mathematics

Article

Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations

Ndivhuwo Ndou *, Phumlani Dlamini and Byron Alexander Jacobs

Department of Mathematics and Applied Mathematics, University of Johannesburg, Auckland Park,
Johannesburg 2006, South Africa; pgdlamini@uj.ac.za (P.D.); byronj@uj.ac.za (B.A.J.)
* Correspondence: 220164883@student.uj.ac.za

Abstract: In this study, we develop the enhanced unconditionally positive finite difference method (EUPFD), and use it to solve linear and nonlinear advection–diffusion–reaction (ADR) equations. This method incorporates the proper orthogonal decomposition technique to the unconditionally positive finite difference method (UPFD) to reduce the degree of freedom of the ADR equations. We investigate the efficiency and effectiveness of the proposed method by checking the error, convergence rate, and computational time that the method takes to converge to the exact solution. Solutions obtained by the EUPFD were compared with the exact solutions for validation purposes. The agreement between the solutions means the proposed method effectively solved the ADR equations. The numerical results show that the proposed method greatly improves computational efficiency without a significant loss in accuracy for solving linear and nonlinear ADR equations.

Keywords: proper orthogonal decomposition; unconditionally positive finite difference method; advection–diffusion–reaction equations; enhanced unconditionally positive finite difference method

MSC: 65M06; 65M12; 65N06

Citation: Ndou, N.; Dlamini, P.; Jacobs, B.A. Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations. *Mathematics* **2022**, *10*, 2639. https://doi.org/10.3390/math10152639

Academic Editor: Peng-Nan Sun and Tao Jiang

Received: 26 May 2022
Accepted: 11 July 2022
Published: 27 July 2022

1. Introduction

Many problems in science and engineering are modeled by partial differential equations that are often difficult to solve for an exact solution and, in some cases, impossible. Numerical methods are a good alternative in providing approximate solutions close to the exact solution. As a result, several accurate numerical methods, such as the Crank–Nicolson [1], finite element [2], finite volume [3], spectral [4], and nonstandard finite difference [5,6] methods were developed. Developing better numerical methods continues to stimulate much interest among researchers. Motivated by the continued need for better-performing numerical methods, in this work, we propose a novel hybrid method that is based on combining elements of the unconditionally positive finite difference method (UPFD) and the proper orthogonal decomposition method (POD), called the enhanced unconditionally positive finite difference method (EUPFD).

The UPFD was developed by Chen-Charpentier and Kojouharov [7] for the advection–diffusion–reaction equation (ADR), and has since been used by several other researchers [8–10]. It is a finite difference-based method that guarantees the positivity of solutions independent of time step and mesh size. Numerical schemes that preserve the positivity of solutions are important in physical applications. To have significant meaning, quantities such as chemical species concentration, population sizes, and neutron numbers require positive solutions. On the other hand, the POD is a powerful technique widely used in statistics and image processing [11,12] to reduce a large number of interdependent variables to a much smaller number of uncorrelated variables while retaining as much possible of the variation in the original variables. By utilizing the POD, the EUPFD involves the extraction of a set of basis functions from the UPFD solution called the snapshot matrix; then, it uses a small subset of leading basis functions to construct state variable approximations. One of the most persistent challenges faced by several numerical methods when solving PDEs is that they require many grid points and numerous iterations to converge to the exact solution. That significantly affects their efficiency in terms of computational speed and memory requirement. The POD has been used to reduce the dimensions of numerical methods, such as the Crank–Nicolson method [13–18]. The POD is useful when it is impossible to perform numerical simulations due to large-scale computing requirements.

The POD modes are updated until the EUPFD satisfies the theoretical accuracy requirements. Luo and Sun [19] combined the POD with the finite volume, the finite element, and the finite difference methods to solve Navier–Stokes, Sobolev, Burger's, and hyperbolic equations. In this work, we extend the POD coupling by hybridising it with the UPFD to obtain a novel numerical method that exhibits higher accuracy with less computing time, fewer degrees of freedom in numerical computations, and reduced truncation error, ensuring the positivity of the solutions.

We test the applicability of the EUPFD on the nonlinear ADR equations given below:

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

where U_a is the advection coefficient, D is the diffusion coefficient, $u_t(t,x)$ is the rate at which the concentration of substances changes over time, $u_x(t,x)$ is the advection term, $u_{xx}(t,x)$ is the diffusion term, and $f(t,x,u)$ is the reaction term or the source [20]. The ADR equation governs the process of advection and diffusion simultaneously. It is used to model exponential travelling waves, absorption of pollutants in soil, semiconductors, modelling of biological systems, and diffusion of neutrons [21–24]. Among other numerical methods, the UPFD method has been used to solve ADR equations, is stable, accurate, and consistent in rectangular domains, and it guarantees the positivity of the solution that is independent of the step sizes [25]. The Rosenbrock scheme is another interesting approach that preserves solution accuracy and is mostly used to solve stiff ODEs problems. Al'shin, Al'shina, Kalitkin, and Koryagina [26] proposed the complex Rosenbrock schemes for the numerical integration of differential algebraic systems by the ε-embedding method. The method converges quadratically, and the scheme applies even to superstiff systems. The method enables computations to be performed with guaranteed accuracy. However, as always, numerical methods need to be refined and improved to converge faster to the exact solution.

The rest of the paper is organised as follows. In Section 2, we present the UPFD method, and illustrate how it is used to solve linear and nonlinear ADR equations. We also provide the stability and consistency analysis of the UPFD scheme. In Section 3, we formulate the POD basis to optimise the UPFD scheme and this results in the EUPFD method. In Section 4, numerical results are presented to illustrate the efficiency and effectiveness of the EUPFD method to solve the discussed problems. Section 5 provides the conclusion of the study.

2. Unconditionally Positive Finite Difference Method

In this section, we present details of the unconditionally positive finite difference method (UPFD), and illustrate how it is utilized to solve linear and nonlinear ADR equations.

The discretization of the linear and nonlinear ADR equation is obtained by using the derivative schemes given below to obtain the UPFD schemes [7],

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

$$\frac{\partial u}{\partial x} \approx \frac{u_i^{n+1} - u_{i-1}^{n}}{\Delta x} \tag{3}$$

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}, \tag{4}$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^n - 2u_i^{n+1} - u_{i-1}^n}{(\Delta x)^2}. \tag{5}$$

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

When approximating the first and second space derivatives, the terms in the finite difference schemes are evaluated at different time levels n and $n+1$. This is required to preserve the positivity of the solution. Equation (2) is used when the coefficient of the first space derivative is negative, while Equation (3) is used when the derivative is positive. On the other hand, Equation (5) is used whether the coefficient of the second space derivative is positive or negative.

2.1. Linear Advection–Diffusion–Reaction Equation

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

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

subject to the following initial and boundary conditions:

$$\begin{cases} u(x,0) = e^{-x} & 0 \leq x \leq 6, \\ u(0,t) = e^{t} & 0 \leq t \leq T, \\ u_x(6,t) = -u(6,t), & \end{cases} \tag{7}$$

with the exact solution given by $u(x,t) = e^{(t-x)}$ [7].

