

Article

Solving the Advection Diffusion Reaction Equations by Using the Enhanced Higher-Order Unconditionally Positive Finite Difference Method

Ndivhuwo Ndou ^{1,2,*} , Phumlani Dlamini ¹ and Byron Alexander Jacobs ¹ 

¹ Department of Mathematics and Applied Mathematics, University of Johannesburg, P.O. Box 524, Auckland Park 2006, South Africa; pgdlamini@uj.ac.za (P.D.); byronj@uj.ac.za (B.A.J.)

² Department of Mathematical and Computational Sciences, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa

* Correspondence: 220164883@student.uj.ac.za or ndivhuwo.ndou@univen.ac.za

Abstract: In this paper, the enhanced higher-order unconditionally positive finite difference method is developed to solve the linear, non-linear and system advection diffusion reaction equations. Investigation into the effectiveness and efficiency of the proposed method is carried out by calculating the convergence rate, error and computational time. A comparison of the solutions obtained by the enhanced higher-order unconditionally positive finite difference and exact solution is conducted for validation purposes. The numerical results show that the developed method reduced the time taken to solve the linear and non-linear advection diffusion reaction equations as compared to the results obtained by the higher-order unconditionally positive finite difference method.

Keywords: enhanced higher-order unconditionally positive finite difference method; unconditionally positive finite difference method; advection diffusion reaction equations; von Neumann stability analysis; consistency; stability analysis

MSC: 65M12; 65N06; 65N12



Citation: Ndou, N.; Dlamini, P.; Jacobs, B.A. Solving the Advection Diffusion Reaction Equations by Using the Enhanced Higher-Order Unconditionally Positive Finite Difference Method. *Mathematics* **2024**, *12*, 1009. <https://doi.org/10.3390/math12071009>

Academic Editor: Nikolaos L. Tsitsas

Received: 20 February 2024

Revised: 18 March 2024

Accepted: 22 March 2024

Published: 28 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Numerical methods typically require large mesh sizes to give better approximations to solutions of differential equations. Consequently, their memory requirements are huge and, therefore, computational efficiency is affected. In this work, we aim to improve the computational efficiency of the higher-order unconditionally positive finite difference method (HUPFD). The HUPFD is based on using higher-order finite difference schemes to solve differential equations, with its main attribute being the preservation of the positivity of solutions. The positivity of solutions is essential in several physical settings—for example, quantities like population sizes, chemical species concentration and neutron numbers require positive solutions in order to be meaningful.

One of the techniques researchers employ to reduce the computational span of computational methods is the proper orthogonal decomposition method (POD) [1]. This has previously been used to reduce the dimensions of numerical methods such as non-standard finite difference and Crank–Nicolson to solve problems such as the Sobolev, hyperbolic, Burgers and Navier–Stokes equations [2–4]. In this paper, we couple the POD and HUPFD to obtain a new enhanced hybrid method that we name the enhanced higher-order unconditionally positive finite difference method (EHUPFD). By utilizing the POD, the EHUPFD extracts a set of basis functions from the HUPFD solution called the snapshot matrix and then uses a small subset of leading basis functions to construct state variable approximations.

We test the applicability of the EHUPFD on the advection diffusion reaction equation below

$$\begin{aligned}
\frac{\partial u}{\partial t} + U_a \frac{\partial u}{\partial x} - D \frac{\partial^2 u}{\partial x^2} &= f(t, x, u), \quad a \leq x \leq b \quad t > 0, \quad x \in \partial\Omega \\
u(0, x) &= u_0(x) \geq 0, \\
u(a, t) &= u_a(t) \quad u(b, t) = u_b(t),
\end{aligned} \tag{1}$$

where D is the diffusion coefficient, U_a is the advection coefficient, $u_x(t, x)$ is the advection term, $u_{xx}(t, x)$ is the diffusion term, $u_t(t, x)$ represent the rate of change of concentration of substances and $f(t, x, u)$ is the source or reaction term [5]. The ADR equation governs the process of diffusion and advection simultaneously [6]. The ADR equation is used to model the exponential traveling waves semiconductor modeling, biological system modeling, pollutant absorption in soil and neutron diffusion [6–10]. The UPFD method has previously been used to solve the ADR equation, amongst other known numerical methods reported in the literature, such as the standard finite difference, Crank–Nicolson finite difference, and non-standard finite difference methods and acts. The UPFD method has been proven to be a stable numerical method, accurate and consistent when solving the ADR equation in rectangular domains. The method is also known to preserve the positivity of solution regardless of step size [11]. For example, Appadu, A and Rao used UPFD to approximate the solution to the linear ADR equation, which models exponential traveling waves. The authors studied various scenarios in which advection or diffusion is dominating, as well as when both are equal. In this study, they discovered that the UPFD scheme is successful in solving the ADR equation when advection or diffusion factors are dominating [12]. However, as always, there is a need to constantly refine the numerical methods in order to improve the method's accuracy, computational time and convergence rate.

The HUPFD method was developed to solve the advection diffusion reaction equations to improve the solution's accuracy [13]. The higher-order numerical methods have been used to solve mathematical problems such as [13–20]. The standard finite difference methods, such as the backward, forward and central difference schemes, struggle to approach the exact solution of most partial differential models with large step sizes. As a result, relatively small step sizes are chosen to improve the accuracy of the solution to the problem at the cost of increased computational time due to the increased number of grid points. With a smaller grid size, HUPFD calculates a more accurate solution to the ADR equation's linear, non-linear and system versions compared to the UPFD results.

The highly accurate finite difference method has gained much attention in recent years [21–25]. Gemechis FD and Tesfaye AB studied the higher-order numerical method to solve the singularly perturbed delay reaction–diffusion equation [26]. In order to achieve the six higher-order schemes, they used the Richardson extrapolation technique [26]. They found that the developed method is stable and consistent, and produces a more accurate solution than some of the methods reported in the literature. Ranjana KM and Venn G investigated the fourth-order finite difference method based on spline in tension approximation for the solution of second-order quasi-linear hyperbolic equations in one space dimension [27]. They proposed a three-level implicit nine-point compact finite difference formulation of order two in the time and four in the space directions based on split tension approximation in the t-direction for the numerical solution of space dimensional second-order quasi-linear hyperbolic equations with a first-order space derivative term. The proposed numerical method for the wave equation in polar coordinates is conditionally stable [27]. In contrast, the method for the damped wave and telegraphic equations is unconditionally stable. Sari M et al. [28] proposed a tenth-order finite difference scheme for solving a one-dimensional diffusion equation. Gurarslan et al. [29] presented a sixth-order compact difference scheme and a fourth-order Runge–Kutta scheme in time to produce numerical solutions of the one-dimensional advection–diffusion equation, which was found to be accurate in solving the concurrent transport equation for the one-dimensional advection diffusion reaction equation.

The rest of the sections in the paper are organized as follows. Sections 2 and 3 introduce the UPFD and HUPFD, respectively. Subsequently, Sections 4 and 5 apply the schemes in Sections 2 and 3 to solve the ADR equations. Section 6 introduces the POD technique used to enhance the HUPFD, which results in the development of EHUPFD. The results obtained by the EHUPFD are illustrated in Section 6 and a summary of the results is provided in Section 7.

2. The Unconditionally Positive Finite Difference Scheme

The UPFD schemes are presented below [30],

$$\frac{\partial u}{\partial x} \approx \begin{cases} \frac{u_i^n - u_{i-1}^{n+1}}{\Delta x} & (2a) \\ \frac{u_i^{n+1} - u_{i-1}^n}{\Delta x} & (2b) \end{cases}$$

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}, \tag{3}$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^n - 2u_i^{n+1} + u_{i-1}^n}{(\Delta x)^2}, \tag{4}$$

where grid points are calculated by using the formula $x_i = a + \Delta x(i - 1)$ for $1 \leq i \leq m$ and spatial step size $\Delta x = \frac{b - a}{m - 1}$. The time step is calculated by using the formula $\Delta t = \frac{T}{N - 1}$ with grid points given by $t_n = \Delta t(n - 1)$, for $1 \leq n \leq N$. Equation (2a) is used when the coefficient of the first space derivative is negative, whereas Equation (2b) is used when the first space derivative term is positive. Equation (4) is used when the second space derivative is negative or positive.

3. The Higher-Order Unconditionally Positive Finite Difference Schemes for Coupling with POD

In this section, we present the HUPFD that will be coupled with the POD to develop the EHUPFD. The EHUPFD is then tested on the ADR equations.

HUPFD

In this case, we consider the first-order formula (2a) and (2b) for the first space derivative, (3) for the first time derivative, and the fourth order for the second space derivative approximations to obtain the higher-order schemes as given below,

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{-u_{i+2}^{n+1} + 16u_{i+1}^n - 30u_i^{n+1} + 16u_{i-1}^n - u_{i-2}^{n+1}}{12\Delta x^2}, \tag{5}$$

where the spatial step size $\Delta x = \frac{b - a}{m - 1}$ and the grid points are given as $x_i = a + \Delta x(i - 1)$ and $2 \leq i \leq m + 1$. The time step size $\Delta t = \frac{T}{N - 1}$, with the corresponding grid points given as $t_n = \Delta t(n - 1)$, $1 \leq n \leq N$.

In the development of the higher-order unconditionally positive finite difference (HUPFD), the goal is to produce a higher-order method, based on the principles of the UPFD, that will still preserve the positivity of solutions. We test the effectiveness of the scheme on the ADR equations. In practice, selecting different orders in finite difference schemes for space derivatives involves a tradeoff between accuracy, stability, consistency and computational cost. Engineers and scientists choose different orders of the schemes based on the specific requirements of their simulations and the constraints of their computational resources, which is often about achieving accuracy. In this study, the HUPFD

schemes are constructed by using different orders in finite difference schemes for the space derivative scheme with the idea of investigating a combination that gives better results.

4. Implementation of the HUPFD Schemes on the Linear ADR Equation

In this section, we set $f(t, x, u) = -u$, $U_a = 1$ and $D = 1$ in Equation (1) to obtain the following equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} - u, \tag{6}$$

The boundary and initial conditions used in this problem are given below

$$\begin{cases} u(x, 0) = e^{-x}, & 0 \leq x \leq 10 \\ u(0, t) = e^t, & 0 \leq t \leq 0.85 \\ u_x(10, t) = -u(10, t), & 0 \leq t \leq 0.85 \end{cases} \tag{7}$$

with the exact solution given by $u(x, t) = e^{(t-x)}$ [30].

