9430×10^{-3}	4.1187×10^{-2}	52.029
$T_{\max}/1100$	2.1324×10^{-3}	3.1229×10^{-2}	61.695
$T_{\max}/1200$	1.4815×10^{-3}	2.2908×10^{-2}	77.506
$T_{\max}/1300$	9.7728×10^{-4}	1.5854×10^{-2}	97.953
$T_{\max}/1400$	6.3200×10^{-4}	9.8018×10^{-3}	98.065
$T_{\max}/1500$	4.5216×10^{-4}	4.5522×10^{-3}	110.000
$T_{\max}/1600$	5.0673×10^{-4}	5.3751×10^{-3}	110.565
$T_{\max}/1700$	8.4460×10^{-4}	7.6933×10^{-3}	115.565
$T_{\max}/1800$	1.1475×10^{-3}	1.0160×10^{-2}	118.565

$$\begin{aligned}
 &= \frac{ku_{xx} + (1 + k\rho)u}{1 + k\rho u + \frac{k\rho h^2}{3}u_{xx}} + (\epsilon_1 + \epsilon_2 + \epsilon_3)u \\
 &\quad + h(\epsilon_3 - \epsilon_1)u_x + \frac{h^2}{2}(\epsilon_1 + \epsilon_2)u_{xx} + \frac{h^3}{6}(\epsilon_3 - \epsilon_1)u_{xxx} + O(h^4), \tag{82}
 \end{aligned}$$

which can be written as

$$\begin{aligned}
 &\left(1 + k\rho u + \frac{k\rho h^2}{3}u_{xx}\right)u_t + \left(\frac{k}{2} + \frac{k}{2}\rho u + \frac{k^2}{6}\rho h^2u_{xx}\right)u_{tt} \\
 &\quad + \left(\frac{k^2}{6} + \frac{k^3}{6}\rho u + \frac{k^3}{18}\rho h^2u_{xx}\right)u_{ttt} + \rho u^2 + \frac{\rho h^2}{3}uu_{xx} \\
 &= u_{xx} + \rho u + (\epsilon_1 + \epsilon_2 + \epsilon_3)u\left(\frac{1}{k} + \frac{\rho h^2}{3}u_{xx}\right) \\
 &\quad + (\epsilon_3 - \epsilon_1)u_x\left(\frac{h}{k} + h\rho u\frac{\rho h^3}{3}u_{xx}\right) + (\epsilon_1 + \epsilon_2)u_{xx}\left(\frac{h^2}{k} + h^2\rho u + \frac{\rho h^4}{3}u_{xx}\right) \\
 &\quad + (\epsilon_3 - \epsilon_1)u_{xxx}\left(\frac{h^3}{k} + h^3\rho u\frac{\rho h^5}{3}u_{xx}\right) + O(k^4) + O(h^4). \tag{83}
 \end{aligned}$$

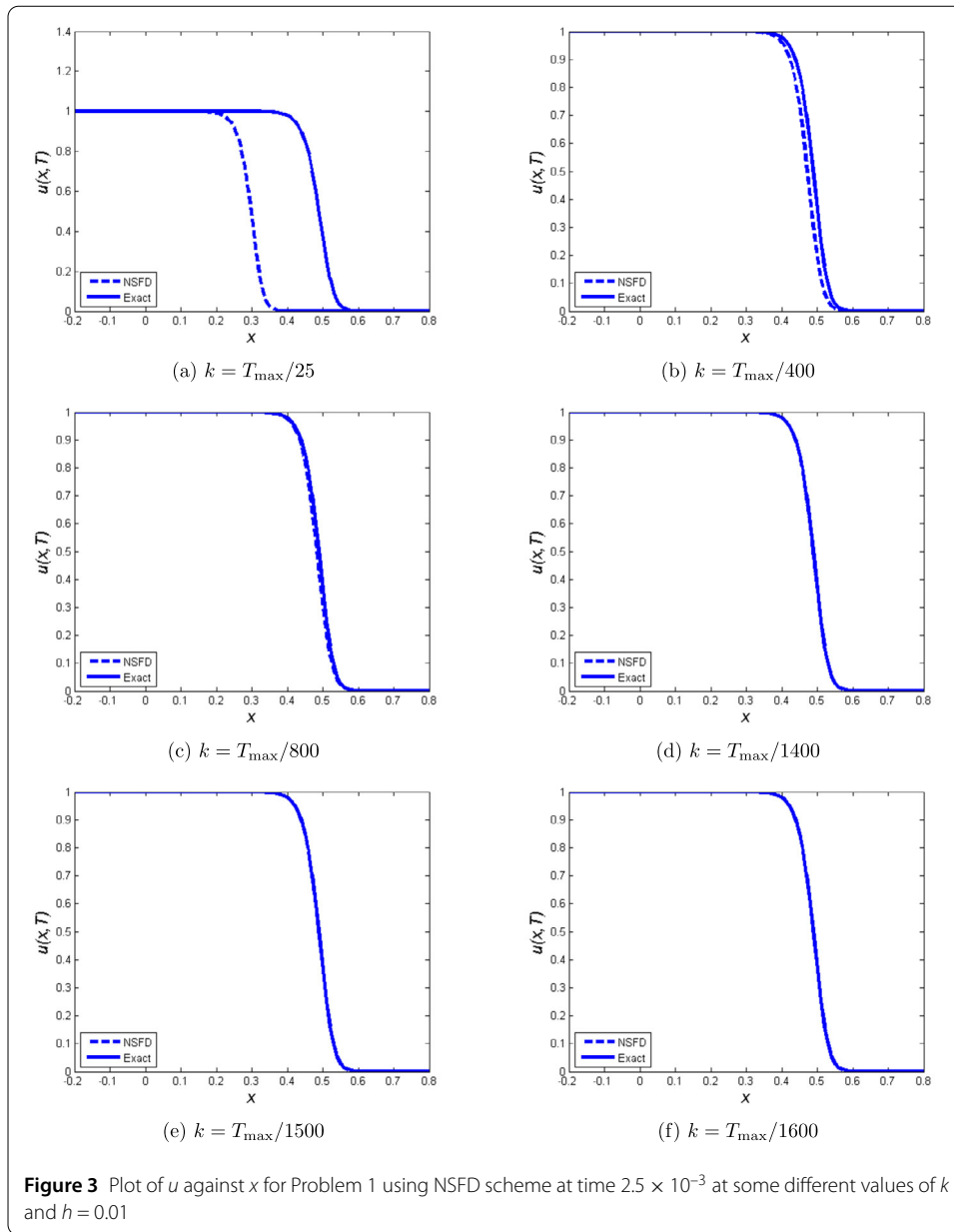
As $k, h \rightarrow 0$, we recover

$$u_t = u_{xx} + \rho u(1 - u) \tag{84}$$

if $\epsilon_1 + \epsilon_2 + \epsilon_3 = 0$ and $\epsilon_1 - \epsilon_3 = 0$. It thus follows that $\epsilon_1 = \epsilon_3 = \epsilon$ and $\epsilon_2 = -2\epsilon$.