Since the coefficient of $\dfrac{\partial u}{\partial x}$ in Equation (6) is positive, we used Equation (3) to find the UPFD scheme. Therefore, the UPFD discretization of Equation (6) is given by

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{u_i^{n+1} - u_{i-1}^n}{\Delta x} = \frac{u_{i+1}^n - 2u_i^{n+1} + u_{i-1}^n}{(\Delta x)^2} - u_i^{n+1}. \tag{8}$$

Equation (8) can be written as

$$u_i^{n+1} = \frac{u_i^n + (\lambda + \beta)u_{i-1}^n + \beta u_{i-1}^n}{1 + \Delta t + \lambda + 2\beta}. \tag{9}$$

where $\lambda = \dfrac{\Delta t}{\Delta x}$ and $\beta = \dfrac{\Delta t}{(\Delta x)^2}$.

2.1.1. Stability

To obtain the region of stability of the UPFD scheme, we used the Fourier series analysis of the form $u_i^n = \xi^n e^{j\Delta t i \Delta x}$ on Equation (9). The amplification factor is then given by

$$\xi = \frac{1+(\lambda+2\beta)\cos\omega}{1+\Delta t+\lambda+2\beta} - I\left[\frac{\lambda\sin\omega}{1+\Delta t+\lambda+2\beta}\right] \tag{10}$$

where ω is the phase angle, $\omega = \theta\Delta x$ where θ is the wave number, and Δt is the spatial step size, and $\xi = \dfrac{\xi^{n+1}}{\xi^n}$.

By using the $\mid \xi \mid^2 \leq 1$ in (10), where $\mid \xi \mid = \sqrt{(\Re(\xi))^2 + (\Im(\xi))^2}$, the following is obtained.

$$\mid \xi \mid = \sqrt{\left(\frac{1+(\lambda+2\beta)\cos\omega}{1+\Delta t+\lambda+2\beta}\right)^2 + \left(\frac{\lambda\sin\omega}{1+\Delta t+\lambda+2\beta}\right)^2} \tag{11}$$

The UPFD method presented here guarantees the positivity of the solutions independent of the time step and mesh size. The UPFD method works with terms that are the sum of a positive and negative function of the unknown, and one or both may be zero. The scheme is positive definite for all values of $\Delta t, \Delta x > 0$, hence, the scheme is unconditionally stable for all $\Delta x, \Delta t > 0$.

2.1.2. Consistency

In this section, we investigate the consistency of Equation (6). We use the Taylor expansion to approximate the u_i^{n+1}, u_{i+1}^n, u_{i-1}^n, thus

$$\begin{aligned}
u_i^{n+1} &\approx u_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 u}{\partial t^2} + \frac{(\Delta t)^3}{6}\frac{\partial^3 u}{\partial t^3} + \cdots \\
u_{i-1}^n &\approx u_i^n - \Delta x\frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{(\Delta x)^3}{6}\frac{\partial^3 u}{\partial x^3} + \cdots \\
u_{i+1}^n &\approx u_i^n + \Delta x\frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{(\Delta x)^3}{6}\frac{\partial^3 u}{\partial x^3} + \cdots
\end{aligned} \tag{12}$$

The UPFD scheme terms are approximated by using Equation (12), which are then substituted into Equation (8) to obtain the truncation error for the UPFD method. After simplification, we obtain the following.

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} - u - \left(\Delta t + \frac{\Delta t}{\Delta x} + \frac{2\Delta t}{(\Delta x)^2}\right)\frac{\partial^2 u}{\partial t^2} - \left(\frac{\Delta t}{2} + \frac{(\Delta t)^2}{2} + \frac{(\Delta t)^2}{2(\Delta x)} + \frac{(\Delta t)^2}{(\Delta x)^2}\right)\frac{\partial^2 u}{\partial t^2} \\
- \left(\frac{(\Delta t)^2}{6} + \frac{(\Delta t)^3}{6} + \frac{(\Delta t)^3}{6(\Delta x)} + \frac{(\Delta t)^3}{3(\Delta x)^2}\right)\frac{\partial^3 u}{\partial t^3} + \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} - \frac{(\Delta x)^2}{6}\frac{\partial^3 u}{\partial x^3} + \cdots
\end{aligned} \tag{13}$$

The above scheme is not consistent when $\Delta t \to 0$, $\Delta x \to 0$; as a result, $\Delta t = (\Delta x)^3$ to obtain the following equation.

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} - u - \left((\Delta x)^3 + (\Delta x)^2 + 2(\Delta x)\right)\frac{\partial^2 u}{\partial t^2} \\
- \frac{(\Delta x)^3}{2}\left(1 + (\Delta x)^2 + 2\Delta x + (\Delta x)^3\right)\frac{\partial^2 u}{\partial t^2} \\
- \frac{(\Delta x)^6}{6}\left(1 + (\Delta x)^2 + 2\Delta x + (\Delta x)^3\right)\frac{\partial^3 u}{\partial t^3} + \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} - \frac{(\Delta x)^2}{6}\frac{\partial^3}{\partial x^3} + \cdots
\end{aligned} \tag{14}$$

when $\Delta x \longrightarrow 0$, $\Delta t \longrightarrow 0$, Equation (14) reduces to Equation (6); hence, the UPFD scheme is consistent when $\Delta t = (\Delta x)^3$.

2.2. Nonlinear Advection–Diffusion–Reaction Equation

In this section, we set $f(t,x,u) = ru(1-u)$, and $U_a = 0$ in Equation (1) to obtain the nonlinear reaction diffusion equation given below.

$$\frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial x^2} = ru(1-u) \qquad (x,t) \in [0,1] \times [0,T] \tag{15}$$

with initial and boundary conditions given by

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

$$\begin{aligned} u(0,t) &= 0, \qquad 0 \leq t \leq T \\ u(1,t) &= 0, \end{aligned}$$

where $\sigma = 80, x_0 = 0.5$ [7].

With the exact solution given as

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

where $D = 0.0002, r = 0.05$.

The discretization of the nonlinear ADR Equation (15) using Equation (5) is given below.

$$u_i^{n+1} = \frac{(1+r\Delta t)u_i^n + \lambda(u_{i+1}^n - u_i^n)}{1 + 2\lambda + r\Delta t u_i^n}. \tag{18}$$

where $\lambda = \dfrac{D\Delta t}{(\Delta x)^2}$.

2.2.1. Stability

By applying Fourier series analyses of the form $u_{i,n} = \xi^n e^{j\Delta t i \Delta x}$, and freezing coefficient technique [27] to Equation (18), we obtain the following amplification factor.

$$\xi = \frac{1 + \Delta t r + 2\lambda\cos\omega}{1 + 2\lambda + r\Delta t u_{\max}} \tag{19}$$

where by the freezing coefficient technique, u^2 is replace by uu_{max}, and u is frozen to a constant u_{max}, which represents the maximum value of the solution u.

The scheme is unconditionally stable for all Δx, $\Delta t > 0$.

2.2.2. Consistency

Through the consistency of the UPFD scheme [25] by the Taylor expansion to Equation (18), we obtain the following.

$$\begin{aligned} \frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial x^2} &= ru(1-u) - \frac{\Delta t}{2}\frac{\partial^2 u}{\partial x^2} - \left((\Delta x)^3 ru + 2\Delta x D\right)\frac{\partial u}{\partial t} \\ &- \frac{(\Delta x)^3}{2}\left(1 + 2\Delta x D + r(\Delta x)^3 u\right)\frac{\partial^2 u}{\partial t^2} - \frac{(\Delta x)^6}{6}\left(1 + 2\Delta x D + r(\Delta x)^3 u\right)\frac{\partial^3 u}{\partial t^3}\cdots\cdots \end{aligned} \tag{20}$$

As $\Delta x \longrightarrow 0$ and $\Delta t \longrightarrow 0$, Equation (20) reduces to Equation (15); therefore, the scheme is consistent.