4.1. Solving the Linear ADR Equation Using the HUPFD Scheme

Because the coefficient of $\frac{\partial u}{\partial x}$ in Equation (6) is positive, we find the UPFD scheme using Equations (2b), (3) and (5). As a result, the following equation gives the HUPFD discretization of Equation (6)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{u_i^{n+1} - u_{i-1}^n}{\Delta x} = \frac{-u_{i+2}^{n+1} + 16u_{i+1}^n - 30u_i^{n+1} + 16u_{i-1}^n - u_{i-2}^{n+1}}{12\Delta x^2} - u_i^{n+1}, \tag{8}$$

which simplifies to the following

$$\begin{aligned} \frac{1}{12\Delta x^2}u_{i-2}^{n+1} + \left(\frac{1}{\Delta t} + \frac{1}{\Delta x} + \frac{30}{12\Delta x^2} + 1\right)u_i^{n+1} \\ + \frac{1}{12\Delta x^2}u_{i+2}^{n+1} = \frac{1}{\Delta t}u_i^n + \left(\frac{1}{\Delta x} + \frac{16}{12Dx^2}\right)u_{i-1}^n + \frac{16}{12\Delta x^2}u_{i+1}^n, \end{aligned} \tag{9}$$

let $p_1 = \frac{1}{\Delta t} + \frac{1}{\Delta x} + \frac{30}{12\Delta x^2} + 1, p_2 = \frac{1}{12\Delta x^2}, p_3 = \frac{1}{2\Delta t}, p_4 = \frac{1}{\Delta x} + \frac{16}{12\Delta x^2}$, thus

$$p_1u_i^{n+1} + p_2u_{i+2}^{n+1} + p_2u_{i-2}^{n+1} = p_3u_i^n + p_4u_{i-1}^n + 16p_2u_{i+1}^n, \tag{10}$$

Equation (10) is simplified to the following

$$Au^{n+1} = Bu^n + d^n, \tag{11}$$

where

$$Au^{n+1} = \begin{bmatrix} p_1 & 0 & p_2 & 0 & 0 & \cdots & 0 \\ 0 & p_1 & 0 & p_2 & 0 & \cdots & \vdots \\ p_2 & 0 & p_1 & 0 & p_2 & \ddots & \vdots \\ 0 & p_2 & 0 & p_1 & 0 & \ddots & 0 \\ 0 & 0 & p_2 & 0 & p_1 & \ddots & p_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & p_2 & 0 & p_1 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ u_{M-1}^{n+1} \\ u_M^{n+1} \end{bmatrix}, \tag{12}$$

$$Bu^n = \begin{bmatrix} p_3 & 16p_2 & 0 & 0 & 0 & \dots & 0 \\ p_4 & p_3 & 16p_2 & 0 & 0 & \dots & \vdots \\ 0 & p_4 & p_3 & 16p_2 & 0 & \ddots & \vdots \\ 0 & 0 & p_4 & p_3 & 16p_2 & \ddots & 0 \\ 0 & 0 & 0 & p_4 & p_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 16p_2 \\ 0 & \dots & \dots & \dots & 0 & p_4 & p_3 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ \vdots \\ u_{M-1}^n \\ u_M^n \end{bmatrix}, \tag{13}$$

and

$$d^n = p_3 \begin{bmatrix} u_0^n \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_M^n \end{bmatrix} - p_2 \begin{bmatrix} u_0^{n+1} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_{M+1}^{n+1} \end{bmatrix}, \tag{14}$$

where u is the unknown vector of the order $M \times 1$ at any time level t^{n+1} , and A and B are the coefficient square matrix of order $M \times M$ with a triangular structure.

4.2. Solving the Non-Linear ADR Equation Using the HUPFD Scheme

In this section, the EHUPFD is used to solve the non-linear advection diffusion reaction equation given below, with $f(t, x, u) = ru(1 - u)$ and $U_a = 0$

$$\frac{\partial u}{\partial t} - D \frac{\partial^2 u}{\partial x^2} = ru(1 - u), \tag{15}$$

The boundary and initial conditions of (15) are given below

$$u(x, 0) = \begin{cases} 0.7e^{-\sigma(x-x_0)^2}, & |x - x_0| \leq 0.06 \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$u(0, t) = 0, \quad 0 \leq t \leq T$$

$$u(1, t) = 0,$$

where $\sigma = 80$ and $x_0 = 0.5$ [30].

The exact solution is given as follows

$$u(t, x) = \frac{1}{\left[1 + \left(\frac{1 - \sqrt{u(x, 0)}}{\sqrt{u(x, 0)} \exp\left(\frac{1}{6D}\sqrt{6Drx}\right)} \right) \exp\left(-\frac{5}{6}rt + \frac{1}{6D}\sqrt{6Drx}\right) \right]^2}, \tag{17}$$

where $D = 0.0002$ and $r = 0.05$.

The discretization of the non-linear ADR using Equations (2b), (3) and (15) is given below

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} - D \frac{-u_{i+2}^{n+1} + 16u_{i+1}^n - 30u_i^{n+1} + 16u_{i-1}^n - u_{i-2}^{n+1}}{12\Delta x^2} = ru_i^n - u_{max}u_i^{n+1}, \tag{18}$$

where u_{max} is a frozen coefficient. By simplifying Equation (18), the following is obtained

$$\left(\frac{1}{2\Delta t} + \frac{30}{12(\Delta x)^2} + ru_{max}\right)u_i^{n+1} + \frac{D}{12(\Delta x)^2}(u_{i+2}^{n+1} + u_{i-2}^{n+1}) = \frac{D}{12(\Delta x)^2}(16u_{i+1}^n + 16u_{i-1}^n) + ru_i^n + \frac{1}{2\Delta t}u_i^{n-1},$$

Let $s_1 = \frac{1}{2\Delta t} + \frac{30}{12\Delta x} + ru_{max}$, $s_2 = \frac{D}{12(\Delta x)^2}$ and $s_3 = \frac{1}{2\Delta t}$; thus,

$$s_2u_{i-2}^{n+1} + s_1u_i^{n+1} + s_2u_{i+2}^{n+1} = 16s_2u_{i-1}^n + ru_i^n + 16s_2u_{i+1}^n + s_3u_i^{n-1}, \tag{19}$$

Equation (19) is simplified to the following equation

$$Au^{n+1} = Bu^n + d^n \tag{20}$$

where

$$Au^{n+1} = \begin{bmatrix} s_1 & 0 & s_2 & 0 & 0 & \cdots & 0 \\ 0 & s_1 & 0 & s_2 & 0 & \cdots & \vdots \\ s_2 & 0 & s_1 & 0 & s_2 & \ddots & \vdots \\ 0 & s_2 & 0 & s_1 & 0 & \ddots & 0 \\ 0 & 0 & s_2 & 0 & s_1 & \ddots & s_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & s_2 & s_1 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ u_{M-1}^{n+1} \\ u_M^{n+1} \end{bmatrix}, \tag{21}$$

$$Bu^n = \begin{bmatrix} r & 16s_2 & 0 & 0 & 0 & \cdots & 0 \\ 16s_2 & r & 16s_2 & 0 & 0 & \cdots & \vdots \\ 0 & 16s_2 & r & 16s_2 & 0 & \ddots & \vdots \\ 0 & 0 & 16s_2 & r & 16s_2 & \ddots & 0 \\ 0 & 0 & 0 & 16s_2 & r & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 16s_2 \\ 0 & \cdots & \cdots & \cdots & 0 & 16s_2 & r \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ \vdots \\ \vdots \\ u_{M-1}^n \\ u_M^n \end{bmatrix}, \tag{22}$$

and

$$d^n = s_3 \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ \vdots \\ \vdots \\ u_{M-1}^n \\ u_M^n \end{bmatrix}. \tag{23}$$

4.3. Implementation of the UPFD and the HUPFD Schemes on the Linear Schnakenberg Model

The linear Schnakenberg model is given by the equation below

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = (q_1 - 1)u + (q_2 + s)v + \mu_1 \frac{\partial^2 u}{\partial x^2}, \tag{24}$$

$$\frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} = -q_1 u - q_2 v + \mu_2 \frac{\partial^2 v}{\partial x^2}, \tag{25}$$

subject to the boundary and initial conditions given below

$$\begin{cases} u(x, 0) = e^{-x}, & 0 \leq x \leq 10 \\ u(0, t) = e^t, & 0 \leq t \leq 0.85 \\ u_x(10, t) = -u(10, t), & 0 \leq t \leq 0.85 \end{cases} \tag{26}$$

and

$$\begin{cases} v(x, 0) = e^{-x}, & 0 \leq x \leq 10 \\ v(0, t) = e^t, & 0 \leq t \leq 0.85 \\ v_x(10, t) = -v(10, t), & 0 \leq t \leq 0.85 \end{cases} \tag{27}$$

with constants $s = 5, c = 0, q_1 = q_2 = 0, \mu_1 = 1$ and $\mu_2 = 5$.

4.3.1. Solving the Linear Schnakenberg Model Using the UPFD Scheme

The UPFD scheme is applied to the linear Schnakenberg model and the discretization of the scheme is given below

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^{n+1} - u_i^n}{\Delta x} = (q_1 - 1)u_i^{n+1} + (q_2 + s)v_i^n + \mu_1 \left(\frac{u_{i+1}^n - 2u_i^{n+1} + u_{i-1}^n}{(\Delta x)^2} \right), \tag{28}$$

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} + c \frac{v_i^{n+1} - v_i^n}{\Delta x} = -q_1 u_i^{n+1} - q_2 v_i^{n+1} + \mu_2 \left(\frac{v_{i+1}^n - 2v_i^{n+1} + v_{i-1}^n}{(\Delta x)^2} \right), \tag{29}$$

which simplifies to the following

$$\begin{aligned} \left(\frac{1}{\Delta t} + \frac{c}{\Delta x} + (q_1 - 1) + \frac{2\mu_1}{(\Delta x)^2} \right) u_i^{n+1} &= (q_2 + s)v_i^n + \frac{\mu_1}{\Delta x} (u_{i+1}^n + u_{i-1}^n) + \frac{c}{\Delta x} u_i^n \\ &+ \frac{1}{\Delta t} u_i^n, \end{aligned} \tag{30}$$

$$\begin{aligned} \left(\frac{1}{\Delta t} + \frac{c}{\Delta x} + q_2 + \frac{2\mu_2}{(\Delta x)^2} \right) u_i^{n+1} &= -q_1 u_i^{n+1} + \frac{1}{\Delta t} v_i^n + \frac{c}{\Delta x} v_i^n \\ &+ \frac{\mu_2}{(\Delta x)^2} (v_{i+1}^n + v_{i-1}^n), \end{aligned} \tag{31}$$

Let $s_1 = \left(\frac{1}{\Delta t} + \frac{c}{\Delta x} + (q_1 - 1) + \frac{2\mu_1}{(\Delta x)^2} \right)$ and $s_2 = \left(\frac{1}{\Delta t} + \frac{c}{\Delta x} + q_2 + \frac{2\mu_2}{(\Delta x)^2} \right)$; thus,

$$u_i^{n+1} = \frac{(q_2 + s)}{s_1} v_i^n + \frac{\mu_1}{\Delta x s_1} (u_{i+1}^n + u_{i-1}^n) + \frac{c}{\Delta x s_1} u_i^n + \frac{1}{\Delta t s_1} u_i^n, \tag{32}$$

$$u_i^{n+1} = -\frac{q_1}{s_2} u_i^{n+1} + \frac{1}{\Delta t s_2} v_i^n + \frac{c}{\Delta x s_2} v_i^n + \frac{\mu_2}{(\Delta x)^2 s_2} (v_{i+1}^n + v_{i-1}^n), \tag{33}$$

4.3.2. Prove the Stability of the System ADR Equation with the UPFD

By discretizing (54) and (55) by using the UPFD scheme, we obtain the following

$$\begin{aligned} \left(\frac{1}{\Delta t} + (1 - q_1) + \frac{2\mu_1}{(\Delta x)^2} \right) u_i^{n+1} + \frac{c}{\Delta x} u_{i+1}^{n+1} &= q_2 v_i^n + \frac{\mu_1}{(\Delta x)^2} (u_{i+1}^n + u_{i-1}^n) \\ &+ \frac{1}{\Delta t} u_i^n + \frac{c}{\Delta x} u_i^n, \end{aligned} \tag{34}$$

$$\left(\frac{1}{\Delta t} + q_2 + \frac{2\mu_2}{(\Delta x)^2}\right)v_i^{n+1} + \frac{c}{\Delta x}v_{i+1}^{n+1} = -q_1u_i^{n+1} + \frac{1}{\Delta t}v_i^n + \frac{c}{\Delta x}v_i^n + \frac{\mu_2}{(\Delta x)^2}(v_{i+1}^n + v_{i-1}^n), \tag{35}$$