For accuracy, we next consider Eq. (83) with $\epsilon_1 = \epsilon_3 = \epsilon$ and $\epsilon_2 = -2\epsilon$. It follows that

$$\left(1 + k\rho u + \frac{k\rho h^2}{3}u_{xx}\right)u_t + \left(\frac{k}{2} + \frac{k}{2}\rho u + \frac{k^2}{6}\rho h^2u_{xx}\right)u_{tt}$$



$$\begin{aligned}
 & + \left(\frac{k^2}{6} + \frac{k^3}{6}\rho u + \frac{k^3}{18}\rho h^2 u_{xx} \right) u_{ttt} + \rho u^2 + \frac{\rho h^2}{3} u u_{xx} \\
 & = u_{xx} + \rho u - \epsilon u_{xx} \left(\frac{h^2}{k} + h^2 \rho u + \frac{\rho h^4}{3} u_{xx} \right) + O(k^4) + O(h^4). \tag{85}
 \end{aligned}$$

We rewrite Eq. (85) in the form

$$\begin{aligned}
 & u_t - u_{xx} - \rho u + \rho u^2 \\
 & = - \left(k \rho u + \frac{k \rho h^2}{3} u_{xx} \right) u_t - \left(\frac{k}{2} + \frac{k}{2} \rho u + \frac{k^2}{6} \rho h^2 u_{xx} \right) u_{tt} \\
 & \quad - \epsilon \left(\frac{\rho h^2}{3} u + \frac{h^2}{k} + h^2 \rho u + \frac{\rho h^4}{3} u_{xx} \right) u_{xx} - \left(\frac{k^2}{6} + \frac{k^3}{6} \rho u + \frac{k^3}{18} \rho h^2 u_{xx} \right) u_{ttt}
 \end{aligned}$$

$$+ O(k^4) + O(h^4). \tag{86}$$

We conclude that NSFD- ϵ has the first-order accuracy in time and the first-order accuracy in space.

8.1 Positivity and boundedness

We study the positivity of the method. We rewrite Eq. (81) as

$$u_m^{n+1} = \frac{\Gamma u_m^n + R(u_{m+1}^n + u_{m-1}^n)}{1 + k\rho\left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3}\right)}, \tag{87}$$

where

$$\begin{aligned} \Gamma &= 1 + k\rho - \frac{2k}{h^2} - 2\epsilon \left[1 + k\rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right) \right] \quad \text{and} \\ R &= \frac{k}{h^2} + \epsilon \left[1 + k\rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right) \right]. \end{aligned} \tag{88}$$

We assume that $0 \leq u_m^n \leq 1$. Then u_m^{n+1} is positive if

$$\Gamma = 1 + k\rho - \frac{2k}{h^2} - 2\epsilon \left[1 + k\rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right) \right] \geq 0. \tag{89}$$

By taking the maximum of $u_j^n, j = m - 1, m, m + 1$, we have

$$\Gamma = 1 + k\rho - \frac{2k}{h^2} - 2\epsilon \left(1 + k\rho \frac{3|u_{\max}|}{3} \right) \geq 0, \tag{90}$$

which gives

$$\Gamma = 1 + k\rho - \frac{2k}{h^2} - 2\epsilon(1 + k\rho) \geq 0, \tag{91}$$

since $u_{\max} = 1$. It follows from Eq. (91) that

$$1 - 2\epsilon - \Gamma = k \left[\frac{2}{h^2} - \rho(1 - 2\epsilon) \right] \leq 1 - 2\epsilon. \tag{92}$$

Hence u_m^{n+1} is positive if

$$k \leq \frac{h^2}{2} \left[\frac{1 - 2\epsilon - \Gamma}{1 - \frac{\rho h^2}{2}(1 - 2\epsilon)} \right] \quad \text{and} \quad 0 \leq \Gamma < 1 - 2\epsilon. \tag{93}$$

For stability analysis, we apply Fourier series analysis to Eq. (87) to obtain the amplification factor ξ . Thus

$$\xi = \frac{1 + k\rho - \frac{2k}{h^2}}{1 + k\rho|u_{\max}|} - 2\epsilon + 2 \left[\frac{\frac{k}{h^2}}{1 + k\rho|u_{\max}|} + \epsilon \right] \cos(w). \tag{94}$$

For the stability, we have $|\xi| \leq 1$. Since $u_{\max} = 1$, we solve

$$-2 \leq -2 \left[2\epsilon + \frac{2k}{1+k\rho} \right] \sin^2 \left(\frac{w}{2} \right) \leq 0 \tag{95}$$

and obtain that, for $w \in [-\pi, \pi]$,

$$k \leq \frac{h^2}{2} \left[\frac{1-2\epsilon}{1 - \frac{\rho h^2}{2}(1-2\epsilon)} \right]. \tag{96}$$

Hence

$$k \leq \frac{h^2}{2} \left[\frac{1-2\epsilon-\Gamma}{1 - \frac{\rho h^2}{2}(1-2\epsilon)} \right]. \tag{97}$$

Since $\epsilon = 0.01$ and $h = 0.01$, we have $k \leq 9.6078 \times 10^{-5}$ or $k \leq T_{\max}/26$ for positivity or stability.

For the boundedness of NSFD- ϵ , we use the symmetric condition by taking $R = \Gamma$ from Eq. (87). It follows that

$$1 + k\rho - \frac{2k}{h^2} - \frac{k}{h^2} = 3\epsilon \left(1 + k\rho \frac{3|u_{\max}|}{3} \right). \tag{98}$$

With $u_{\max} = 1$, we have

$$k = \frac{h^2}{3} \left[\frac{1-3\epsilon}{1 - \frac{\rho h^2}{3}(1-3\epsilon)} \right] \tag{99}$$

and

$$\frac{k}{(1-3\epsilon)h^2} = \frac{1}{3} \left[\frac{1}{1 - \frac{\rho h^2}{3}(1-3\epsilon)} \right], \tag{100}$$

which is a symmetric condition. Using Eq. (100), Eq. (87) becomes

$$u_m^{n+1} = \frac{\Gamma(u_{m+1}^n + u_m^n + u_{m-1}^n)}{1 + k\rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right)}. \tag{101}$$

We know that $0 \leq u_j^n \leq 1, j = m-1, m, m+1$. It follows that

$$\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \leq 1. \tag{102}$$

Multiplying and dividing Eq. (102) by $(1 - \frac{\rho h^2}{3}(1-3\epsilon))$, we have

$$\left[1 - \frac{\rho h^2}{3}(1-3\epsilon) \right] \frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3 \left[1 - \frac{\rho h^2}{3}(1-3\epsilon) \right]} \leq 1. \tag{103}$$