3. Formulation of the POD Basis

In this section, we discuss the POD basis that reduces the degree of freedom for numerical methods [28].

The POD is used to reduce the computational time accumulated in the UPFD method. The EUPFD method is built by extracting a snapshot matrix from the matrix solution of the UPFD method. The snapshot matrix is then used to construct a small subset of leading basis functions, which is an eigenvector generated by singular value decomposition (SVD), to approximate state variables ψ_i. The SVD technique is used to decompose a matrix into several component matrices, exposing many of the original matrix's useful and interesting properties [29]. The POD method seeks a low-dimensional basis that could be used to approximate the state variables well. POD basis functions are typically the left singular vectors of the matrix.

From the standard UPFD solution, denoted by u_i^n, where $(1 \leq i \leq m), 0 \leq n \leq N$, we extract the first L sequence of solutions $\{u_i^n\}_{(n=1)}^{L}$ $(1 \leq i \leq m$, and $L \leq N)$ as snapshots, yielding the formulated $m \times L$ snapshot matrix shown below.

$$A = \begin{pmatrix} u_1^1 & u_1^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_1^L \\ u_2^1 & u_2^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_2^L \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & & & & & & \cdot & \cdot \\ u_m^1 & u_m^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_m^L \end{pmatrix} \tag{21}$$

where m and L are the mesh points of the snapshot. When the snapshot is tall and skinny, the method of snapshot is an efficient alternative way to compute the POD basis. The POD modes are computed by using the method of snapshots developed by Sirovich [30].

The SVD is then used to factorise snapshot matrix A.

$$A = U\Sigma V^T = U\begin{pmatrix} \Sigma_{l\times l} & O_{l\times(L-l)} \\ O_{(m-1)\times l} & O_{(m-l)\times(L-l)} \end{pmatrix} V^T \tag{22}$$

where l = $rankA$, $\Sigma_{l\times l}$ = $diag(\sigma_1, \sigma_2, \ldots, \sigma_l)$ is a diagonal matrix that is generated from the SVD of A as arranged from the decreasing order $\sigma_1 \geq \sigma_2 \geq \cdots\cdots \geq \sigma_l > 0$, $U = (\psi_1, \psi_2, \ldots, \psi_m)$ is an $m \times m$ orthogonal matrix consisting of the eigenvectors of AA^T, whereas $V = (\phi_1, \phi_2, \ldots, \phi_L)$ is an $L \times L$ orthogonal matrix consisting of the orthogonal eigenvectors of $A^T A$ and O is a zero matrix [31].

The eigenvalues of $A^T A$ and AA^T are identical and satisfy $\Lambda_j = \sigma_j^2 (j = 1, 2, \ldots, l)$. Therefore, we may first find the eigenvalues $\Lambda_1 \geq \Lambda_2 \geq \cdots\cdots \geq \Lambda_l > 0 (l = rankA)$ for matrices AA^T and corresponding eigenvectors ψ_j. That means that the following relationship applies.

$$\psi_j = A\phi_j / \sqrt{\Lambda_j}, j = 1, 2, \cdots\cdots, l. \tag{23}$$

Eigenvectors $\phi_j (j = 1, 2, \cdots\cdots, l)$ that correspond to the nonzero eigenvalues for matrix AA^T are obtained. Therefore,

$$A_M = U\Sigma V^T = U\begin{pmatrix} \Sigma_{M\times M} & O_{M\times(L-M)} \\ O_{(m-M)\times M} & O_{(m-M)\times(L-M)} \end{pmatrix} V^T \tag{24}$$

where $\Sigma_{M\times M} = diag(\sigma_1, \sigma_1, \cdots, \sigma_m)$ represents the diagonal matrix that consists of the first M positive singular values of diagonal matrix $\Sigma_{l\times l}$. The 2-norm of matrix A is defined as follows: $\|A\|_2 = sup_{u\in\Re^L}\|Au\|_2/\|u\|_2$ (where $\|u\|_2$ is the l^2 norm for vector u).

Theorem 1. *Let $\phi = (\psi_1, \psi_2, \cdots, \psi_M)$ consist of the first M eigenvectors $U = (\psi_1, \psi_2, \cdots, \psi_m)$; then, we have*

$$A_M = \sum_{i=1}^{M} \sigma_i \psi_i \phi_i^T = \phi\phi^T A. \tag{25}$$

Proof.

$$A_M = \sum_{i=1}^{M} \sigma_i \psi_i \phi_i^T = (\psi_1, \cdots\cdots, \psi_M) \begin{pmatrix} \sigma_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & & & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_M \end{pmatrix} \begin{pmatrix} \phi_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_M^T \end{pmatrix} \tag{26}$$

$$A = (\psi_1, \cdots\cdots, \psi_l) \begin{pmatrix} \sigma_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & & & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_l \end{pmatrix} \begin{pmatrix} \phi_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_l^T \end{pmatrix} \tag{27}$$

$$\phi\phi^T A = \phi \begin{pmatrix} \psi_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_M^T \end{pmatrix} (\psi_1, \cdots\cdots, \psi_l) \begin{pmatrix} \sigma_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & & & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_l \end{pmatrix} \begin{pmatrix} \phi_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_l^T \end{pmatrix} \tag{28}$$

$$= \phi\begin{pmatrix} I_M & O \end{pmatrix}\begin{pmatrix} \sigma_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & & & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_l \end{pmatrix}\begin{pmatrix} \phi_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_l^T \end{pmatrix} \tag{29}$$

$$= \phi\begin{pmatrix} \sigma_1 & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & \cdot & & & & \cdot \\ \cdot & & & & \cdot & & & & \cdot \\ \cdot & & & & \cdot & & & & \cdot \\ \cdot & & & & \cdot & & & & \cdot \\ \cdot & & & & \cdot & & & & \cdot \\ \cdot & & & & \cdot & & & & \cdot \\ 0 & \cdot & \cdot & \cdot & \sigma_M & \cdot & \cdot & \cdot & \sigma_l \end{pmatrix}\begin{pmatrix} \phi_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_l^T \end{pmatrix} \tag{30}$$

$$= (\psi_1, \cdots\cdots, \psi_M)\begin{pmatrix} \sigma_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & & & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_M \end{pmatrix}\begin{pmatrix} \phi_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_M^T \end{pmatrix} = A_M \tag{31}$$

Thus, by the relationship between the matrix norm and its spectral radius, we have

$$\min_{rank(B)\le M} \|A - B\|_{2,2} = \|A - A_M\|_{2,2} = \|A - \phi\phi^T A\|_{2,2} = \sqrt{\Lambda_{M+1}} \tag{32}$$

where $\phi = (\phi_1, \phi_2, \cdots\cdots, \phi_M)$ consist of the first M eigenvectors $U = (\psi_1, \psi_2, \cdots\cdots, \psi_m)$.