By using the parameters $s = 5, c = 0, q_1 = q_2 = 0, \mu_1 = 1$ and $\mu_2 = 5$, Equations (34) and (35) are simplified to

$$\left(\frac{1}{\Delta t} + 1 + \frac{2}{(\Delta x)^2}\right)u_i^{n+1} = \frac{1}{(\Delta x)^2}(u_{i+1}^n + u_{i-1}^n) + \frac{1}{\Delta t}u_i^n \tag{36}$$

$$\left(\frac{1}{\Delta t} + \frac{10}{(\Delta x)^2}\right)v_i^{n+1} = \frac{1}{\Delta t}v_i^n + \frac{5}{(\Delta x)^2}(v_{i+1}^n + v_{i-1}^n) \tag{37}$$

The von Neumann stability analysis, $u_i^n = \zeta^n e^{j\Delta t i \Delta x}$, is used to check the stability region in Equations (36) and (37) to obtain the following

$$\left(\frac{1}{\Delta t} + 1 + \frac{2}{(\Delta x)^2}\right)\zeta^{n+1} e^{j\Delta t i \Delta x} = \frac{1}{(\Delta x)^2}(\zeta^n e^{j\Delta t(i+1)\Delta x} + \zeta^n e^{j\Delta t(i-1)\Delta x}) + \frac{1}{\Delta t}\zeta^n e^{j\Delta t i \Delta x}, \tag{38}$$

$$\left(\frac{1}{\Delta t} + \frac{10}{(\Delta x)^2}\right)\zeta^{n+1} e^{j\Delta t i \Delta x} = \frac{1}{\Delta t}\zeta^n e^{j\Delta t i \Delta x} + \frac{5}{(\Delta x)^2}(\zeta^n e^{j\Delta t(i+1)\Delta x} + \zeta^n e^{j\Delta t(i-1)\Delta x}), \tag{39}$$

Equations (38) and (39) are simplified to obtain the following

$$\zeta = \frac{\frac{2\mu_1}{(\Delta x)^2} \cos(\omega) + \frac{1}{\Delta t}}{\frac{1}{\Delta t} + 1 + \frac{2}{(\Delta x)^2}}, \tag{40}$$

and

$$\zeta = \frac{\frac{1}{\Delta t} + \frac{10 \cos(\omega)}{(\Delta x)^2}}{\frac{1}{\Delta t} + \frac{10}{(\Delta x)^2}}, \tag{41}$$

where $\omega = \Delta t \Delta x$. By using the inequality $|\zeta| \leq 1$, when $\omega = \pi$, we obtain the following, respectively

$$\Delta t > -2. \tag{42}$$

4.3.3. Proof of Consistency with the System ADR Equation by Using UPFD Scheme

By discretizing (24) and (25) by using the UPFD scheme, we obtain the following

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_{i+1}^{n+1} - u_i^2}{\Delta x} = (q_1 - 1)u_i^{n+1} + (q_2 + s)v_i^n + \frac{\mu_1}{(\Delta x)^2}(u_{i+1}^n - 2u_i^{n+1} + u_{i-1}^n), \tag{43}$$

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} + c \frac{v_{i+1}^{n+1} - v_i^2}{\Delta x} = -q_1u_i^{n+1} - q_2v_i^n + \frac{\mu_2}{(\Delta x)^2}(v_{i+1}^n - 2v_i^{n+1} + v_{i-1}^n). \tag{44}$$

From (43) and (44), we obtain the following Taylor’s expansions

$$u_i^{n+1} \approx u_i^n + \Delta t \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} + \frac{(\Delta t)^3}{6} \frac{\partial^3 u}{\partial t^3} + \dots \tag{45}$$

$$u_i^{n-1} \approx u_i^n - \Delta t \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} - \frac{(\Delta t)^3}{6} \frac{\partial^3 u}{\partial t^3} + \dots \tag{46}$$

$$u_{i-1}^n \approx u_i^n - \Delta x \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u}{\partial x^2} - \frac{(\Delta x)^3}{6} \frac{\partial^3 u}{\partial x^3} + \dots \tag{47}$$

$$u_{i+1}^n \approx u_i^n + \Delta x \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u}{\partial x^2} + \frac{(\Delta x)^3}{6} \frac{\partial^3 u}{\partial x^3} + \dots \tag{48}$$

$$u_{i-2}^{n+1} \approx u_i^n + \Delta t \frac{\partial u}{\partial t} - 2\Delta x \frac{\partial u}{\partial x} + 2(\Delta x)^2 \frac{\partial^2 u}{\partial x^2} - 2(\Delta t \Delta x) \frac{\partial^2 u}{\partial t \partial x} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} + \dots \quad (49)$$

$$u_{i+2}^{n+1} \approx u_i^n + \Delta t \frac{\partial u}{\partial t} + 2\Delta x \frac{\partial u}{\partial x} + 2(\Delta x)^2 \frac{\partial^2 u}{\partial x^2} + 2(\Delta t \Delta x) \frac{\partial^2 u}{\partial t \partial x} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} + \dots \quad (50)$$

By applying Taylor’s expansions to Equations (43) and (44), we obtain the following simplified equations

$$\begin{aligned} (1 - q_1)u_i^n + \left(1 + (1 - q_1)\Delta t \frac{2\mu_1}{\Delta x^2} \Delta t + \frac{c\Delta t}{\Delta x}\right) \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} \\ + \left(\frac{\Delta t}{2} + (1 - q_1) \frac{(\Delta t)^2}{2} + \frac{2\mu_1}{(\Delta x)^2} + \frac{2\mu_1(\Delta t)^3}{6(\Delta x)^2} + \frac{c(\Delta t)}{\Delta x}\right) \frac{\partial^2 u}{\partial t^2} \\ + 2c\Delta t \frac{\partial^2 u}{\partial t \partial x} - q_2 v_i^n - \mu_1 \frac{\partial^2 u}{\partial x^2} = 0, \end{aligned} \quad (51)$$

and

$$\begin{aligned} \left(\frac{1}{\Delta t} + q_2 + \frac{2\mu_2}{(\Delta x)^2} + \frac{c}{\Delta x}\right) v_i^n + \left(1 + q_2 \Delta t + \frac{2\mu_2 \Delta t}{(\Delta x)^2} + \frac{c\Delta t}{\Delta x}\right) \frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} \\ + \left(\frac{\Delta t}{2} + \frac{q_2(\Delta t)^2}{2} + \frac{\mu_2(\Delta t)^2}{(\Delta x)^2} + \frac{c(\Delta t)^2}{2(\Delta x)}\right) \frac{\partial^2 v}{\partial t^2} + 2\Delta t \frac{\partial^2 v}{\partial t \partial x} + \frac{c(\Delta x)}{2} \frac{\partial^2 v}{\partial x^2} \\ = -1_1 u_i^n + \left(\frac{1}{\Delta t} + \frac{2\mu_2}{(\Delta x)^2}\right) v_i^n - q_1 \Delta t \frac{\partial u}{\partial t} - q_1 \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} \\ - \frac{q_1(\Delta t)^3}{6} \frac{\partial^3 u}{\partial t^3} + \mu_2 \frac{\partial^2 v}{\partial x^2} - \frac{c}{\Delta x} v_i^n, \end{aligned} \quad (52)$$

Let $\Delta t = (\Delta x)^3$, which leads to Equations (51) and (52) being written as follows

$$\begin{aligned} (1 - q_1)u_i^n + \left(1 + (1 - q_1)(\Delta x)^3 + 2\mu_1 \Delta x + c(\Delta x)^2\right) \frac{\partial u}{\partial x} \\ + c \frac{\partial u}{\partial x} + \left(\frac{(\Delta x)^3}{2} + (1 - q_1) \frac{(\Delta x)^6}{2} + \frac{\mu_1(\Delta x)^4}{3} + c(\Delta x)^2\right) \frac{\partial^2 u}{\partial t^2} \\ + 2c(\Delta x)^3 \frac{\partial^2 u}{\partial t \partial x} - q_2 v_i^n - \mu_1 \frac{\partial^2 u}{\partial x^2} = 0, \end{aligned}$$

and

$$\begin{aligned} q_2 v_i^n + q_1 u_i^n + \left(1 + q_2(\Delta x)^3 + 2\mu_2 \Delta x + c(\Delta x)^2 + q_1(\Delta x)^3\right) \frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} \\ + \left(\frac{(\Delta x)^3}{2} + q_2 \frac{(\Delta x)^6}{2} + \mu_2(\Delta x)^4 + d \frac{(\Delta x)^5}{2} + q_1 \frac{(\Delta x)^6}{2}\right) \frac{\partial^2 v}{\partial t^2} \\ + 2\Delta t \frac{\partial^2 v}{\partial t \partial x} + \left(\frac{c(\Delta x)}{2} - \mu_2\right) = 0, \end{aligned} \quad (53)$$

When $\Delta x \rightarrow 0$, we obtain the original system ADR equations

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = (q_1 - 1)u + (q_2 + s)v + \mu_1 \frac{\partial^2 u}{\partial x^2} \quad (54)$$

$$\frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} = -q_1 u - q_2 v + \mu_2 \frac{\partial^2 v}{\partial x^2}, \quad (55)$$

4.3.4. Solving the Schnakenberg Model Using the HUPFD Scheme

The HUPFD scheme is applied to the linear Schnakenberg model and the discretization of the scheme is given below

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^{n+1} - u_{i-1}^n}{\Delta x} = (q_1 - 1)u_i^{n+1} + (q_2 + s)v_i^n, \\ + \mu_1 \left(\frac{-u_{i+2}^{n+1} + 16u_{i+1}^n - 30u_i^{n+1} + 16u_{i-1}^n - u_{i-2}^{n+1}}{12(\Delta x)^2} \right), \tag{56}$$

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} + c \frac{v_i^{n+1} - v_{i-1}^n}{\Delta x} = -q_1 u_i^{n+1} - q_2 v_i^{n+1}, \\ + \mu_2 \left(\frac{-v_{i+2}^{n+1} + 16v_{i+1}^n - 30v_i^{n+1} + 16v_{i-1}^n - v_{i-2}^{n+1}}{12(\Delta x)^2} \right), \tag{57}$$

which are simplified to the following

$$d_1 u_i^{n+1} + d_2 u_{i+2}^{n+1} + d_2 u_{i-2}^{n+1} = d_4 u_i^n + (q_2 + s)v_i^n + 16d_2 u_{i+1}^n + d_3 u_{i-1}^n, \tag{58}$$

$$p_1 v_i^{n+1} + p_2 v_{i+2}^{n+1} + p_2 v_{i-2}^{n+1} + q_1 u_i^{n+1} = d_4 v_i^{n+1} + p_3 v_{i-1}^n + 16p_2 v_{i+1}^n, \tag{59}$$

where $a_1 = \frac{1}{\Delta t} + \frac{c}{\Delta x} + \frac{30\mu_1}{12(\Delta x)^2}$, $a_2 = \frac{\mu_1}{12(\Delta x)^2}$, $a_3 = \frac{c}{\Delta x} + \frac{16\mu_1}{12(\Delta x)^2}$, $a_4 = \frac{1}{\Delta t}$, $p_1 = \frac{1}{\Delta t} + \frac{c}{\Delta x} + q_2 + \frac{30\mu_2}{12(\Delta x)^2}$, $p_2 = \frac{\mu_2}{12(\Delta x)^2}$ and $p_3 = \frac{c}{\Delta x} + \frac{16\mu_2}{12(\Delta x)^2}$