It follows that

$$\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3[1 - \frac{\rho h^2}{3}(1 - 3\epsilon)]} \leq 1 + \frac{\rho}{3} \left(\frac{h^2}{3}\right) \left[\frac{1 - 3\epsilon}{1 - \frac{\rho h^2}{3}(1 - 3\epsilon)}\right] (u_{m+1}^n + u_m^n + u_{m-1}^n). \tag{104}$$

From Eq. (99) and the symmetry condition (100) we have

$$\Gamma(u_{m+1}^n + u_m^n + u_{m-1}^n) \leq 1 + \frac{k\rho}{3}(u_{m+1}^n + u_m^n + u_{m-1}^n). \tag{105}$$

It follows that

$$0 \leq u_m^{n+1} = \frac{\Gamma(u_{m+1}^n + u_m^n + u_{m-1}^n)}{1 + \frac{k\rho}{3}(u_{m+1}^n + u_m^n + u_{m-1}^n)} \leq 1. \tag{106}$$

Hence the boundedness of NSFD- ϵ method. Generally without symmetry condition (100), we consider the boundedness of u_m^{n+1} in case of NSFD, and we find the boundedness of NSFD- ϵ by stating from Eq. (81) that

$$0 \leq u_m^{n+1} \leq 1 + \epsilon(u_{m-1}^n - 2u_m^n + u_{m+1}^n). \tag{107}$$

We rewrite $\epsilon(u_{m-1}^n - 2u_m^n + u_{m+1}^n)$ as

$$\epsilon(u_{m-1}^n - 2u_m^n + u_{m+1}^n) = \epsilon(u_{m-1}^n + u_{m+1}^n) - 2\epsilon u_m^n. \tag{108}$$

Since $u_j^n, j = m - 1, m, m + 1$, are bounded ($0 \leq u_j^n \leq 1$), and hence from Eq. (108) we have

$$\epsilon(u_{m-1}^n + u_{m+1}^n) - 2\epsilon u_m^n \leq 2\epsilon - 2\epsilon u_m^n, \tag{109}$$

due to the fact $\epsilon(u_{m-1}^n + u_{m+1}^n) \leq 2\epsilon$. It follows that

$$\epsilon(u_{m-1}^n + u_{m+1}^n) - 2\epsilon u_m^n \leq 2\epsilon - 2\epsilon u_m^n = 2\epsilon(1 - u_m^n). \tag{110}$$

The quantity $1 - u_m^n$ in Eq. (110) is bounded by

$$0 \leq (1 - u_m^n) \leq 1. \tag{111}$$

It follows from Eqs. (110) and (111) that

$$\epsilon(u_{m-1}^n - 2u_m^n + u_{m+1}^n) = \epsilon(u_{m-1}^n + u_{m+1}^n) - 2\epsilon u_m^n \leq 2\epsilon. \tag{112}$$

Hence the boundedness of u_m^{n+1} for NSFD- ϵ method from Eq. (107):

$$0 \leq u_m^{n+1} \leq 1 + 2\epsilon. \tag{113}$$

We tabulate the L_1 and L_∞ errors and CPU time when Problems 1 and 2 are solved using the NSFD- ϵ scheme at some different values of time-step size k with spatial step size $h = 0.01$. The errors are displayed in Tables 8 and 9. We obtain plots of u vs x at some value of time-step size with $h = 0.01$ in Fig. 4.

Table 8 L_1 and L_∞ errors and CPU time at some different values of time-step size k for Problem 1 at time 2.5×10^{-3} with spatial mesh size $h = 0.01$, $\epsilon = 0.01$ using NSFD- ϵ scheme, $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/26$	1.8636×10^{-1}	9.8705×10^{-1}	0.653
$T_{\max}/50$	1.2102×10^{-1}	9.1868×10^{-1}	0.662
$T_{\max}/100$	6.5470×10^{-2}	6.7456×10^{-1}	0.670
$T_{\max}/200$	2.4582×10^{-2}	2.9094×10^{-1}	0.682
$T_{\max}/300$	4.8806×10^{-3}	5.9139×10^{-2}	0.699
$T_{\max}/333$	1.0091×10^{-3}	9.5627×10^{-3}	0.715
$T_{\max}/334$	9.9448×10^{-4}	1.0714×10^{-2}	0.707
$T_{\max}/335$	1.0119×10^{-3}	1.1866×10^{-2}	0.787
$T_{\max}/400$	8.5987×10^{-3}	1.0311×10^{-1}	0.722
$T_{\max}/500$	1.9385×10^{-2}	2.2445×10^{-1}	0.780