If the L column vectors of A are denoted by $u^n = (u_1^n, u_2^n, \cdots\cdots, u_m^n)^T (n = 1, 2, \cdots\cdots, L)$ we have

$$\|u^n - u_M^n\|_2 = \|(A - \phi\phi^T A)\xi_n\|_2 \le \|A - \phi\phi^T A\|_{2,2}\|\xi\|_2 = \sqrt{\Lambda_{M+1}}, \tag{33}$$

where

$$u_M^n = \sum_{j=1}^{M} (\psi_j, u^n)\psi_j \tag{34}$$

which represents the projection of u^n onto $\phi = (\phi_1, \phi, \cdots\cdots, \psi_M)(\psi_j, u^n)$ is the inner product of ψ_j and u^n, and ξ_n denotes the unit vector with the nth component being 1. Inequality (33) shows that u_M^n is an optimal approximation u^n whose error is no more than $\sqrt{\Lambda_{M+1}}$; thus, ϕ is just an orthogonal optimal POD base of A. □

3.1. Formulation of the POD to Reduce the Order of the Finite Difference Scheme for the Advection–Diffusion–Reaction Equations

The EUPFD method is built on the basis of POD. The solution vector of the UPFD method is given by $u^n = (u_1^n, u_2^n, \dots, u_m^n)^T (n = 1, 2, \cdots\cdots, L, L+1, \cdots\cdots, N)$ and have a finite difference scheme, as follows:

$$u^{n+1} = u^n + \frac{\Delta t}{\Delta x^2} B u^n + \Delta t F^n \tag{35}$$

where $F^n = \left(f_1^n, f_2^n, \cdots\cdots, f_m^n\right)$ and

$$B = \begin{pmatrix} -2 & 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & 1 & -2 & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & & & & & & & & \cdot \\ \cdot & & & & & & & & \cdot \\ \cdot & & & & & & & & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & -2 & 1 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 1 & -2 \end{pmatrix} \tag{36}$$

Matrix B is a tridiagonal matrix with the following eigenvalues:

$$\Lambda_j = -2 + 2\cos[(2j-1)\pi/(2m+1)], \qquad j = 1, 2, \cdots\cdots\cdots, m. \tag{37}$$

Taking the norm of matrix B, we obtain the following.

$$\begin{aligned} \|B\|_{2,2} &= |2 - 2\cos[(2m-1)\pi/(2m+1)]| \\ &= |2 - 2\cos[\pi - 2\pi/(2m+1)]| \\ &= |2 + 2\cos[2\pi/(2m+1)]| < 4 \end{aligned} \tag{38}$$

Replacing u^n in Equation (35) with $u^{*n} = \phi\phi^T A\xi_n (n = 1, 2, \cdots\cdots, L)$, which is $u^{*n} = \phi\alpha^n (n = L+1, L+2, \cdots\cdots, N)$, where $\alpha^n = \phi^T A\xi_n$, we obtain the EUPFD scheme as follows.

$$\begin{cases} u^{*n} = \phi\phi^T A\xi_n, \cdots\cdots, n = 1, 2, \cdots\cdots, L \\ \phi\alpha^{n+1} = \phi\alpha^n + \dfrac{\Delta t}{\Delta x^2} B\phi\alpha^n + \Delta t F^n, \\ n = L, L+1, \cdots\cdots, N-1 \end{cases} \tag{39}$$

where $\alpha^n = (\alpha_1^n, \alpha_2^n, \cdots\cdots, \alpha_M^n)^T$ are vectors that are yet to be determined.

Using orthogonal vectors in ϕ^T multiplied by (39), we obtain

$$\begin{cases} \alpha^n = \phi^T u^n, , \cdots\cdots, n = 1, 2, \cdots\cdots, L \\ \alpha^{n+1} = \alpha^n + \dfrac{\Delta t}{\Delta x^2} \phi^T B\phi\alpha^n + \Delta t \phi^T F^n, \\ n = L, L+1, \cdots\cdots, N-1 \end{cases} \tag{40}$$

We obtain the $\alpha^n (n = L, L+1, \cdots\cdots, N)$ from Equation (40); then, we obtain the EUPFD solution as follows.

$$u^{*n} = \phi\alpha^n, \qquad n = 1, 2, \cdots\cdots, N. \tag{41}$$

Therefore, the EUPFD solution component of Equation (6) is given as follows.

$$u_{j,k}^{*n} = u_i^{*n}, \cdots\cdots 0 \le j \le J, 0 \le k \le K, \ \ 1 \le i = k(J+1) + j + 1 \le m = (K+1)(J+1)$$

3.2. Error Estimates of the EUPFD for the Advection–Diffusion–Reaction Equation

By finding the norm of Equations (39) and (40), we obtain the following.

$$\|u^n - u^{*n}\|_2 = \|u^n - u^n_M\|_2 = \|(A - \phi\phi^T A)\xi_n\|_2 \le \sqrt{\Lambda_{M+1}} \tag{42}$$

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

Theorem 2 (Accuracy)**.** *The errors between solutions u^n_i from the UPFD scheme and solution u^{*n}_i from the EUPFD scheme satisfy the following estimates:*

$$|u^n_i - u^{*n}_i| \le (1+\delta)^{n-L}\sqrt{\Lambda_M + 1} \quad for \quad (L+1 \le n \le N) \tag{43}$$

where $\delta = \dfrac{\Delta t}{\Delta x^2}\|B\|_{2,2}$.

The errors between exact solution $u(x,t)$ and solution u^n_m from the POD-reduced-order FD scheme satisfy the following estimates [31]

$$|u(x_i, t_n) - u^{*n}_i| = O\Big((1+\delta)^{n-L}\sqrt{\Lambda_M + 1}, \Delta t, \Delta x^2\Big), \quad 1 \le i \le M, \quad 1 \le n \le N. \tag{44}$$

Proof.

$$|u^n - u^{*n}|_2 = |u^n - u^n_M|_2 = |(A - \alpha\alpha^T A)\epsilon_n|_2 \le \sqrt{\Lambda_{M+1}}, \quad n = 1, 2, \cdots\cdots, L. \tag{45}$$

Equation (39) is rewritten as follows.

$$u^{*n+1} = u^{*n} + \frac{\Delta t}{(\Delta x)^2} + \Delta t F^n, \qquad n = L, L+1, \cdots, N-1. \tag{46}$$

By putting $\delta = \Delta t |B|_2 / (\Delta x)^2$, we obtain the following simplified equation of Equation (45).

$$|u^{n+1} - u^{*n+1}|_2 \le (1+\delta)|u^n - u^{*n}|_2 \tag{47}$$

$$\begin{aligned} &\le \cdots \\ &\le (1+\delta)^{n+1-L}|u^L - u^{*L}|_2 \\ &\le (1+\delta)^{n+1-L}\sqrt{\Lambda_{M+1}}, n = L, L+1, \cdots, N-1. \end{aligned} \tag{48}$$

□

Remark 1. *The error terms containing $\sqrt{\Lambda_{M+1}}$ are caused by the POD-based reduced-order for the classical FD scheme, which could be used to select the number of the POD basis, i.e., it is necessary to take M such that $\sqrt{\Lambda_{M+1}} = O(\Delta t, \Delta x^2)$. Whereas $E(n) = (1+\delta)^{n-L}$ $(L+1 \le n \le N)$ are caused by extrapolating iterations that could be used as the guide for renewing the POD basis, i.e., if $(1+\delta)^{n-L}\sqrt{\Lambda_{M+1}} > max(\Delta t, \Delta x^2)$ it is necessary to update the POD basis. If we take Λ_{M+1} that satisfies $(1+\delta)^{n-L}\sqrt{\Lambda_{M+1}} = O(\Delta t, \Delta x^2)$, then the EUPFD scheme converges; thus, we did not have to update the POD basis [31].*

3.3. Implementation of the EUPFD Scheme Algorithm for Advection–Diffusion–Reaction Equations

In order to facilitate the use of the EUPFD scheme for the linear and nonlinear ADR equation, the following implementation steps of the algorithm for the EUPFD scheme are useful [31].