Equations (58) and (59) are simplified to the following

$$A_1 u^{n+1} = B_1 u^n + d_1^n \text{ and } A_2 u^{n+1} = B_2 u^n + d_2^n, \text{ respectively} \tag{60}$$

where

$$A_1 u^{n+1} = \begin{bmatrix} a_1 & 0 & a_2 & 0 & 0 & \dots & 0 \\ 0 & a_1 & 0 & a_2 & 0 & \dots & \vdots \\ a_2 & 0 & a_1 & 0 & a_2 & \ddots & \vdots \\ 0 & a_2 & 0 & a_1 & 0 & \ddots & 0 \\ 0 & 0 & a_2 & 0 & a_1 & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \dots & a_2 & 0 & a_1 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ \vdots \\ u_{M-1}^{n+1} \\ u_M^{n+1} \end{bmatrix}, \tag{61}$$

$$B_1 u^n = \begin{bmatrix} a_4 & 16a_2 & 0 & 0 & 0 & \dots & 0 \\ a_3 & a_4 & 16a_2 & 0 & 0 & \dots & \vdots \\ 0 & a_3 & a_4 & 16a_2 & 0 & \ddots & \vdots \\ 0 & 0 & a_3 & a_4 & 16a_2 & \ddots & 0 \\ 0 & 0 & 0 & d_3 & a_4 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 16a_2 \\ 0 & \dots & \dots & \dots & 0 & a_3 & a_4 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ \vdots \\ u_{M-1}^n \\ u_M^n \end{bmatrix}, \tag{62}$$

and

$$d_1^n = a_2 \begin{bmatrix} u_0^n \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_M^n \end{bmatrix} - a_3 \begin{bmatrix} u_0^{n+1} \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_{M+1}^{n+1} \end{bmatrix}, \tag{63}$$

and

$$A_2 v^{n+1} = \begin{bmatrix} p_1 & 0 & p_2 & 0 & 0 & \cdots & 0 \\ 0 & p_1 & 0 & p_2 & 0 & \cdots & \vdots \\ p_2 & 0 & p_1 & 0 & p_2 & \ddots & \vdots \\ 0 & p_2 & 0 & p_1 & 0 & \ddots & 0 \\ 0 & 0 & p_2 & 0 & p_1 & \ddots & p_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & p_2 & 0 & p_1 \end{bmatrix} \begin{bmatrix} v_1^{n+1} \\ v_2^{n+1} \\ v_3^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ v_{M-1}^{n+1} \\ v_M^{n+1} \end{bmatrix}, \tag{64}$$

$$B_2 v^n = \begin{bmatrix} d_4 & 16p_2 & 0 & 0 & 0 & \cdots & 0 \\ p_3 & d_4 & 16p_2 & 0 & 0 & \cdots & \vdots \\ 0 & p_3 & d_4 & 16p_2 & 0 & \ddots & \vdots \\ 0 & 0 & p_3 & d_4 & 16p_2 & \ddots & 0 \\ 0 & 0 & 0 & p_3 & d_4 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 16p_2 \\ 0 & \cdots & \cdots & \cdots & 0 & p_3 & d_4 \end{bmatrix} \begin{bmatrix} v_1^n \\ v_2^n \\ v_3^n \\ \vdots \\ \vdots \\ \vdots \\ v_{M-1}^n \\ v_M^n \end{bmatrix}, \tag{65}$$

and

$$d_2^n = a_2 \begin{bmatrix} v_0^n \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v_M^n \end{bmatrix} - a_3 \begin{bmatrix} v_0^{n+1} \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v_{M+1}^{n+1} \end{bmatrix}, \tag{66}$$

respectively, where u is the unknown vector of the order $M \times 1$ at any time level t^{n+1} , and A and B are the coefficient square matrix of order $M \times M$ with a triangular structure.

4.3.5. Prove Stability When Solving System ADR Equation by the HUPFD

By discretizing (56) and (57) by using the UPFD scheme, we obtain the following

$$\begin{aligned} \frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_{i+1}^{n+1} - u_i^n}{\Delta x} &= (q_1 - 1)u_i^{n+1} + (q_2 + s)v_i^n \\ &+ \mu_1 \left(\frac{-u_{i+2}^{n+1} + 16u_{i+1}^n - 30u_i^{n+1} + 16u_{i-1}^n - u_{i-2}^{n+1}}{12(\Delta x)^2} \right), \end{aligned} \tag{67}$$

$$\begin{aligned} \frac{v_i^{n+1} - v_i^n}{\Delta t} + c \frac{v_{i+1}^{n+1} - v_i^n}{\Delta x} &= -q_1u_i^{n+1} - q_2v_i^{n+1} \\ &+ \mu_1 \left(\frac{-v_{i+2}^{n+1} + 16v_{i+1}^n - 30v_i^{n+1} + 16v_{i-1}^n - v_{i-2}^{n+1}}{12(\Delta x)^2} \right), \end{aligned} \tag{68}$$

By using parameters $s = 5, c = 0, q_1 = q_2 = 0, \mu_1 = 1$ and $\mu_2 = 5$, Equations (67) and (70) are simplified to

$$\left(\frac{1}{\Delta t} + \frac{30}{12(\Delta x)^2} \right) v_i^{n+1} + \frac{5}{12(\Delta x)^2} (v_{i+2}^{n+1} + v_{i-1}^{n+1}) = \frac{1}{\Delta t} v_i^n + \frac{80}{12(\Delta x)^2} (v_{i+1}^n + v_{i-1}^n), \tag{69}$$

$$\begin{aligned} \frac{v_i^{n+1} - v_i^n}{\Delta t} + c \frac{v_{i+1}^{n+1} - v_i^n}{\Delta x} &= -q_1u_i^{n+1} - q_2v_i^{n+1} \\ &+ \mu_1 \left(\frac{-v_{i+2}^{n+1} + 16v_{i+1}^n - 30v_i^{n+1} + 16v_{i-1}^n - v_{i-2}^{n+1}}{12(\Delta x)^2} \right), \end{aligned} \tag{70}$$

The von Neumann stability analysis, $u_i^n = \zeta^n e^{j\Delta t i \Delta x}$, is used to check the stability region in Equations (67) and (70) to obtain the following

$$\begin{aligned} \left(\frac{1}{\Delta t} + 1 - q_1 + \frac{30\mu_1}{12(\Delta x)^2} \right) \zeta^{n+1} e^{j\Delta t i \Delta x} + \frac{1}{12(\Delta x)^2} \zeta^{n+1} e^{j\Delta t (i+1) \Delta x} \\ + \frac{1}{12(\Delta x)^2} \zeta^{n+1} e^{j\Delta t (i-1) \Delta x} = \frac{16}{12(\Delta x)^2} \zeta^n e^{j\Delta t (i-1) \Delta x} \\ + \frac{1}{\Delta t} \zeta^n e^{j\Delta t i \Delta x} + \frac{16}{12(\Delta x)^2} \zeta^n e^{j\Delta t (i+1) \Delta x}, \end{aligned} \tag{71}$$

$$\begin{aligned} \left(\frac{1}{\Delta t} + \frac{30}{12(\Delta x)^2} \right) \zeta^{n+1} e^{j\Delta t i \Delta x} + \frac{5}{12(\Delta x)^2} (\zeta^{n+1} e^{j\Delta t (i+1) \Delta x} + \zeta^{n+1} e^{j\Delta t (i-2) \Delta x}) \\ = \frac{1}{\Delta t} \zeta^n e^{j\Delta t i \Delta x} + \frac{80}{12(\Delta x)^2} (\zeta^n e^{j\Delta t (i+1) \Delta x} + \zeta^n e^{j\Delta t (i-1) \Delta x}), \end{aligned} \tag{72}$$

Equations (71) and (72) are simplified to obtain the following

$$\zeta = \frac{\frac{16}{6(\Delta x)^2} \cos(\omega) + \frac{1}{\Delta t}}{\frac{1}{\Delta t} + 1 + \frac{30}{12(\Delta x)^2} + \frac{1}{6(\Delta x)^2} \cos(\omega)}, \tag{73}$$

and

$$\zeta = \frac{\frac{1}{\Delta t} + \frac{80 \cos(\omega)}{6(\Delta x)^2}}{\frac{1}{\Delta t} + \frac{30}{(\Delta x)^2} + \frac{5}{6(\Delta x)^2} \cos(\omega)}, \tag{74}$$

where $\omega = \Delta t \Delta x$. By using the inequality $|\zeta| \leq 1$, when $\omega = \pi$, we obtain the following, respectively

$$\Delta t > \frac{24}{4 - 12(\Delta x)^2}, \tag{75}$$

and

$$\Delta t > \frac{6(\Delta x)^2}{35}, \tag{76}$$

4.3.6. Consistency with the System of ADR by Using HUPFD

By discretizing (56) and (57) by using the HUPFD1 scheme, we obtain the following

$$\begin{aligned} \frac{v_i^{n+1} - v_i^n}{\Delta t} + c \frac{v_{i+1}^{n+1} - v_i^n}{\Delta x} = & -q_1 u_i^{n+1} - q_2 v_i^{n+1} \\ & + \mu_1 \left(\frac{-v_{i+2}^{n+1} + 16v_{i+1}^n - 30v_i^{n+1} + 16v_{i-1}^n - v_{i-2}^{n+1}}{12(\Delta x)^2} \right), \end{aligned} \tag{77}$$

From (77), we obtain the following Taylor’s expansions

$$u_i^{n+1} \approx u_i^n + \Delta t \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} + \frac{(\Delta t)^3}{6} \frac{\partial^3 u}{\partial t^3} + \dots \tag{78}$$

$$u_i^{n-1} \approx u_i^n - \Delta t \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} - \frac{(\Delta t)^3}{6} \frac{\partial^3 u}{\partial t^3} + \dots \tag{79}$$

$$u_{i-1}^n \approx u_i^n - \Delta x \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u}{\partial x^2} - \frac{(\Delta x)^3}{6} \frac{\partial^3 u}{\partial x^3} + \dots \tag{80}$$

$$u_{i+1}^n \approx u_i^n + \Delta x \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u}{\partial x^2} + \frac{(\Delta x)^3}{6} \frac{\partial^3 u}{\partial x^3} + \dots \tag{81}$$

$$u_{i-2}^{n+1} \approx u_i^n + \Delta t \frac{\partial u}{\partial t} - 2\Delta x \frac{\partial u}{\partial x} + 2(\Delta x)^2 \frac{\partial^2 u}{\partial x^2} - 2(\Delta t \Delta x) \frac{\partial^2 u}{\partial t \partial x} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} + \dots \tag{82}$$

$$u_{i+2}^{n+1} \approx u_i^n + \Delta t \frac{\partial u}{\partial t} + 2\Delta x \frac{\partial u}{\partial x} + 2(\Delta x)^2 \frac{\partial^2 u}{\partial x^2} + 2(\Delta t \Delta x) \frac{\partial^2 u}{\partial t \partial x} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} + \dots \tag{83}$$

and

$$v_i^{n+1} \approx v_i^n + \Delta t \frac{\partial v}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 v}{\partial t^2} + \frac{(\Delta t)^3}{6} \frac{\partial^3 v}{\partial t^3} + \dots \tag{84}$$

$$v_i^{n-1} \approx v_i^n - \Delta t \frac{\partial v}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 v}{\partial t^2} - \frac{(\Delta t)^3}{6} \frac{\partial^3 v}{\partial t^3} + \dots \tag{85}$$

$$v_{i-1}^n \approx v_i^n - \Delta x \frac{\partial v}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 v}{\partial x^2} - \frac{(\Delta x)^3}{6} \frac{\partial^3 v}{\partial x^3} + \dots \tag{86}$$

$$v_{i+1}^n \approx v_i^n + \Delta x \frac{\partial v}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 v}{\partial x^2} + \frac{(\Delta x)^3}{6} \frac{\partial^3 v}{\partial x^3} + \dots \tag{87}$$

$$v_{i-2}^{n+1} \approx v_i^n + \Delta t \frac{\partial v}{\partial t} - 2\Delta x \frac{\partial v}{\partial x} + 2(\Delta x)^2 \frac{\partial^2 v}{\partial x^2} - 2(\Delta t \Delta x) \frac{\partial^2 v}{\partial t \partial x} + \frac{(\Delta t)^2}{2} \frac{\partial^2 v}{\partial t^2} + \dots \tag{88}$$

$$v_{i+2}^{n+1} \approx v_i^n + \Delta t \frac{\partial v}{\partial t} + 2\Delta x \frac{\partial v}{\partial x} + 2(\Delta x)^2 \frac{\partial^2 v}{\partial x^2} + 2(\Delta t \Delta x) \frac{\partial^2 v}{\partial t \partial x} + \frac{(\Delta t)^2}{2} \frac{\partial^2 v}{\partial t^2} + \dots \tag{89}$$