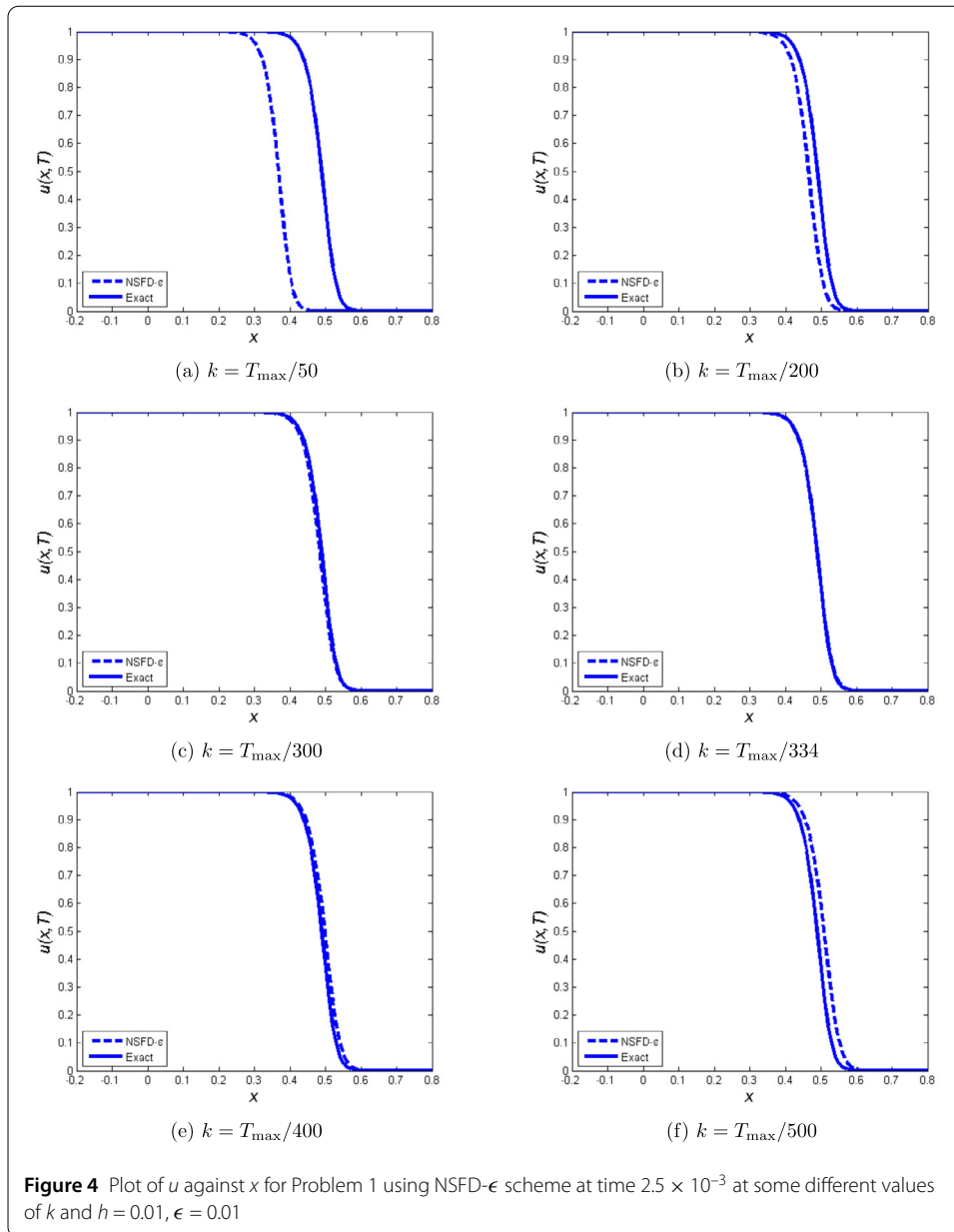
Table 9 L_1 and L_∞ errors and CPU time at some different values of time-step size, k for Problem 2 at time 2.5×10^{-3} with spatial mesh size, $h = 0.01$, $\epsilon = 0.01$ using NSFD- ϵ scheme, $T_{\max} = 2.5 \times 10^{-3}$

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/26$	1.8636×10^{-1}	9.8705×10^{-1}	1.137
$T_{\max}/50$	1.2102×10^{-1}	9.1868×10^{-1}	1.485
$T_{\max}/100$	6.5470×10^{-2}	6.7456×10^{-1}	2.391
$T_{\max}/200$	2.4582×10^{-2}	2.9094×10^{-1}	4.880
$T_{\max}/300$	4.8806×10^{-3}	5.9139×10^{-2}	8.207
$T_{\max}/333$	1.0091×10^{-3}	9.5627×10^{-3}	9.437
$T_{\max}/334$	9.9448×10^{-4}	1.0714×10^{-2}	9.534
$T_{\max}/335$	1.0119×10^{-3}	1.1866×10^{-2}	9.540
$T_{\max}/400$	8.5987×10^{-3}	1.0311×10^{-1}	12.181
$T_{\max}/500$	1.9385×10^{-2}	2.2445×10^{-1}	17.074

9 Artificial viscosity

We observe from the numerical results of FTCS and NSFD that the schemes are plagued by dispersion at time step size quite close to the stability limit of the temporal step size at $h = 0.01$. We propose to use the artificial viscosity approach. The method of artificial viscosity has been first introduced by Neumann et al. [32], who explicitly added a viscosity term to the inviscid gas dynamics equations to allow the computation of shock waves. Their approach was to change the momentum and energy equations by adding dissipation in the form of viscosity term to the pressure that would give the thickness of shock waves and also to space out the computational mesh. The artificial viscosity was purposely made proportional to the second derivative u_{xx} (which is positive in compression and negative in expansion) to ensure the mathematical consistency. It should satisfy some constraints: the modified equation must possess solutions without any discontinuities, the Rankine–Hugoniot conditions must hold (for conservation equations), and the dissipative term must be negligible outside of the shock waves [33].

It is also known that artificial viscosity can be expressed implicitly following the work of Noh et al. [34], who was the first to present an analysis of the implicit artificial viscosity of the upwind method applied to linear advection equation. It has been proved that, for a given unstable numerical method, in general, for first-order linear equations, it can be stabilized by adding a sufficiently large viscosity term. Other forms of artificial viscosity can be found in Landshoff [35] and Wilkins [36], who attribute it to Kurapatenko [37].



Finally, it has been specified by Caramana et al. [38] that an artificial viscosity should have the following properties:

1. Dissipativity: The artificial viscosity must only act to decrease the kinetic energy.
2. Galilean invariance: The viscosity should vanish smoothly as the velocity field becomes constant.
3. Self-similar motion invariance: The viscosity should vanish for uniform contraction and rigid rotation.
4. Wave-front invariance: The viscosity should have no effect along a wave front of constant phase on a grid aligned with shocks.
5. Viscous force continuity: The viscous force should go to zero continuously as compression vanishes and remain zero for expansion.

We start with a simple linear advection equation, $u_t + u_x = 0$ (Eq. (35) with $c = 1$). We add the artificial viscosity σhu_{xx} to obtain

$$u_t + u_x = \sigma hu_{xx}. \tag{114}$$

As $h \rightarrow 0$, we recover the initial linear advection equation $u_t + u_x = 0$, and σ is a real parameter. The numerical discretization for Eq. (114) is

$$\frac{u_m^{n+1} - u_m^n}{k} + D_0 u_m^n = \sigma h D_+ D_- u_m^n, \tag{115}$$

where $D_+ u_m^n = \frac{u_{m+1}^n - u_m^n}{\Delta x}$, $D_- u_m^n = \frac{u_m^n - u_{m-1}^n}{\Delta x}$, and $D_0 u_m^n = \frac{u_{m+1}^n - u_{m-1}^n}{2\Delta x}$ are respectively forward, backward, and centered differencing operators. Rewriting Eq. (115), we have

$$u_m^{n+1} = u_m^n - k D_0 u_m^n + \sigma kh D_+ D_- u_m^n. \tag{116}$$

As $h \rightarrow 0$, the scheme given by Eq. (116) is a consistent approximation of Eq. (35) with $c = 1$.