Step 1: Standard UPFD computation and extraction of snapshots:
Write the standard UPFD scheme in vector form and find solution vectors $u^n = (u^n_1, u^n_2, \cdots, u^n_m)^T (n = 1, 2, \cdots, L)$ at the few L steps.

Step 2: Snapshot matrix A and eigenvalues of A^TA: Formulate snapshot matrix $A = (u_i^n)_{x\times L}$ and compute eigenvalues $\Lambda_1 \geq, \Lambda_2 \geq, \cdots, \Lambda_l > 0 (l =$Rank $A)$ and eigenvectors $\phi_j (j = 1, 2, \cdots, \tilde{M})$ of matrix A^TA.
Step 3: Choice of POD basis: For error tolerance $\mu = O(\Delta t, \Delta x^2)$, decide the numbers $M \leq \tilde{M}$ of POD basis, such that $\sqrt{\Lambda_{u(M+1)}} \leq \mu$, and formulate the POD bases $\phi = (\psi_1, \psi_2, \cdots, \psi_M)$(where $\psi_j = \dfrac{A\phi_j}{\sqrt{\Lambda_j}}, j = 1, 2, \cdots, M)$.
Step 4: Solve or compute the EUPFD model:
Solve Equation (40) and use (41) to obtain the reduced-order solution vectors of the EUPFD given in a form of $u^{*n} = (u_1^{*n}, u_2^{*n}, \cdots, u_m^{*n})$, further obtain the component forms $u_{j,k}^{*n} = u_i^{*n}$, where $(0 \leq j \leq J, 0 \leq k \leq K,\ \ i = k(J+1) + j + 1.\ 1 \leq i \leq m = (K+1)(J+1))$.
Step 5: Check accuracy and renew POD basis to continue:

Set $\delta = \dfrac{\Delta t \mid\mid B \mid\mid_{2,2}}{(\Delta x)^2}$. If $(1+\delta)^{n-L}\sqrt{\Lambda_{u(M+1)}} \leq \mu$, then $u^{*n} = (u_1^{*n}, u_2^{*n}, \cdots, u_m^{*n})^T$ is the solution vectors for the EUPFD scheme that satisfy the accuracy requirement. Else, if $(1+\delta)^{n-L}\sqrt{\Lambda_{u(M+1)}} > \mu$, put $u^1 = u^{*(n-L)}, u^2 = u^{*(n-L+1)}, \cdots, u^L = u^{*(n-1)}$ and return to step 2.

4. Numerical Results

This section presents solutions to the linear and nonlinear ADR equations using the UPFD and EUPFD. Exact solutions are also used for comparison purposes. Convergence rates, absolute errors, and computational time were calculated to evaluate the methods' performance. The numerical and analytical solutions are denoted by $\bar{u}_{i,n}$ and $u_{i,n}$, respectively. The error is denoted by $e_{i,n}$,and its magnitude is measured by the L_∞-norm given by the formula below.

$$||e_{i,n}||_\infty = ||u_{i,n} - \bar{u}_{i,n}||_\infty = Max|u_{i,n} - \bar{u}_{i,n}| \tag{49}$$

The convergence rate is calculated by using the formula below.

$$q(\Delta t_k) = \frac{log_2(\frac{e_k}{e_{k+1}})}{log_2(\frac{\Delta t}{\Delta t_{k+1}})} \quad \text{or} \quad q(\Delta x_k) = \frac{log_2(\frac{e_k}{e_{k+1}})}{log_2(\frac{\Delta x}{\Delta x_{k+1}})} \tag{50}$$

The accuracy of the EUPFD relies on the size of the snapshot, L. Therefore, finding the best snapshot size such that the accuracy is maximal is paramount. To determine the best snapshot size, we first set $L = 1$ and found the solution. We then produced increments of 1 on L until there was no significant change in the solution. For this, we set up a tolerance as follows.

$$||u_{exact} - u_L||_\infty \leq \epsilon, \quad L = 1, 2, 3, \ldots \tag{51}$$

where u_{exact} is the exact solution, and u_L is the solution at the specific value of L. The first value of L that satisfies (51) is the optimal value of L. In this work, we set ϵ to be 10^{-10}.

4.1. Numerical Results for the Linear Advection–Diffusion–Reaction Equation

The convergence rates of the UPFD and the EUPFD with respect to the time and spatial steps are shown in Tables 1 and 2. The convergence rates of UPFD and EUPFD decreased with a decrease in time and spatial step sizes. Table 3 compares the exact, UPFD, and the EUPFD solutions at $(t, x) = (0.008, 0.05), (0.016, 0.1)$, and $(0.032, 0.2)$, as well as the computational times for the UPFD and EUPFD methods. EUPFD took less computational time to converge to the solution of the linear ADR equation compared to the UPFD.

Table 1. Convergence rates for the linear ADR equation by the UPFD and EUPFD with respect to Δt and a fixed $\Delta x = 0.025$.

	$q(\Delta t_k)$	
Δt_k	**UPFD**	**EUPFD**
0.001	3.668	3.568
0.002	3.573	3.448
0.004	3.530	3.375

Table 2. Convergence rates for the linear ADR equation by the UPFD and EUPFD with respect to Δx and a fixed $\Delta t = 0.001$.

	$q(\Delta x_k)$	
Δx_k	**UPFD**	**EUPFD**
0.0125	3.269	3.320
0.025	3.001	3.075
0.05	3.074	3.146

Table 3. Numerical results of the linear ADR equation at a fixed $\Delta x = 0.025$ and $\Delta t = 0.004$.

(t, x)	**Exact Solution**	**UPFD Solution**	**Error**	**EUPFD Solution**	**Error**
$(0.008, 0.05)$	0.979	0.975	0.003	0.975	0.003
$(0.016, 0.1)$	0.938	0.928	0.010	0.928	0.010
$(0.032, 0.2)$	0.9003	0.883	0.016	0.883	0.016
Time		0.008816		0.004314	

In Table 4, we compare the infinity norms of the proposed EUPFD with the Crank–Nicolson method in solving the linear ADR equation. For this example, results show that Crank–Nicolson achieved better accuracy. However, Crank–Nicolson did not preserve the positivity of solutions, which is the main focus of this work.

Table 4. Infinity norm results of the linear ADR equation by using the exact, Crank Nicolson, and EUPFD solutions.

(t, x)	**Exact**	**Crank–Nicolson**	**Error**	**EUPFD**	**Error**
$u(:, 0.05)$	0.9231	0.9246	0.0015	0.9056	0.0175
$u(:, 0.1)$	0.9608	0.9621	0.0013	0.9071	0.0536
$u(:, 0.2)$	1.0000	0.9987	0.0013	0.9087	0.0913

Figure 1a shows the contribution of each POD basis mode. The first 12 modes contained the predominant information where the EUPFD met the theoretical accuracy requirement. The 12 modes were significant in optimizing the linear ADR equation. Figure 1b shows the decreasing error when the number of modes increased.