By applying Taylor’s expansions to (77), respectively, we obtain the following simplified equations

$$\begin{aligned} & (1 - q_1)u_i^n + \left(1 + \Delta t - q_1 \Delta t + \frac{32\mu_2 \Delta t}{12(\Delta x)^2} + \frac{c \Delta t}{12} \right) \frac{\partial u}{\partial t} \\ & + c \frac{\partial u}{\partial x} + \left(\frac{\Delta t}{2} + \frac{(\Delta t)^2}{2} - \frac{q_1 (\Delta t)^2}{2} + \frac{17\mu_1 (\Delta t)^2}{12(\Delta x)^2} + \frac{c(\Delta t)^2}{2\Delta x} \right) \frac{\partial^2 u}{\partial t^2} \\ & + \left(\frac{c(\Delta x)}{2} - \mu_1 \right) \frac{\partial^2 u}{\partial x^2} + \frac{c \Delta t}{2} \frac{\partial^2 u}{\partial t \partial x} - q_2 v_i^n = 0, \end{aligned} \tag{90}$$

and

$$\begin{aligned}
 q_2 v_i^n + \left(1 + q_2 \Delta t + \frac{32 \Delta t \mu_2}{12 (\Delta x)^2} + \frac{c (\Delta t)^2}{\Delta x} \right) \frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} \\
 + \left(\Delta t + q_2 (\Delta t)^2 + \frac{32 \mu_2 (\Delta t)^2}{12 (\Delta x)^2} + \frac{c (\Delta t)^2}{2 (\Delta x)} \right) \frac{\partial^2 v}{\partial t^2} + \left(\frac{c (\Delta x)}{2} \frac{4 \mu_2}{12} - \frac{12 \mu_2}{12} \right) \frac{\partial^2 v}{\partial x^2} \\
 + q_1 \left(u_i^n + \Delta t \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2} \dots \dots \right) = 0, \tag{91}
 \end{aligned}$$

Let $\Delta t = (\Delta x)^3$, which leads to Equations (90) and (91) being written as follows

$$\begin{aligned}
 (1 - q_1) u_i^n + \left(1 + (\Delta x)^3 - q_1 (\Delta x)^3 + \frac{32 \mu_1 \Delta x}{12} + c (\Delta x)^2 \right) \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} \\
 + \left(\frac{(\Delta x)^3}{2} + \frac{(\Delta x)^6}{2} - \frac{q_1 (\Delta x)^6}{2} + \frac{17 \mu_1 (\Delta x)^4}{12} + \frac{c (\Delta x)^5}{2} \right) \frac{\partial^2 u}{\partial t^2} + \left(\frac{c (\Delta x)}{2} - \mu_1 \right) \frac{\partial^2 u}{\partial x^2} \tag{92} \\
 + \frac{c (\Delta x)^3}{2} \frac{\partial^2 u}{\partial t \partial x} - q_2 v_i^n = 0,
 \end{aligned}$$

and

$$\begin{aligned}
 q_2 v_i^n + \left(1 + q_2 (\Delta x)^3 + \frac{32 \mu_2 \Delta x}{12} + c (\Delta x)^5 \right) \frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} \\
 + \left((\Delta x)^3 + q_2 (\Delta x)^6 + \frac{32 \mu_2 (\Delta x)^4}{12} + \frac{c (\Delta x)^5}{2} \right) \frac{\partial^2 v}{\partial t^2} \tag{93} \\
 + \left(\frac{c (\Delta x)}{2} - \mu_2 \right) \frac{\partial^2 v}{\partial x^2} + q_1 \left(u_i^n + (\Delta x)^3 \frac{\partial u}{\partial t} + \frac{(\Delta x)^6}{2} \frac{\partial^2 u}{\partial t^2} \right) = 0,
 \end{aligned}$$

When $\Delta x \rightarrow 0$, we obtain the original system ADR equations

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = (q_1 - 1)u + (q_2 + s)v + \mu_1 \frac{\partial^2 u}{\partial x^2}, \tag{94}$$

$$\frac{\partial v}{\partial t} + c \frac{\partial v}{\partial x} = -q_1 u - q_2 v + \mu_2 \frac{\partial^2 v}{\partial x^2}, \tag{95}$$

Theorem 1 (Solution positivity). *Let $u^0(x) \geq 0$, $0 \leq f^0(x) \leq 1$ and $v^0(x) \geq 0$ for the numerical scheme (11), (20) and (60); we have $u_i^{n+1} \geq 0$, $0 \leq f_i^{n+1}$ and $v_i^{n+1} \geq 0$, independent of the mesh step size [31].*

Proof. Let us assume that $u_i^{n+1} \geq 0$, $0 \leq f_i^{n+1} \leq 1$ and $v_i^n \geq 0$, for $i = 0, 1, \dots, N$, $n = 0, \dots, N$.

Equations (11), (20) and (60) comprise a coefficient matrix with a strictly diagonally dominant tridiagonal matrix, with positive values on the diagonal and sub-diagonals, which implies Equations (11), (20) and (60) yield non-negative inverses. Thus, the solution to the algebraic system is non-negative in the form of $u_i^{n+1} \geq 0$ and $v_i^{n+1} \geq 0$, for $i = 0, 1, \dots, N$.

Furthermore, it is observed that $f_i^{n+1} \geq 0$ in equations (6), (15), (54) and (55). By applying the same consideration at each time level, we complete the proof. □

5. Formulation of the POD Technique

In this section, the POD basis function is constructed and used to reduce the degree of freedom to the UPFD and HUPFD scheme.

The POD technique is used for the HUPFD by extracting the snapshot matrix in order to produce the EHUPFD, which leads to the construction of the approximate state variable ψ_i by developing a small subset of the leading basis function obtained from the SVD factorization. The POD technique seeks a low-dimensional basis function that can accurately approximate the state variable, which is a left singular vector from the SVD.

The first L sequence of solutions is extracted from the HUPFD solutions as given by u_i^n, v_i^n and s_i^n , which gives the formulated $n \times L$ snapshot matrix given below

$$A = \begin{pmatrix} u_1^1 & u_1^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_1^L \\ u_2^1 & u_2^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_2^L \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_m^1 & u_m^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_m^L \end{pmatrix}, \tag{96}$$

where m and L represent the size of the snapshot matrix, which is factorized by using the singular value decomposition (SVD) as given below

$$A = U\Sigma V^T = U \begin{pmatrix} \Sigma_{l \times l} & O_{l \times (L-l)} \\ O_{(m-l) \times l} & O_{(m-l) \times (L-l)} \end{pmatrix} V^T, \tag{97}$$

where $U = (\psi_1, \psi_2, \dots, \psi_m)$ is an $m \times m$ matrix representing an orthogonal matrix consisting of the eigenvectors of AA^T , $\Sigma_{l \times l} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_l)$ represents the diagonal matrix from the SVD, given in decreasing order $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_l > 0$, and $V = (\phi_1, \phi_2, \dots, \phi_L)$ is an $L \times L$ orthogonal matrix which represent the orthogonal eigenvector of $A^T A$ and $l = \text{rank}A$; O is a zero matrix [32].

From identical eigenvalues of $A^T A$ and AA^T , we obtain the eigenvalues of $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_l > 0 (l = \text{rank}A)$ for matrices AA^T and corresponding eigenvectors ψ_j , which is given by the formula below

$$\psi_j = A\phi_j / \sqrt{\lambda_j}, j = 1, 2, \dots, l, \tag{98}$$

This leads to the construction of the matrix A_M given below

$$A_M = U\Sigma V^T = U \begin{pmatrix} \Sigma_{M \times M} & O_{M \times (L-M)} \\ O_{(m-M) \times M} & O_{(m-M) \times (L-M)} \end{pmatrix} V^T, \tag{99}$$

where the diagonal matrix given by $\Sigma_{M \times M} = \text{diag}(\sigma_1, \sigma_1, \dots, \sigma_m)$ is constituted of the first M positive singular values of the diagonal matrix.

5.1. The Implementation of the Algorithm of the EHUPFD Scheme for the Advection Diffusion Reaction Equations

Below are the steps to implement the algorithm of the EHUPFD to solve the linear, non-linear and system ADR equations [32].

- Step 1: Snapshot matrix for HUPFD is extracted.
- Step 2: Formulation of the matrix $A = (u_i^n)_{m \times L}$ and calculation of eigenvalues and eigenvectors of matrix $A^T A$ and AA^T executed.
- Step 3: Choice of POD basis for the error tolerance of $\mu = O(\Delta t, \Delta x^2)$ is executed.
- Step 4: The EHUPFD computes the reduced-order solution.
- Step 5: Accuracy of the solution is checked for the renewal of the POD basis.

5.2. The EHUPFD Error Estimate

The following equation is used to estimate the error of the EHUPFD when solving the ADR equations.

$$\|u^n - u^{*n}\|_2 = \|u^n - u_M^n\|_2 = \|(A - \phi\phi^T A)\xi_n\|_2 \leq \sqrt{\lambda_{M+1}}, \tag{100}$$

$$n = 1, 2, \dots, L$$

Theorem 2 (Accuracy). The error estimate for the HUPFD u_i^n and u_i^{*n} from the EHUPFD is calculated by using the following equation

$$|u_i^n - u_i^{*n}| \leq (1 + \delta)^{n-L} \sqrt{\lambda_M + 1} \text{ for } (L + 1 \leq n \leq N), \tag{101}$$

$$\text{where } \delta = \frac{\Delta t}{\Delta x^2} \|B\|_{2,2}. \tag{102}$$

The absolute error, calculated between the actual and approximate solutions from the POD reduced-order model satisfy the following equation [32]

$$|u(x_i, t_n) - u_i^{*n}| = O\left((1 + \delta)^{n-L} \sqrt{\lambda_M + 1}, \Delta t, \Delta x^2\right), \quad 1 \leq i \leq M, \quad 1 \leq n \leq N, \tag{103}$$

6. Results and Discussion

In this section, we present solutions to the linear, non-linear and system ADR equations solved by using the EHUPFD, UPFD, NSFD and Crank–Nicolson methods. To check the performance of the methods, absolute errors, convergence rates and computational time are measured. The actual and numerical solutions are denoted by $u_{i,j}$ and $\bar{u}_{i,j}$, respectively. The error is represented by $e_{i,j}$ and its magnitude is calculated by using norm to infinity denoted by L_∞ -norm given by the formula below,

$$\|e_{i,n}\|_\infty = \|u_{i,n} - \bar{u}_{i,n}\|_\infty = \text{Max}|u_{i,n} - \bar{u}_{i,n}|, \tag{104}$$

The formula to calculate the convergence rate is given below,

$$q(\Delta t_k) = \frac{\log_2\left(\frac{e_k}{e_{k+1}}\right)}{\log_2\left(\frac{\Delta t}{\Delta t_{k+1}}\right)} \quad \text{or} \quad q(\Delta x_k) = \frac{\log_2\left(\frac{e_k}{e_{k+1}}\right)}{\log_2\left(\frac{\Delta x}{\Delta x_{k+1}}\right)}, \tag{105}$$

Finding the best snapshot size is paramount because it influences the EHUPFD solution accuracy. The best snapshot is determined by finding the solution to the problem when $L = 1$ and then an increment of 1 to L is evaluated until there is no significant change in the solution. The tolerance is calculated using the formula below

$$\|u_{exact} - u_L\|_\infty \leq \epsilon, \quad L = 1, 2, 3, \dots \tag{106}$$

where u_L is a solution for a specific value of L and u_{exact} is the exact solution. The first value of L that satisfies Equation (106) is the optimal value of L . We set $\epsilon = 10^{-15}$.