9.1 FTCS with artificial viscosity

We need to solve

$$u_t = u_{xx} + \rho u(1 - u). \tag{117}$$

We add σhu_{xx} and obtain the new equation

$$u_t = u_{xx} + \rho u(1 - u) + \sigma hu_{xx}. \tag{118}$$

The numerical scheme used to discretize Eq. (118) is

$$\frac{u_m^{n+1} - u_m^n}{k} = \frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{h^2} + \rho u_m^n (1 - u_m^n) + \sigma h \left(\frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{h^2} \right), \tag{119}$$

which can be rewritten as

$$u_m^{n+1} = \left[1 - \frac{2k}{h^2} (1 + \sigma h) \right] u_m^n + k \rho u_m^n (1 - u_m^n) + \frac{k}{h^2} (1 + \sigma h) (u_{m+1}^n + u_{m-1}^n). \tag{120}$$

For the order of accuracy, the Taylor series expansion about the point (n, m) of (120) gives

$$\begin{aligned} & u + ku_t + \frac{k^2}{2} u_{tt} + \frac{k^3}{6} u_{ttt} \\ &= \left(1 - \frac{2k}{h^2} (1 + \sigma h) \right) u + k \rho u(1 - u) \\ &+ \frac{k}{h^2} (1 + \sigma h) (2u + h^2 u_{xx}) + O(k^4) + O(h^4), \end{aligned} \tag{121}$$

Table 10 L_1 and L_∞ errors CPU time at some different values of time-step size k for Problem 1 with $\rho = 10^4$ at time 2.5×10^{-3} with spatial mesh size $\sigma = 2.0$ and $h = 0.01$ using FTCS with artificial viscosity

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/53$	2.6947×10^{-1}	1.5155	0.737
$T_{\max}/100$	6.2546×10^{-2}	6.8175×10^{-1}	0.737
$T_{\max}/200$	3.0034×10^{-2}	3.6771×10^{-1}	0.742
$T_{\max}/300$	1.7645×10^{-2}	2.1809×10^{-1}	0.758
$T_{\max}/400$	1.1108×10^{-2}	1.3983×10^{-1}	0.782
$T_{\max}/500$	7.0695×10^{-3}	8.9529×10^{-2}	0.800
$T_{\max}/600$	4.3263×10^{-3}	5.5222×10^{-2}	0.847
$T_{\max}/700$	2.3414×10^{-3}	3.0552×10^{-2}	0.856
$T_{\max}/800$	8.5592×10^{-4}	1.2040×10^{-2}	0.907
$T_{\max}/850$	3.1048×10^{-4}	4.4291×10^{-3}	0.913
$T_{\max}/860$	2.2700×10^{-4}	3.0141×10^{-3}	0.918
$T_{\max}/870$	1.6308×10^{-4}	1.8608×10^{-3}	0.923
$T_{\max}/880$	1.3833×10^{-4}	2.3032×10^{-3}	0.926
$T_{\max}/890$	2.3267×10^{-4}	2.9465×10^{-3}	0.930
$T_{\max}/900$	3.3881×10^{-4}	3.6042×10^{-3}	0.951
$T_{\max}/1000$	1.2861×10^{-3}	1.4273×10^{-2}	0.967
$T_{\max}/1100$	2.0648×10^{-3}	2.3789×10^{-2}	1.001
$T_{\max}/1200$	2.7162×10^{-3}	3.1756×10^{-2}	1.042
$T_{\max}/1300$	3.2692×10^{-3}	3.8522×10^{-2}	1.071
$T_{\max}/1400$	3.7445×10^{-3}	4.4336×10^{-2}	1.124
$T_{\max}/1500$	4.1573×10^{-3}	4.9385×10^{-2}	1.142

which gives

$$u_t - u_{xx} - \rho u(1 - u) = -\frac{k}{2}u_{tt} - \frac{k^2}{6}u_{ttt} + \sigma hu_{xx} + O(k^3) + O\left(\frac{h^4}{k}\right). \tag{122}$$

We conclude that FTCS with artificial viscosity has the first-order accuracy in time and the first-order accuracy in space.

If $\sigma = 0$, then we recover the FTCS scheme. The amplification factor ξ of the FTCS scheme with artificial viscosity is

$$\xi = 1 + \frac{2k}{h^2}(1 + \sigma h)(\cos(w) - 1) + k\rho(1 - |u_{\max}|). \tag{123}$$

We choose $u_{\max} = 1$ based on the numerical experiment chosen and have

$$\xi = 1 - \frac{4k}{h^2}(1 + \sigma h)\sin^2\left(\frac{w}{2}\right). \tag{124}$$