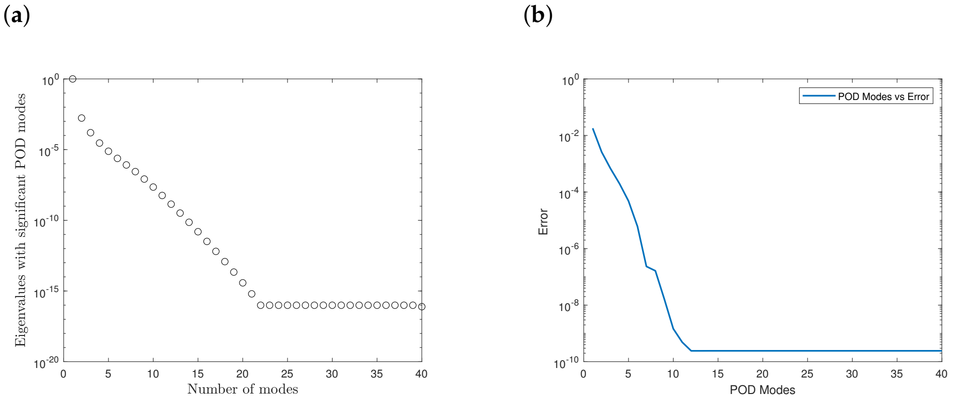

Figure 1. (**a**) POD modes for linear reaction diffusion equation. (**b**) Number of POD modes vs. error.

Figures 2a,b and 3a show the surface plots of the exact, UPFD and EUPFD solutions of linear ADR. Figure 3b shows the increase in error when the POD modes increased.

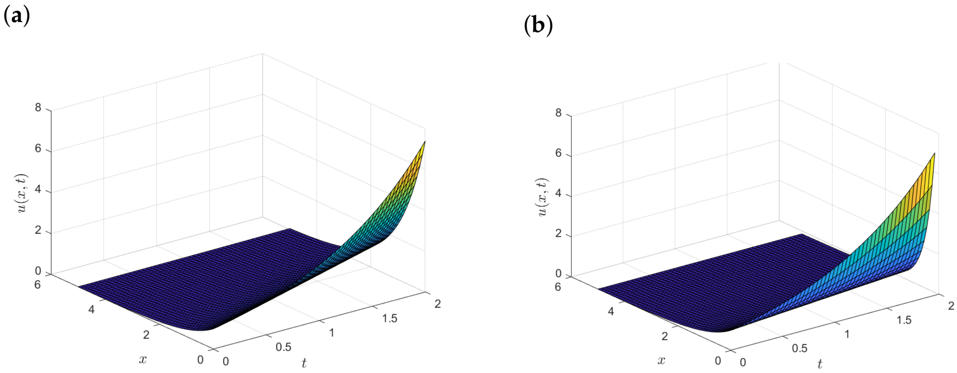

Figure 2. (**a**) Surface plot of the exact solution of the linear ADR equation. (**b**) Surface plot of the UPFD solution of the linear ADR equation.

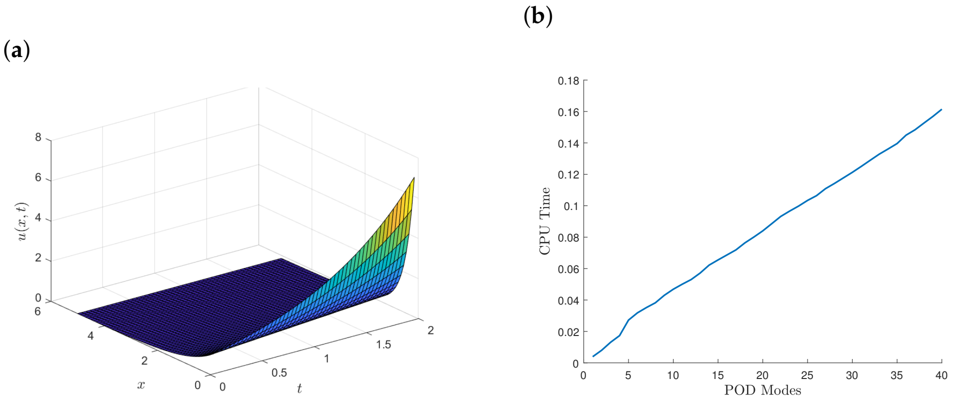

Figure 3. (**a**) Surface plot of the EUPFD solution of the linear ADR equation. (**b**) Number of POD modes vs. CPU time for the linear reaction diffusion equation.

Figure 4a,b show the absolute errors between the exact solution and EUPFD solution for the linear ADR against time and space, respectively.

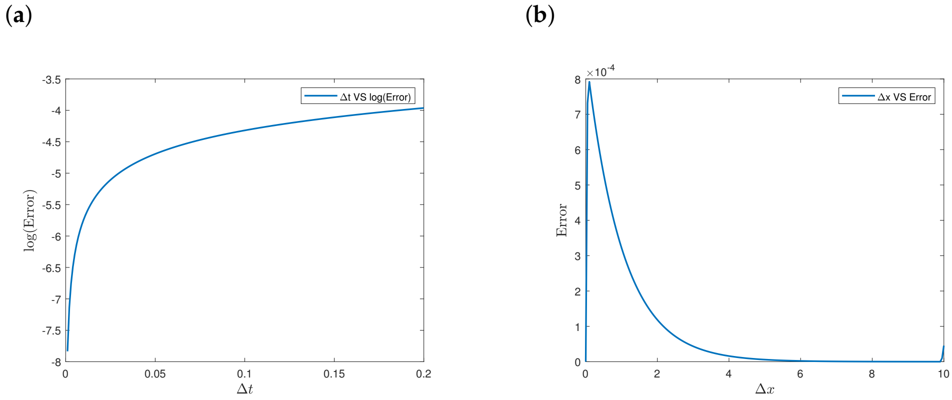

Figure 4. (**a**) Absolute error between the exact and EUPFD solutions of the linear ADR against time. (**b**) Absolute error between the exact and EUPFD solutions for the linear ADR against space.

4.2. Numerical Results for the Nonlinear Reaction Diffusion Equation

Tables 5 and 6 show the convergence rates of the UPFD and EUPFD for the nonlinear reaction–diffusion equation with respect to time and spatial steps. In these tables, the convergence rate of the UPFD and the EUPFD decreases with the increasing time step, and increases with the increasing spatial step sizes.

Table 5. Convergence rates for the nonlinear diffusion reaction by the UPFD and EUPFD equations with respect to Δt and a fixed $\Delta x = 0.01$.

	$q(\Delta t_k)$	
Δt_k	**UPFD**	**EUPFD**
0.00625	2.041	2.041
0.0125	2.028	2.028
0.025	1.983 7	1.983

Table 6. Convergence rates for the nonlinear diffusion reaction equation by the UPFD and EUPFD with respect to Δx and a fixed $\Delta t = 0.025$.

	$q(\Delta x_k)$	
Δx_k	**UPFD**	**EUPFD**
0.0005	7.160	4.767
0.001	5.405	4.320
0.002	3.878	3.882

Table 7 shows the comparison of the exact, UPFD, and EUPFD solutions obtained at $(t, x) = (0.05, 0.02), (0.1, 0.04)$, and $(0.2, 0.08)$, and the computational times taken by the UPFD and EUPFD methods. The EUPFD took less computational time to converge to the solution of the nonlinear reaction–diffusion equation than the UPFD did.

Table 7. Numerical results of the nonlinear ADR equation at a fixed $\Delta x = 0.01$, $\Delta t = 0.025$, $r = 0.05$ and $D = 0.002$.