6.1. Results Obtained to Solve the Linear ADR Equation

The convergence rates of the UPFD, Crank–Nicolson, NSFD and HUPFD with respect to the time and spatial steps are shown in Table 1. The results show that the EHUPFD converges to the solution faster than the UPFD. Table 2 compares solutions obtained by the exact, UPFD, Crank–Nicolson, NSFD and EHUPFD solutions at $(t, x) = (1.33334, 0.0002), (2.6668, 0.0004)$ and $(4.0002, 0.0006)$. Table 3 compares the infinity norm results of the developed EHUPFD method against the Crank–Nicolson and NSFD, as well as their computational times. The results show that the EHUPFD takes less computational time to converge to the solution as compared to the HUPFD methods.

Table 3 compares the infinity norms of the proposed EHUPFD methods with the UPFD, Crank–Nicolson and NSFD methods in solving the linear ADR equation. The results show that EHUPFD methods preserve solution accuracy.

Figure 1 shows the contribution made by each POD basis mode. The first five modes contain the predominant information where the EHUPFD meets the theoretical accuracy requirement. We observed that five modes are significant in optimizing the linear ADR

equation. Figure 2b compares the number of POD modes against the absolute error. We observe that only five POD modes are required in order to converge to the toleration of 10^{-14} .

Table 1. Convergence rate results obtained from the UPFD, NSFD, Crank–Nicolson and HUPFD methods when solving the ADR equation.

| Δt_k | Δx_k | $q(\Delta t_k)$ | | | | |
|--------------|--------------|-----------------|--------|--------|----------------|--------|
| | | UPFD | HUPFD | EHUPFD | Crank–Nicolson | NSFD |
| 0.00005 | 0.6667 | 2.1675 | 2.5233 | 2.5233 | 4.5606 | 2.0579 |
| 0.0001 | 0.6667 | 0.9973 | 1.0054 | 1.0054 | 1.1996 | 1.9427 |
| 0.0002 | 0.6667 | 1.8201 | 1.9990 | 1.9990 | 2.7372 | 1.8871 |
| 0.0004 | 0.6667 | | | | | |

Table 2. Numerical results obtained when solving the linear ADR equation by using the UPFD, HUPFD, EHUPFD, NSFD and Crank–Nicolson methods.

| (x, t) | Exact Solution | UPFD Solution | Absolute Error | EHUPFD | Absolute Error | Crank–Nicolson | Absolute Error | NSFD | Absolute Error |
|------------------|----------------|---------------|-------------------------|--------|-------------------------|----------------|-------------------------|--------|-------------------------|
| (1.3334, 0.0002) | 0.5136 | 0.5137 | 9.3425×10^{-5} | 0.5125 | 0.0011 | 0.5138 | 1.6145×10^{-4} | 0.5137 | 9.4142×10^{-5} |
| (2.6668, 0.0004) | 0.1355 | 0.1356 | 8.1664×10^{-5} | 0.1366 | 0.0011 | 0.1356 | 6.7682×10^{-5} | 0.1356 | 7.4518×10^{-5} |
| (4.0002, 0.0006) | 0.0357 | 0.0358 | 5.1090×10^{-4} | 0.0363 | 3.2521×10^{-5} | 0.0358 | 2.9601×10^{-5} | 0.0358 | 3.2769×10^{-5} |

Table 3. Infinity norm results obtained by using the exact, UPFD, HUPFD, EHUPFD, Crank–Nicolson and NSFD methods to solve the linear ADR equation.

| (x, t) | Exact | UPFD | Error | HUPFD | Error | EHUPFD | Error | Crank–Nicolson | Error | NSFD | Error |
|-------------------------------|--------|----------|--------|----------|--------|----------|--------|----------------|--------|----------|--------|
| $\ u(\cdot, 0.0002)\ _\infty$ | 1.1654 | 1.1654 | 0.0025 | 1.1654 | 0.0025 | 1.1655 | 0.0025 | 1.1654 | 0.0025 | 1.1654 | 0.0025 |
| $\ u(\cdot, 0.0004)\ _\infty$ | 1.1657 | 1.1657 | 0.0025 | 1.1657 | 0.0025 | 1.1657 | 0.0025 | 1.1657 | 0.0025 | 1.1657 | 0.0025 |
| $\ u(\cdot, 0.0006)\ _\infty$ | 1.1659 | 1.1660 | 0.0025 | 1.1660 | 0.0025 | 1.1660 | 0.0025 | 1.1660 | 0.0025 | 1.1657 | 0.0025 |
| Time | | 0.001230 | | 0.002748 | | 0.002717 | | 0.003089 | | 0.002689 | |

Figure 3a to Figure 4a show the surface plots of the exact, UPFD and EUPFD solutions of the linear ADR. Figure 4b shows the increase in error when the POD modes increases. The figures show that the developed methods achieve positivity of the solutions and the results are comparable with well-known methods such as Crank–Nicolson and NSFD.

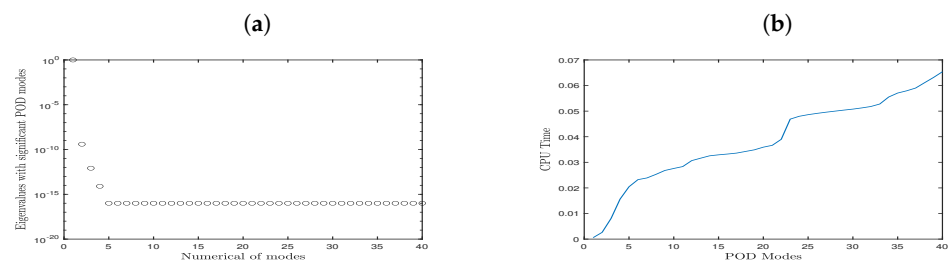


Figure 1. (a) Plot for the number of the POD modes against the eigenvalues with significant POD modes. (b) Plot for the number of POD modes against computational time when solving the linear ADR equation.

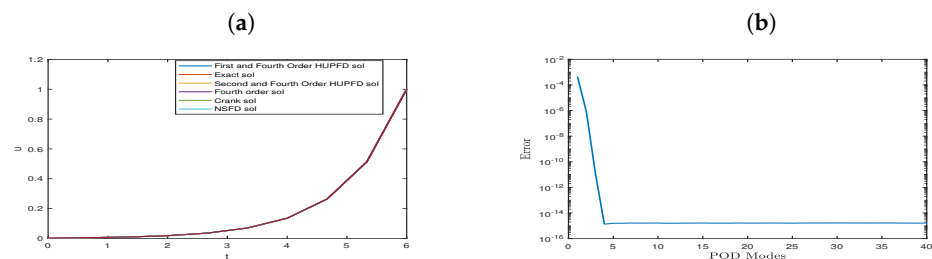


Figure 2. (a) Plots for the UPFD, EHUPFD, Crank–Nicolson and NSFD solutions when solving the linear ADR equation. (b) shows the plot for the number of POD modes against the absolute error.

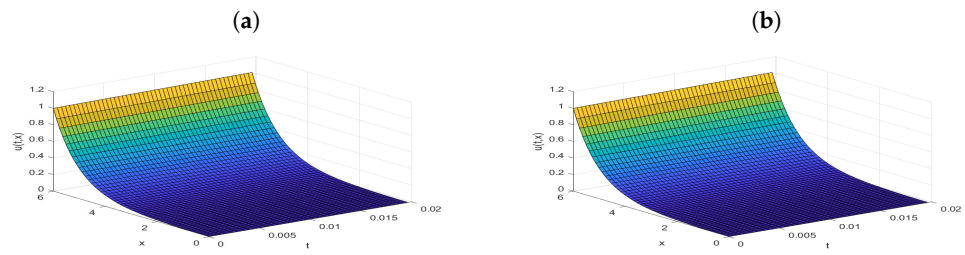


Figure 3. (a) Surface plot of the UPFD solution when solving the linear ADR equation. (b) shows the surface plot for the EHUPFD solution when solving the linear ADR equation.

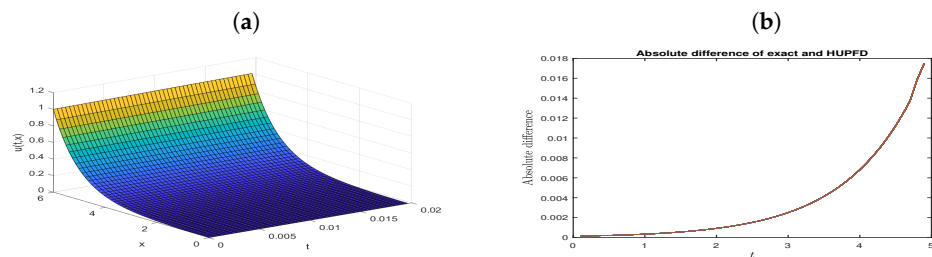


Figure 4. (a) Surface plot of the exact solution when solving the linear ADR equation. (b) Plot for the time against the absolute difference.

6.2. Results Obtained by Using the Non-Linear ADR Equation

The convergence rates of the UPFD, Crank–Nicolson, NSFD and the EHUPFD with respect to the time and spatial steps are shown in Table 4. Table 5 compares the exact, UPFD and EHUPFD solutions at $(t, x) = (0.03338, 0.002)$, $(0.0676, 0.004)$ and $(0.1014, 0.006)$. The HUPFD has a slightly higher convergence rate than UPFD, while the EHUPFD has significantly reduced the computational time taken to converge to the solution of the ADR equation.