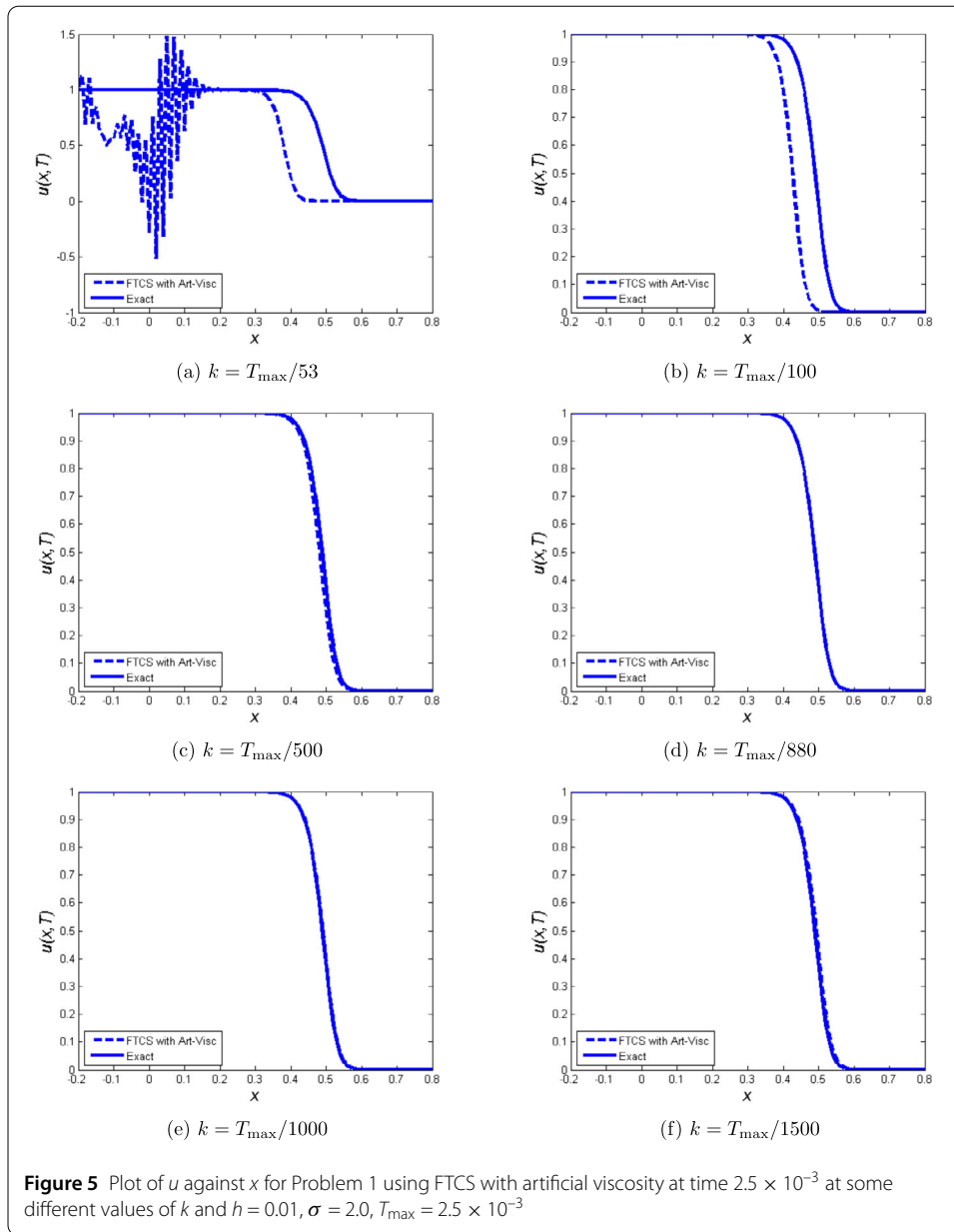
The stability region is

$$k < \frac{h^2}{2} \left[\frac{1}{1 + \sigma h} \right]. \tag{125}$$

We choose $\sigma = 2.0$ and $h = 0.01$ and obtain $k < 4.9020 \times 10^{-5}$ or $T_{\max}/51$.

We tabulate the L_1 and L_∞ errors and CPU time when Problem 1 is solved using FTCS with artificial viscosity at some different values of time-step size k with spatial step size $h = 0.01, \sigma = 2.0$. The errors are displayed in Table 10. Plots of u vs x are displayed in Fig. 5.

From Table 10 we deduce that FTCS with artificial viscosity can give accurate results at a smaller CPU as compared to FTCS, provided that σ is correctly chosen.



9.2 Nonstandard finite difference method with artificial viscosity

The nonstandard finite difference scheme with artificial viscosity to discretize Eq. (118) is

$$\begin{aligned} \frac{u_m^{n+1} - u_m^n}{k} &= \frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{h^2} + \rho u_m^n - \rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right) u_m^{n+1} \\ &+ \sigma h \left(\frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{h^2} \right). \end{aligned} \tag{126}$$

A single expression for the scheme is

$$u_m^{n+1} = \frac{(1 + k\rho - 2\beta)u_m^n + \beta(u_{m+1}^n + u_{m-1}^n)}{1 + k\rho \left(\frac{u_{m+1}^n + u_m^n + u_{m-1}^n}{3} \right)}, \quad \text{where } \beta = \frac{k}{h^2}(1 + \sigma h). \tag{127}$$

The Taylor series expansion of this scheme about (m, n) gives

$$\begin{aligned} & \left(u + ku_t + \frac{k^2}{2}u_{tt} + \frac{k^3}{6}u_{ttt}\right)\left(1 + k\rho u + k\rho\frac{h^2}{3}u_{xx}\right) \\ & = (1 + k\rho)u + ku_{xx} + O(k^4) + O(h^4), \end{aligned} \tag{128}$$

which can be written as

$$\begin{aligned} & u_t - u_{xx} - \rho u + \rho u^2 - \sigma hu_{xx} \\ & = -\left(\rho u\frac{h^2}{3} - \sigma h\right)u_{xx} - k\left(\frac{\rho u}{3} + \frac{\rho h^2}{3}u_{xx}\right)u_t \\ & \quad - \left(\frac{k}{3} + \frac{k^2}{2}\left(\frac{\rho u}{3} + \frac{\rho h^2}{3}u_{xx}\right)\right)u_{tt} \\ & \quad - \left(\frac{k^2}{6} + \frac{k^3}{6}\left(\frac{\rho u}{3} + \frac{\rho h^2}{3}u_{xx}\right)\right)u_{ttt} \\ & \quad + O(k^4) + O(h^4). \end{aligned} \tag{129}$$

We conclude that NSFD with artificial viscosity has the first-order accuracy in time and the first-order accuracy in space.

9.2.1 Positivity and boundedness

The scheme given by Eq. (127) is positive if $\Gamma = 1 + k\rho - \frac{2k}{h^2}(1 + \sigma h) \geq 0$. Hence the scheme is positive definite under the conditions

$$k \leq \frac{h^2}{2} \left[\frac{1 - \Gamma}{1 + \sigma h - \frac{\rho h^2}{2}} \right] \quad \text{and} \quad 0 \leq \Gamma < 1. \tag{130}$$

For $h = 0.01$ and $\sigma = 2.0$, we have $k \leq 9.6154 \times 10^{-5}$.

For boundedness, we start by assuming that $0 \leq u_m^n \leq 1$. We apply the same steps as in case of NSFD without artificial viscosity by letting

$$\Gamma = 1 + k\rho - 2R, \quad R = \beta = \frac{k}{h^2}(1 + \sigma h), \tag{131}$$