(t, x)	**Exact Solution**	**UPFD Solution**	**Error**	**EUPFD Solution**	**Error**
$(0.05, 0.02)$	3.191×10^{-9}	3.214×10^{-9}	2.281×10^{-11}	3.214×10^{-9}	2.285×10^{-11}
$(0.1, 0.04)$	1.488×10^{-8}	1.603×10^{-8}	1.152×10^{-9}	1.603×10^{-8}	1.152×10^{-9}
$(0.2, 0.08)$	6.508×10^{-8}	7.285×10^{-8}	7.764×10^{-9}	7.285×10^{-8}	7.764×10^{-9}
Time		0.004314		0.002357	

In this example, the infinity norms of the EUPFD and Crank–Nicolson errors are shown in Table 8. Results show that EUPFD was slightly more accurate than Crank–Nicolson.

Table 8. Infinity norm results of the nonlinear ADR equation by using the exact, Crank–Nicolson, and EUPFD solutions.

(t,x)	Exact	Crank Nicolson	Error	EUPFD	Error
$u(:,0.02)$	0.7103	0.6998	0.0105	0.7000	0.0101
$u(:,0.04)$	0.7103	0.6993	0.0110	0.7000	0.0101
$u(:,0.08)$	0.7103	0.6989	0.0114	0.7000	0.0101

Figures 5a,b and 6 shows the surface plots of the exact, UPFD and EUPFD solutions of the nonlinear ADR equation. Figure 6b shows the increase in CPU time when the number with the increase in POD modes.

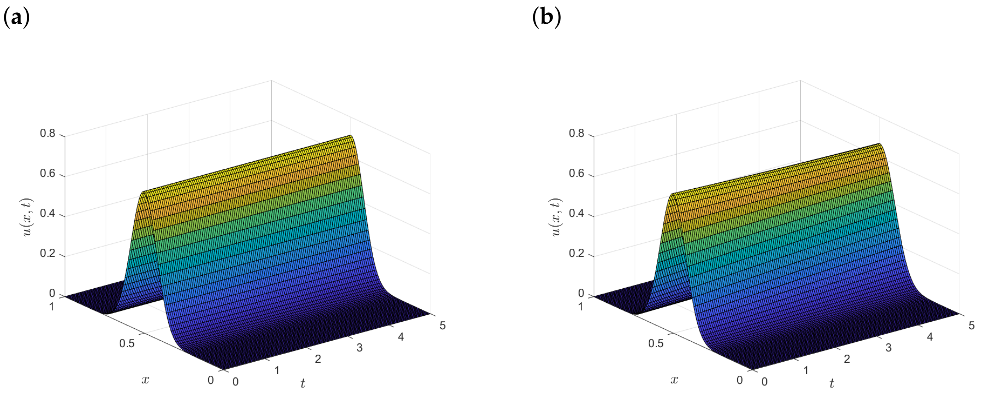

Figure 5. (**a**) Surface plot of the EUPFD solution of the nonlinear ADR equation. (**b**) Number of POD modes vs. CPU time for the nonlinear reaction diffusion equation.

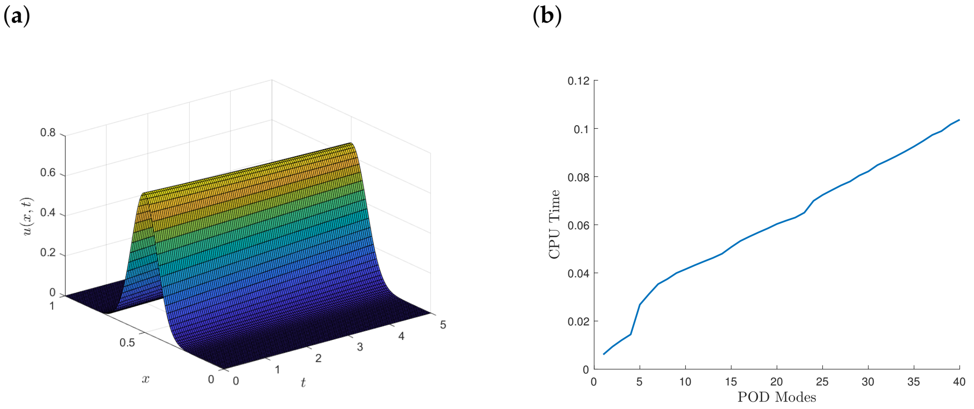

Figure 6. (**a**) Surface plot of the exact solution of the nonlinear reaction diffusion equation. (**b**) Surface plot of the UPFD solution of the nonlinear reaction diffusion equation.

Figure 7a shows the contribution of each of the POD basis modes. The first 9 modes contain the predominant information where the EUPFD met the theoretical accuracy requirement. The 9 modes were significant in optimising the nonlinear ADR equation.

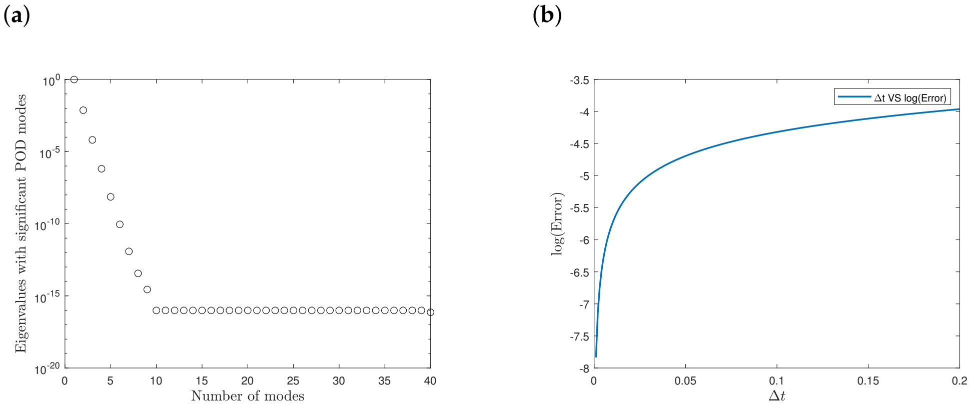

Figure 7. (**a**) Significant POD modes against number of modes. (**b**) Absolute error between the exact solution and EUPFD solution against time.

Figures 7b and 8 shows the absolute error between the exact and EUPFD solutions for the nonlinear ADR against time and space, respectively.

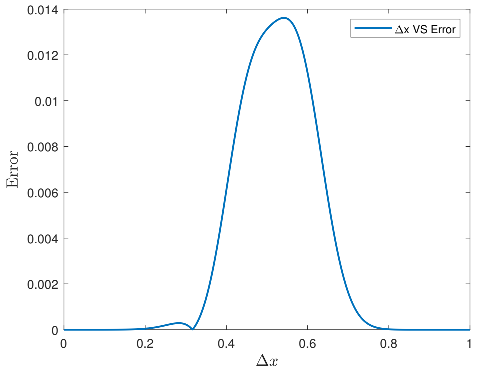

Figure 8. Absolute error between the exact and EUPFD solutions against space.