Table 4. Convergence rate results obtained when solving the non-linear ADR reaction equation by the UPFD, HUPFD, Crank–Nicolson, NSFD and EHUPFD methods.

| | | $q(\Delta t_k)$ | | | | |
|--------------|--------------|-----------------|--------|--------|----------------|---------|
| Δt_k | Δx_k | UPFD | HUPFD | EHUPFD | Crank–Nicolson | NSFD |
| 0.0005 | 0.6667 | 1.7166 | 1.7197 | 1.7197 | 1.7197 | 1.71652 |
| 0.001 | 0.6667 | 1.7160 | 1.7150 | 1.7151 | 1.7150 | 1.71594 |
| 0.002 | 0.6667 | 1.7168 | 1.7195 | 1.7195 | 1.7195 | 1.71671 |
| 0.004 | 0.6667 | | | | | |

Table 5. Solutions obtained by the exact, UPFD, HUPFD and EHUPFD methods when solving the non-linear ADR equation.

| (x, t) | Exact Solution | UPFD Solution | Absolute Error | HUPFD Solution | Absolute Error | EHUPFD Solution | Absolute Error |
|--------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| $(0.03338, 0.002)$ | 6.5475×10^{-9} | 5.4779×10^{-9} | 1.0697×10^{-9} | 5.4763×10^{-9} | 1.0712×10^{-9} | 5.4908×10^{-9} | 1.0567×10^{-9} |
| $(0.0676, 0.004)$ | 8.1536×10^{-8} | 6.8787×10^{-8} | 1.2749×10^{-8} | 6.8751×10^{-8} | 1.2784×10^{-8} | 6.9355×10^{-8} | 1.2181×10^{-8} |
| $(0.1014, 0.006)$ | 8.4478×10^{-7} | 7.1808×10^{-7} | 1.2670×10^{-7} | 7.1759×10^{-7} | 1.2718×10^{-7} | 7.2659×10^{-7} | 1.1819×10^{-7} |

In Table 6, we compare the infinity norms of the proposed HUPFD with the UPFD, Crank–Nicolson and NSFD methods in solving the linear ADR equation. For this example, the results show that the EHUPFD method preserves the accuracy and reduces the computational time taken to converge to the solution of the ADR equation.

Table 6. Infinity norm results obtained when solving the non-linear ADR equation by using the exact, UPFD, NSFD and HUPFD methods.

| (x, t) | Exact | UPFD | Error | HUPFD | Error | EHUPFD | Error | Crank–Nicolson | Error | NSFD | Error |
|------------------------------|--------|----------|--------|----------|--------|----------|--------|----------------|--------|----------|--------|
| $\ u(\cdot, 0.002)\ _\infty$ | 2.0590 | 2.0127 | 0.0512 | 2.0127 | 0.0512 | 2.0126 | 0.0513 | 2.0127 | 0.0512 | 2.0127 | 0.0512 |
| $\ u(\cdot, 0.004)\ _\infty$ | 2.0591 | 2.0127 | 0.0512 | 2.0126 | 0.0514 | 2.0123 | 0.0514 | 2.0126 | 0.0514 | 2.0127 | 0.0512 |
| $\ u(\cdot, 0.004)\ _\infty$ | 2.0592 | 2.0128 | 0.0512 | 2.0126 | 0.0515 | 2.0121 | 0.0515 | 2.0126 | 0.0515 | 2.0128 | 0.0512 |
| Time | | 0.001227 | | 0.004136 | | 0.000492 | | 0.004205 | | 0.003954 | |

Figure 5a shows the contribution made by each POD basis mode. The first five modes contain the predominant information where the EHUPFD meets the theoretical accuracy requirement. We observed five modes are significant in optimizing the linear ADR equation. Figure 5b shows the increase in computational time when the number of modes increases.

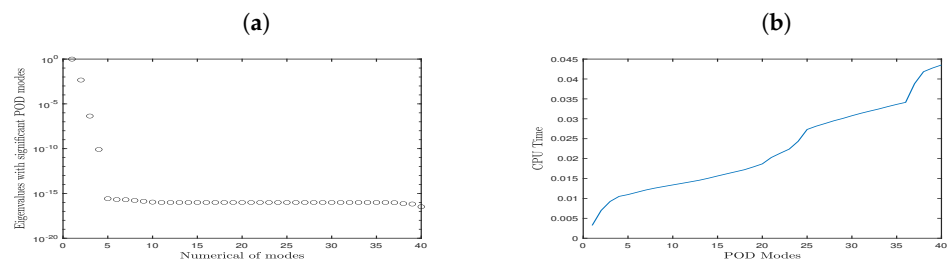


Figure 5. (a) Plot of the number of POD modes against the eigenvalues with significant POD modes. (b) Number of the POD modes against the computational time.

Figure 6a compares the results obtained by the EHUPFD, Crank–Nicolson and NSFD methods. The results show that the EHUPFD solution is comparable with the Crank–Nicolson and NSFD solutions.

Figure 6b compares the error against the number of modes. The results show that the error of the solution decreases when the number of modes increases.

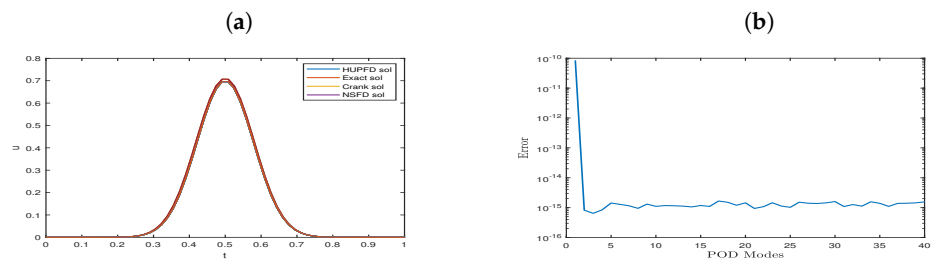


Figure 6. (a) Plot for the solutions obtained by the exact, HUPFD, Crank–Nicolson and NSFD methods when solving the non-linear ADR equation. (b) Plot of the number of POD modes against the absolute error.

Figure 7a to Figure 8a show the surface plots of the exact, UPFD and EUPFD solutions of the non-linear ADR equation. Figure 8b show the absolute errors between the exact solution and the EHUPFD solution for the non-linear ADR against time.

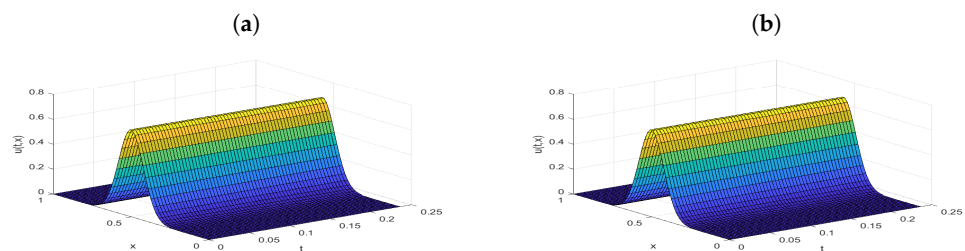


Figure 7. (a) Surface plot of the exact solution when solving the the non-linear ADR equation. (b) Surface plot of the UPFD solutions when solving the non-linear ADR equation.

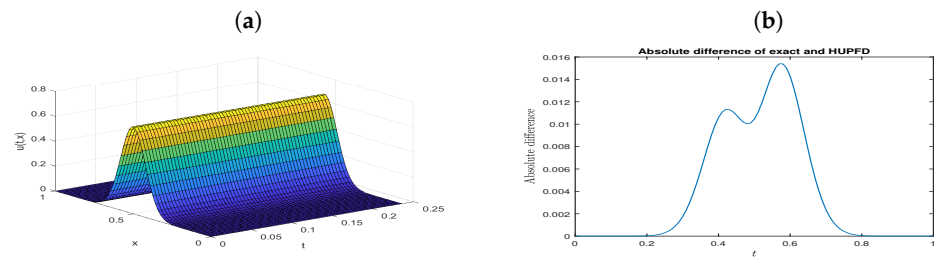


Figure 8. (a) Surface plot of the EHUPFD solution when solving the non-linear ADR equation. (b) Plot of the time against the absolute error.

6.3. Results Obtained When Solving the Schnakenberg Model

The convergence rates of the UPFD, HUPFD and EHUPFD with respect to the time and spatial steps are shown in Table 7. The results show that the convergence rate of the EHUPFD methods is slightly higher than the UPFD when solving the Schnakenberg model. Tables 8 and 9 compare the solutions of the UPFD, HUPFD and EHUPFD solutions at $(t, x) = (1.33334, 0.0002), (2.6668, 0.0004)$ and $(4.0002, 0.0006)$, as well as the computational times. The results show that the EHUPFD takes less computational time to converge to the solution than the UPFD, HUPFD and EHUPFD methods. Figure 9a to Figure 10b show the surface plots of the UPFD, HUPFD and EHUPFD of the Schnakenberg model. The figures show that the developed methods achieve positivity of the solutions. Figure 11 compares the solutions obtained by the exact HUPFD, EHUPFD and UPFD. The results show that the proposed methods are comparable to the known methods of the UPFD.

Table 7. Convergence rate when solving the Schnakenberg model by using the UPFD, HUPFD and EHUPFD methods.

| | | $q(\Delta t_k)$ | | |
|--------------|--------------|-----------------|--------|--------|
| Δt_k | Δx_k | UPFD | HUPFD | EHUPFD |
| 0.00005 | 0.6667 | 1.0002 | 1.0029 | 1.0027 |
| 0.0001 | 0.6667 | 1.0004 | 1.0058 | 1.0055 |
| 0.0002 | 0.6667 | 1.0007 | 1.0116 | 1.0109 |
| 0.0004 | 0.6667 | | | |

Table 8. Solutions u obtained by the UPFD, HUPFD and EHUPFD methods when solving the Schnakenberg system equation.

| (x, t) | UPFD | HUPFD | EHUPFD |
|----------------|----------|----------|----------|
| $u(:, 0.0002)$ | 0.0048 | 0.0050 | 0.0050 |
| $u(:, 0.0004)$ | 0.0182 | 0.0198 | 0.0198 |
| $u(:, 0.0006)$ | 0.0690 | 0.0788 | 0.0788 |
| Time | 0.005959 | 0.024321 | 0.004894 |

Table 9. Solutions v obtained by the UPFD, HUPFD and EHUPFD methods when solving the Schnakenberg system equation.

| (x, t) | UPFD | HUPFD | EHUPFD |
|----------------|----------|----------|----------|
| $u(:, 0.0002)$ | 0.0048 | 0.5310 | 0.5310 |
| $u(:, 0.0004)$ | 0.0186 | 0.1429 | 0.1429 |
| $u(:, 0.0006)$ | 0.0711 | 0.0390 | 0.0390 |
| Time | 0.005312 | 0.021700 | 0.004220 |

The convergence rates of the UPFD, HUPFD and EHUPFD with respect to the time and spatial steps are shown in Table 7. The results show that the convergence rate of

the EHUPFD methods is slightly higher than the UPFD when solving the Schnakenberg model. Tables 8 and 9 compare the UPFD, HUPFD and EHUPFD solutions at $(x, t) = (1.33334, 0.0002), (2.6668, 0.0004)$ and $(4.0002, 0.0006)$, as well as the computational times. The results show that the EHUPFD takes less computational time to converge to the solution than the UPFD, HUPFD and EHUPFD methods. Figure 9a to Figure 10b show the surface plots of the UPFD, HUPFD and EHUPFD of the Schnakenberg model. The figures show that the developed methods achieve positivity of the solutions. Figure 11 compares the solutions obtained by the exact HUPFD, EHUPFD and UPFD. The results show that the proposed methods are comparable to the know methods of the UPFD.