and we use the symmetric condition

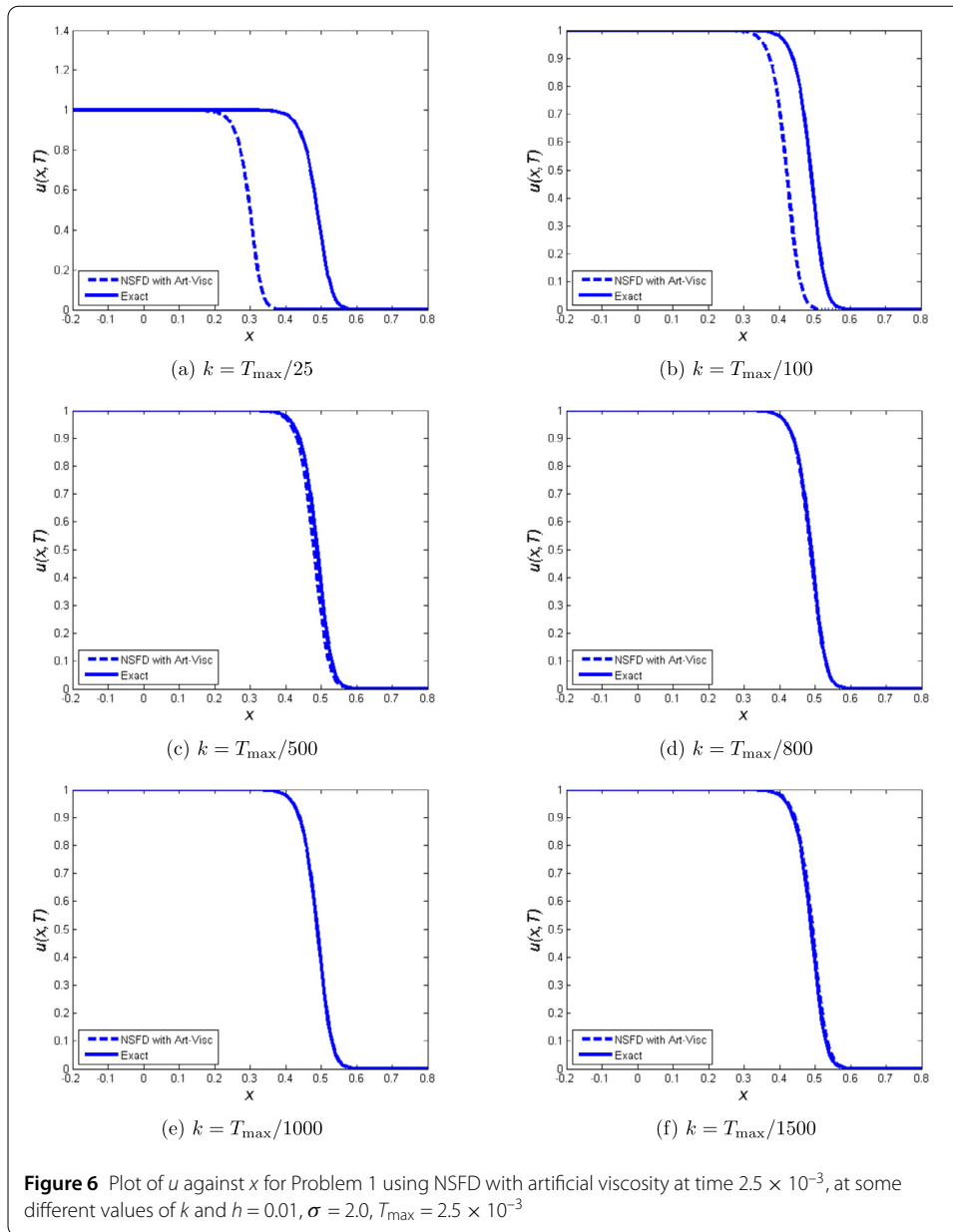
$$R = \frac{k}{h^2} = \left(\frac{1}{3}\right) \left[\frac{1}{1 + \frac{\rho\sigma}{3} - \frac{\rho h^2}{3}} \right]. \tag{132}$$

This gives

$$0 \leq u_m^{n+1} \leq 1. \tag{133}$$

Hence the boundedness of u_m^{n+1} .

We tabulate the L_1 and L_∞ errors, CPU time when Problem 1 is solved using NSFD with artificial viscosity scheme at some different values of time-step size k with spatial step size $h = 0.01$, $\sigma = 2.0$. The errors are displayed in Table 11. Plots of u vs x using NSFD with artificial viscosity are displayed in Fig. 6.



10 Conclusion

In this work, we have initially used the FTCS and NSFD schemes to solve Fisher’s equation when the coefficient of diffusion is much less than the coefficient of reaction term and the initial condition consists of an exponential function. The time-step size must be relatively small to obtain accurate results, and the CPU time becomes large if the domain is large.

We propose four schemes, namely FTCS- ϵ , NSFD- ϵ , FTCS with Artificial Viscosity, and NSFD with artificial viscosity, which give quite accurate results at larger time-step size and, consequently, a smaller CPU time as compared to the FTCS and NSFD methods. Also, the L_1 and L_∞ errors at optimal time step size and $h = 0.01$ are smaller than when using the MMPDE and MMDAE methods.

Table 11 L_1 and L_∞ errors and CPU time at some different values of time step size k for Problem 1 at time 2.5×10^{-3} with spatial mesh size $h = 0.01$, $\sigma = 2.0$ using NSFD with artificial viscosity method

Time step (k)	L_1 error	L_∞ error	CPU (s)
$T_{\max}/25$	1.8982×10^{-1}	9.8811×10^{-1}	0.891
$T_{\max}/50$	1.2102×10^{-1}	9.1869×10^{-1}	0.893
$T_{\max}/100$	6.8010×10^{-2}	6.9128×10^{-1}	0.897
$T_{\max}/200$	3.3252×10^{-2}	3.8653×10^{-1}	0.928
$T_{\max}/300$	1.9973×10^{-2}	2.4143×10^{-1}	0.998
$T_{\max}/400$	1.2961×10^{-2}	1.5892×10^{-1}	0.999
$T_{\max}/500$	8.6255×10^{-3}	1.0838×10^{-1}	1.019
$T_{\max}/600$	5.6799×10^{-3}	7.3290×10^{-2}	1.041
$T_{\max}/700$	3.5494×10^{-3}	4.7746×10^{-2}	1.089
$T_{\max}/800$	1.9544×10^{-3}	2.8406×10^{-2}	1.355
$T_{\max}/900$	8.4110×10^{-4}	1.3294×10^{-2}	1.355
$T_{\max}/950$	5.2648×10^{-4}	7.1035×10^{-3}	1.564
$T_{\max}/1000$	4.4308×10^{-4}	5.5908×10^{-3}	1.664
$T_{\max}/1050$	7.8758×10^{-4}	8.5810×10^{-3}	1.683
$T_{\max}/1100$	1.1865×10^{-3}	1.1836×10^{-2}	1.684
$T_{\max}/1200$	1.8866×10^{-3}	1.9106×10^{-2}	1.687
$T_{\max}/1300$	2.4809×10^{-3}	2.5706×10^{-2}	1.688
$T_{\max}/1400$	2.9918×10^{-3}	3.1732×10^{-2}	1.690
$T_{\max}/1500$	3.4356×10^{-3}	3.6977×10^{-2}	1.745

Acknowledgements

The authors are grateful to the anonymous reviewers for their very constructive comments and suggestions, which have helped to improve the paper considerably.

Funding

A.R. Appadu and K.M. Agbavon are grateful to the South African DST/NRF SARChI on Mathematical Models and Methods in Bioengineering and Biosciences (M^2B^2), Grant 82770 for funding.

Competing interests

None of the three authors has competing interests in the manuscript.

Consent for publication

The authors agree for the paper to be published.

Authors' contributions

Each of the authors contributed to the work and read and approved the final version of the manuscript.