5. Conclusions

In this paper, we presented a new method, the enhanced unconditionally positive finite difference (EUPFD) method. The method uses the POD in conjunction with the unconditionally positive finite difference (UPFD) scheme. We used it to investigate the solutions to the linear and nonlinear diffusion reaction equations. The POD's goal is to find the UPFD's low-dimensional approximated solution. Results show that the EUPFD gives a good approximation to the solutions of these problems, and this validated its efficiency and effectiveness. Another remarkable result of the EUPFD is the reduced computational time compared to that of the UPFD. Besides the reduction in computational time and solution positivity that the EUPFD guarantees, only a small number of modes containing the predominant information are required to meet the theoretical accuracy. As the POD modes increased, the error of EUPFD measured against the exact solution significantly decreased. The spatial steps were taken to be $\Delta x = 0.01, 0.02$ and $0, 04$, while the time step was $\Delta t = 0.025, 0.0125$ and 0.00625. We had $\delta < 1$; thus, we next took 50 snapshots from the UPFD. By performing POD on UPFD, we achieved $\sqrt{\Lambda_9} \leq 4 \times 10^{-4}$ and $\sqrt{\Lambda_{35}} \leq 4 \times 10^{-4}$ for the linear and nonlinear ADR equations, respectively. As long as the first 9 and 12 POD bases were used, the theoretical accuracy requirement could be met. Since the EUPFD only used 9 and 12 POD modes from the selected 50 classical solutions of the UPFD as the snapshot, this would mean that EUPFD saves computational time, reduces degrees of freedom in numerical computations, alleviates truncation error accumulation, and it preserves the positivity of the solution. These render the EUPFD more efficient and effective in solving linear and nonlinear PDEs. However, the accuracy of the EUPFD was less than that of Crank–Nicolson when solving the linear ADR equation.

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: This research was funded by the National Research Foundation of South Africa under grant numbers 129119 and 127567, and the APC was funded by the University of Johannesburg.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Acknowledgments: The authors wish to acknowledge the financial support from the National Research Foundation of South Africa and the University of Johannesburg.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

UPFD	Unconditionally positive finite difference method
EUPFD	Enhanced unconditionally positive finite difference methods
POD	Proper orthogonal decomposition
CPU time	Computational time
PDEs	Partial differential equations
SVD	Singular value decomposition
t	Time variable
x	Spatial variable
Δx	Change in spatial variable
Δt	Change in time variable
U_a	Advection coefficient
D	Diffusion coefficient
u_t	Rate at which the concentration of substances changes over time
u_x	Advection term
u_{xx}	Diffusion term
ξ	Amplification factor
q	Convergence rate

References

1. Crank, J.; Nicolson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Math. Proc. Camb. Philos. Soc.* **1947**, *43*, 50–67. [CrossRef]
2. Bathe, K. Finite element method. In *Wiley Encyclopedia Of Computer Science and Engineering*; Wiley: Hoboken, NJ, USA, 2007; pp. 1–12.
3. Eymard, R.; Gallouët, T.; Herbin, R. Finite volume methods. In *Handbook Of Numerical Analysis*; Elsevier: Amsterdam, The Netherlands, 2000; Volume 7, pp. 713–1018.
4. Dlamini, P. *On Spectral Relaxation and Compact Finite Difference Schemes for Ordinary and Partial Differential Equations*; University of Johannesburg: Johannesburg, South Africa, 2015.
5. Mickens, R. *Applications of Nonstandard Finite Difference Schemes*; World Scientific: Hackensack, NJ, USA, 2000.
6. Angstmann, C.; Henry, B.; Jacobs, B.; McGann, A. Numeric solution of advection–Diffusion equations by a discrete time random walk scheme. *Numer. Methods Partial. Differ. Equ.* **2020**, *36*, 680–704. [CrossRef]
7. Chen-Charpentier, B.; Kojouharov, H. An unconditionally positivity preserving scheme for advection–diffusion reaction equations. *Math. Comput. Model.* **2013**, *57*, 2177–2185. [CrossRef]
8. Drljača, B.; Savović, S. Unconditionally positive finite difference and standard explicit finite difference schemes for power flow equation. *Univ.-Thought-Publ. Nat. Sci.* **2019**, *9*, 75–78. [CrossRef]
9. Saleh, M.; Kovács, E.; Pszota, G. Testing and improving a non-conventional unconditionally positive finite difference method. *Multidiszcip. Tudomanyok* **2020**, *10*, 206–213. [CrossRef]
10. Appadu, A. Analysis of the unconditionally positive finite difference scheme for advection-diffusion-reaction equations with different regimes. *Aip Conf. Proc.* **2016**, *1738*, 030005.

11. Towne, A.; Schmidt, O.; Colonius, T. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *J. Fluid Mech.* **2018**, *847*, 821–867. [CrossRef]
12. Wu, G.; Liang, Y.; Lin, W.; Lee, H.; Lim, S. A note on equivalence of proper orthogonal decomposition methods. *J. Sound Vib.* **2003**, *265*, 1103–1110. [CrossRef]
13. 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]
14. Abbaszadeh, M.; Dehghan, M. A POD-based reduced-order Crank-Nicolson/fourth-order alternating direction implicit (ADI) finite difference scheme for solving the two-dimensional distributed-order Riesz space-fractional diffusion equation. *Appl. Numer. Math.* **2020**, *158*, 271–291. [CrossRef]
15. 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]
16. Luo, Z. A POD-Based Reduced-Order Stabilized Crank–Nicolson MFE Formulation for the Non-Stationary Parabolized Navier–Stokes Equations. *Math. Model. Anal.* **2015**, *20*, 346–368. [CrossRef]
17. Luo, Z.; Jiang, W. A reduced-order extrapolated Crank–Nicolson finite spectral element method for the 2D non-stationary Navier-Stokes equations about vorticity-stream functions. *Appl. Numer. Math.* **2020**, *147*, 161–173. [CrossRef]
18. 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]
19. Luo, Z.; Xie, Z.; Shang, Y.; Chen, J. A reduced finite volume element formulation and numerical simulations based on POD for parabolic problems. *J. Comput. Appl. Math.* **2011**, *235*, 2098–2111. [CrossRef]
20. Britton, N. *Others Reaction-Diffusion Equations and Their Applications to Biology*; Academic Press: Cambridge, MA, USA, 1986.
21. Hao, X.; Liu, L.; Wu, Y. Iterative solution for nonlinear impulsive advection-reaction-diffusion equations. *J. Nonlinear Sci. Appl.* **2016**, *9*, 4070–4077. [CrossRef]
22. Liu, B.; Allen, M.; Kojouharov, H.; Chen, B. Others Finite-element solution of reaction-diffusion equations with advection. In *Computational Methods In Water Resources XI: WIT Press*. **1996**, *7*, 1–7.
23. 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]
24. 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]
25. Mphephu, N. Numerical Solution of 1-d Convection-Diffusion-Reaction Equation. Master’s Thesis, University of Pretoria, Pretoria, South Africa, **2013**.
26. Al’shin, A.; Al’shina, E.; Kalitkin, N.; Koryagina, A. Rosenbrock schemes with complex coefficients for stiff and differential algebraic systems. *Comput. Math. Math. Phys.* **2006**, *46*, 1320–1340. [CrossRef]
27. Wollkind, D.; Segel, L. A nonlinear stability analysis of the freezing of a dilute binary alloy. *Philos. Trans. R. Soc. Lond. Ser. Math. Phys. Sci.* **1970**, *268*, 351–380. [CrossRef]
28. Xia, H.; Luo, Z. An optimized finite difference Crank-Nicolson iterative scheme for the 2D Sobolev equation. *Adv. Differ. Equ.* **2017**, *2017*, 196. [CrossRef]
29. Stewart, G. On the early history of the singular value decomposition. *SIAM Rev.* **1993**, *35*, 551–566. [CrossRef]
30. Sirovich, L. Chaotic dynamics of coherent structures. *Phys. Nonlinear Phenom.* **1989**, *37*, 126–145. [CrossRef]
31. Luo, Z.; Chen, G. *Proper Orthogonal Decomposition Methods for Partial Differential Equations*; Academic Press: Cambridge, MA, USA, 2018; pp. 3–12.