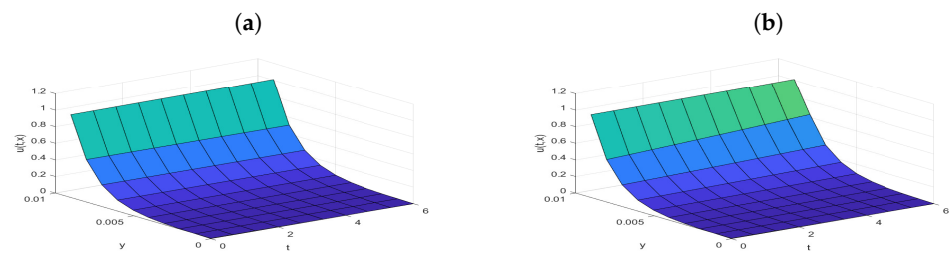


Figure 9. (a) Exact solutions obtained when solving the Schnakenberg system equations. (b) UPFD solutions obtained when solving the Schnakenberg system equations.

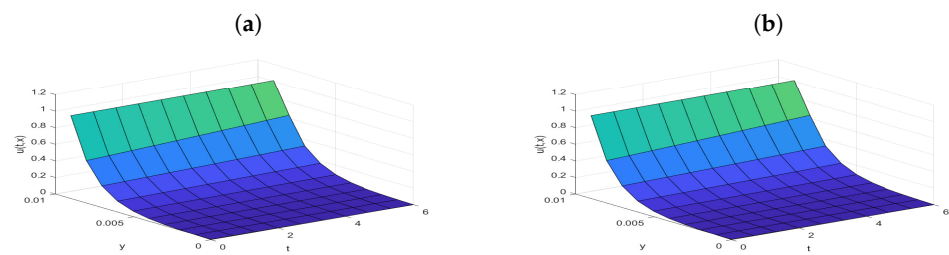


Figure 10. (a) HUPFD solutions obtained when solving the Schnakenberg system equations. (b) EHUPFD solutions when solving the Schnakenberg system equations.

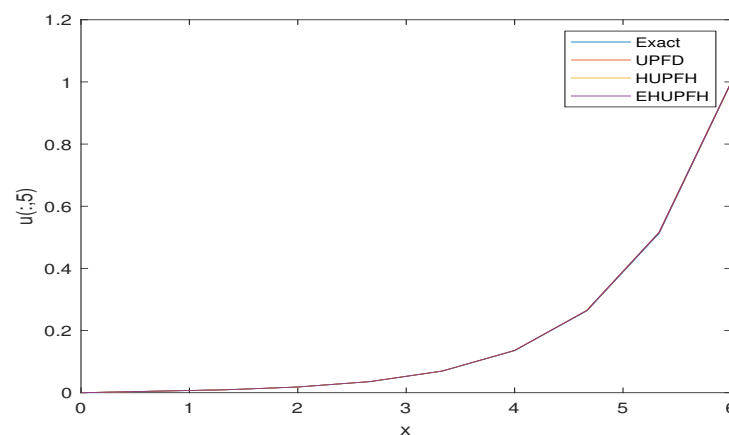


Figure 11. (Comparison of plots for the solutions obtained by the UPFD, HUPFD and EHUPFD methods when solving the Schnakenberg system equations.

7. Conclusions

This paper uses the EHUPFD to find solutions to the linear, non-linear and system advection diffusion reaction equations. The goal was to investigate the performance of the EHUPFD in terms of computational time, absolute error and convergence rate. The results show that the EHUPFD gives solutions that are a good approximation to solve the linear, non-linear and system ADR equations, which validates the method’s efficiency because

it confirms the theoretical results. When solving the ADR equations by EHUPFD, a few modes containing the predominant information are needed to satisfy the theoretical results. The theoretical results are achieved using only 4 POD bases from the HUPFD's 40 classical solutions as a snapshot. This implies that the EHUPFD saves computational time, reduces the degree of freedom and alleviates truncation error. The EHUPFD method also preserves the positivity of the solutions.

Author Contributions: Formal analysis, N.N.; Investigation, N.N.; Supervision, P.D. and B.A.J.; Writing—original draft, N.N.; Writing—review and editing, P.D. and B.A.J. All authors have read and agreed to the published version of the manuscript.

Funding: The authors wish to acknowledge the financial support from the University of Venda.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data was created in this paper.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Abbreviations

| | |
|----------|--|
| EHUPFD | Enhanced higher-order unconditionally positive finite difference methods |
| ADR | Advection diffusion reaction |
| POD | Proper orthogonal decomposition |
| CPU time | Computational time |

Nomenclature

| | |
|------------|---|
| x | Spatial variable |
| t | Time variable |
| Δx | Change in spatial variable |
| Δt | Change in time variable |
| u_t | Rate at which concentration of substances changes over time |
| u_x | Advection term |
| u_{xx} | Diffusion term |
| U | Advection coefficient |
| D | Diffusion coefficient |

References

- Liang, Y.; Lee, H.; Lim, S.; Lin, W.; Lee, K.; Wu, C. Proper orthogonal decomposition and its applications—Part I: Theory. *J. Sound Vib.* **2002**, *252*, 527–544. [[CrossRef](#)]
- Luo, Z.; Teng, F.; Xia, H. A reduced-order extrapolated Crank–Nicolson finite spectral element method based on POD for the 2D non-stationary Boussinesq equations. *J. Math. Anal. Appl.* **2019**, *471*, 564–583. [[CrossRef](#)]
- Luo, Z.; Li, H.; Sun, P. A reduced-order Crank–Nicolson finite volume element formulation based on POD method for parabolic equations. *Appl. Math. Comput.* **2013**, *219*, 5887–5900. [[CrossRef](#)]
- Zhou, Y.; Luo, Z. An optimized Crank–Nicolson finite difference extrapolating model for the fractional-order parabolic-type sine-Gordon equation. *Adv. Differ. Equ.* **2019**, *2019*, 1. [[CrossRef](#)]
- Britton, N. *Others Reaction-Diffusion Equations and Their Applications to Biology*; Academic Press: New York, NY, USA, 1986.
- Hao, X.; Liu, L.; Wu, Y. Iterative solution for nonlinear impulsive advection-reaction-diffusion equations. *J. Nonlinear Sci. Appl.* **2016**, *9*, 4070–4077. [[CrossRef](#)]
- Ndou, N.; Dlamini, P.; Jacobs, B. Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations. *Mathematics* **2022**, *10*, 2639. [[CrossRef](#)]
- Liu, B.; Allen, M.; Kojouharov, H.; Chen, B. Others Finite-element solution of reaction-diffusion equations with advection. *Comput. Methods Water Resour.* **1996**, *1*, 3–12.
- Khan, H.; Gómez-Aguilar, J.; Khan, A.; Khan, T. Stability analysis for fractional order advection–reaction diffusion system. *Phys. Stat. Mech. Its Appl.* **2019**, *521*, 737–751. [[CrossRef](#)]

10. Chapwanya, M.; Lubuma, J.; Mickens, R. Nonstandard finite difference schemes for Michaelis–Menten type reaction-diffusion equations. *Numer. Methods Partial. Differ. Equ.* **2013**, *29*, 337–360. [[CrossRef](#)]
11. Appadu, A. Analysis of the unconditionally positive finite difference scheme for advection-diffusion-reaction equations with different regimes. *Aip Conf. Proc.* **2016**, *1738*, 030005.
12. Appadu, A. Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *Int. J. Numer. Methods Heat Fluid Flow* **2017**, *27*, 1412–1429. [[CrossRef](#)]
13. Vasilyev, O. High order finite difference schemes on non-uniform meshes with good conservation properties. *J. Comput. Phys.* **2000**, *157*, 746–761. [[CrossRef](#)]
14. Schwendt, M.; Pötz, W. Transparent boundary conditions for higher-order finite-difference schemes of the Schrödinger equation in (1+ 1) D. *Comput. Phys. Commun.* **2020**, *250*, 107048. [[CrossRef](#)]
15. Carpenter, M.; Gottlieb, D.; Abarbanel, S. The stability of numerical boundary treatments for compact high-order finite-difference schemes. *J. Comput. Phys.* **1993**, *108*, 272–295. [[CrossRef](#)]
16. Georgakopoulos, S.; Birtcher, C.; Balanis, C.; Renaut, R. Higher-order finite-difference schemes for electromagnetic radiation, scattering, and penetration. 1. Theory. *IEEE Antennas Propag. Mag.* **2002**, *44*, 134–142. [[CrossRef](#)]
17. Bilbao, S.; Hamilton, B. Higher-order accurate two-step finite difference schemes for the many-dimensional wave equation. *J. Comput. Phys.* **2018**, *367*, 134–165. [[CrossRef](#)]
18. Chung, Y.; Tucker, P. Accuracy of higher-order finite difference schemes on nonuniform grids. *AIAA J.* **2003**, *41*, 1609–1611. [[CrossRef](#)]
19. Morinishi, Y.; Lund, T.; Vasilyev, O.; Moin, P. Fully conservative higher order finite difference schemes for incompressible flow. *J. Comput. Phys.* **1998**, *143*, 90–124. [[CrossRef](#)]
20. Visbal, M.; Gaitonde, D. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *J. Comput. Phys.* **2002**, *181*, 155–185. [[CrossRef](#)]
21. Duressa, G.; Bullo, T.; Kiltu, G. Fourth order compact finite difference method for solving one dimensional wave equation. *Int. J. Eng. Appl. Sci.* **2016**, *8*, 30–39. [[CrossRef](#)]
22. Cui, M. Compact finite difference method for the fractional diffusion equation. *J. Comput. Phys.* **2009**, *228*, 7792–7804. [[CrossRef](#)]
23. Tian, Z.; Ge, Y. A fourth-order compact ADI method for solving two-dimensional unsteady convection–diffusion problems. *J. Comput. Appl. Math.* **2007**, *198*, 268–286. [[CrossRef](#)]
24. Li, Y.; Kim, J. An efficient and stable compact fourth-order finite difference scheme for the phase field crystal equation. *Comput. Methods Appl. Mech. Eng.* **2017**, *319*, 194–216. [[CrossRef](#)]
25. Li, L.; Jiang, Z.; Yin, Z. Fourth-order compact finite difference method for solving two-dimensional convection–diffusion equation. *Adv. Differ. Equ.* **2018**, *2018*, 234. [[CrossRef](#)]
26. Duressa, G.; Bullo, T. Higher-Order Numerical Method for Singularly Perturbed Delay Reaction-Diffusion Problems. *Pure Appl. Math.* **2021**, *10*, 68–76. [[CrossRef](#)]
27. Mohanty, R.; Gopal, V. A fourth-order finite difference method based on spline in tension approximation for the solution of one-space dimensional second-order quasi-linear hyperbolic equations. *Adv. Differ. Equ.* **2013**, *2013*, 70. [[CrossRef](#)]
28. Sari, M.; Gürarlan, G.; Zeytinoglu, A. High-order finite difference schemes for solving the advection-diffusion equation. *Math. Comput. Appl.* **2010**, *15*, 449–460. [[CrossRef](#)]
29. Gurarlan, G.; Karahan, H.; Alkaya, D.; Sari, M.; Yasar, M. Numerical solution of advection-diffusion equation using a sixth-order compact finite difference method. *Math. Probl. Eng.* **2013**, *2013*, 672936. [[CrossRef](#)]
30. Chen-Charpentier, B.; Kojouharov, H. An unconditionally positivity preserving scheme for advection-diffusion reaction equations. *Math. Comput. Model.* **2013**, *57*, 2177–2185. [[CrossRef](#)]
31. Kolev, M.; Koleva, M.; Vulkov, L. An Unconditional Positivity-Preserving Difference Scheme for Models of Cancer Migration and Invasion. *Mathematics* **2022**, *10*, 131. [[CrossRef](#)]
32. Luo, Z.; Chen, G. *Proper Orthogonal Decomposition Methods for Partial Differential Equations*; Academic Press: New York, NY, USA, 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.