Author details

¹Department of Mathematics and Applied Mathematics, University of Pretoria, Pretoria, South Africa. ²Department of Mathematics, University of South Africa, Florida, South Africa.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 8 November 2018 Accepted: 3 April 2019 Published online: 18 April 2019

References

- Li, S., Petzold, L., Ren, Y.: Stability of moving mesh systems of partial differential equations. *SIAM J. Sci. Comput.* **20**, 719–738 (1998)
- Qiu, Y., Sloan, D.M.: Numerical solution of Fisher's equation using a moving mesh method. *J. Comput. Phys.* **146**, 726–746 (1998)
- Ruxun, L., Mengping, Z., Ji, W., Xiao-Yuan, L.: The designing approach of difference schemes by controlling the remainder-effect. *Int. J. Numer. Methods Fluids* **31**, 523–533 (1999)
- Zeidan, D., Romenski, E., Slaouti, A., Toro, E.F.: Numerical study of wave propagation in compressible two-phase flow. *Int. J. Numer. Methods Fluids* **54**, 393–417 (2007)
- Zeidan, D.: Assessment of mixture two-phase flow equations for volcanic flows using Godunov-type methods. *Appl. Math. Comput.* **272**, 707–719 (2016)
- Goncalves da Silva, E., Zeidan, D.: Numerical simulation of unsteady cavitation in liquid hydrogen flows. *Int. J. Eng. Syst. Model. Simul.* **9**, 41 (2017)
- Minhajul, Zeidan, D., Raja Sekhar, T.: On the wave interactions in the drift-flux equations of two-phase flows. *Appl. Math. Comput.* **327**, 117–131 (2018)
- Chen, B.M., Kojouharov, H.V.: Non-standard numerical methods applied to subsurface biobarrier formation models in porous media. *Bull. Math. Biol.* **61**, 779–798 (1999)

9. Yatat, V., Couteron, P., Dumont, Y.: Spatially explicit modelling of tree-grass interactions in fire-prone savannas (a partial differential equations framework). *Ecol. Complex.* **36**, 290–313 (2018)
10. Doelman, A., Kaper, T.J., Zegeling, P.A.: Pattern formation in the one-dimensional Gray–Scott model. *Nonlinearity* **10**, 523–563 (1997)
11. Houdek, G., Balmforth, N.J., Christensen-Dalsgaard, J., Gough, D.O.: Amplitudes of stochastically excited oscillations in main-sequence stars (1999). ArXiv Preprint [Astro-ph/9909107](https://arxiv.org/abs/Astro-ph/9909107)
12. Hagberg, A., Meron, E.: From labyrinthine patterns to spiral turbulence. *Phys. Rev. Lett.* **72**, (1994)
13. Fisher, R.A.: The wave of advance of advantageous genes. *Ann. Hum. Genet.* **7**, 355–369 (1937)
14. Kolmogorov, A.N., Petrovsky, I.G., Piskunov, N.S.: Etude de l'équation de la diffusion avec croissance de la quantité de matière et son application à un problème biologique. *J. Mosc. Univ. Math. Bull.* **129**, 1–25 (1937)
15. Hagstrom, T., Keller, H.B.: The numerical calculation of traveling wave solutions of nonlinear parabolic equations. *SIAM J. Sci. Stat. Comput.* **7**, 978–988 (1986)
16. Gazzdag, J., Canosa, J.: Numerical solution of Fisher's equation. *J. Appl. Probab.* **11**, 445–457 (1974)
17. Canosa, J.: On a nonlinear diffusion equation describing population growth. *IBM J. Res. Dev.* **17**, 307–313 (1973)
18. Anguelov, R., Kama, P., Lubuma, J.M.S.: On non-standard finite difference models of reaction–diffusion equations. *J. Comput. Appl. Math.* **175**, 11–29 (2005)
19. Mulholland, L.S., Qiu, Y., Sloan, D.M.: Solution of evolutionary partial differential equations using adaptive finite differences with pseudospectral post-processing. *J. Comput. Phys.* **131**, 280–298 (1997)
20. Huang, W., Ren, Y., Russell, R.D.: Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM J. Numer. Anal.* **31**, 709–730 (1994)
21. Huang, W., Ren, Y., Russell, R.D.: Moving mesh methods based on moving mesh partial differential equations. *J. Comput. Phys.* **113**, 279–290 (1994)
22. Chen-Charpentier, B.M., Kojouharov, H.V.: An unconditionally positivity preserving scheme for advection–diffusion reaction equations. *Math. Comput. Model.* **57**, 2177–2185 (2013)
23. Durran, D.R.: *Numerical Methods for Fluid Dynamics with Applications to Geophysics*, vol. 32. Springer, Berlin (2010)
24. Taha, T.R., Ablowitz, M.I.: Analytical and numerical aspects of certain nonlinear evolution equations, III. Numerical, Korteweg–de Vries equation. *J. Comput. Phys.* **55**, 231–253 (1984)
25. Zabusky, N.J., Kruskal, M.D.: Interaction of solitons in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.* **15**, 240 (1965)
26. Appadu, A.R., Lubuma, J.M.S., Mphephu, N.: Computational study of three numerical methods for some linear and nonlinear advection–diffusion–reactions problems. *Prog. Comput. Fluid Dyn.* **17**, 114–129 (2017)
27. Mickens, R.E.: Exact solutions to a finite-difference model of a nonlinear reaction–advection equation: implications for numerical analysis. *Numer. Methods Partial Differ. Equ.* **5**, 313–325 (1989)
28. Mickens, R.E.: Dynamic consistency: a fundamental principle for constructing nonstandard finite difference schemes for differential equations. *J. Differ. Equ. Appl.* **11**, 645–653 (2005)
29. Mickens, R.E.: Nonstandard finite difference models of differential equations (1994)
30. Mickens, R.E.: Relation between the time and space step-sizes in nonstandard finite-difference schemes for the Fisher equation. *Numer. Methods Partial Differ. Equ.* **13**, 51–55 (1997)
31. Mickens, R.E.: Nonstandard finite difference schemes for differential equations. *J. Differ. Equ. Appl.* **8**, 823–847 (2002)
32. VonNeumann, J., Richtmyer, R.D.: A method for the numerical calculation of hydrodynamic shocks. *J. Appl. Phys.* **21**, 232–237 (1950)
33. Campbell, J.C., Shashkov, M.J.: A tensor artificial viscosity using a mimetic finite difference algorithm. *J. Comput. Phys.* **172**, 739–765 (2001)
34. Noh, W.F., Protter, M.H.: Difference methods and the equations of hydrodynamics. *J. Math. Mech.* **12**, 149–191 (1963)
35. Landshoff, R.: A numerical method for treating fluid flow in the presence of shocks (1955)
36. Wilkins, M.L.: Use of artificial viscosity in multidimensional fluid dynamic calculations. *J. Comput. Phys.* **36**, 281–303 (1980)
37. Kurapatenko, V.F.: Difference methods for solutions of problems of mathematical physics. *Am. Math. Soc.* (1967)
38. Caramana, E.J., Shashkov, M.J., Whalen, P.P.: Formulations of artificial viscosity for multi-dimensional shock wave computations. *J. Comput. Phys.* **144**, 70–97 (1998)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
