Available online at www.garmian.edu.krd

Passer Journal

Passer 6 (Issue 2) (2024) 333-343

http://passer.garmian.edu.krd/

High-Order Solution of Reaction-Diffusion Systems Using Compact Finite Difference and Explicit Runge-Kutta Methods

Mardan A. Pirdawood[1,*], Younis A. Sabawi[1, 2], Mohammed I. Sadeeq[3], Bashdar O. Hussen[1], Yad A. Kareem[1], Hero M. Hussein[1] and Twana A. Ali[1]

[1]*Department of Mathematics, Faculty of Science and Health, Koya University, Koya KOY45, Kurdistan Region, Iraq.*
[2]*College of Information Technology, Imam Ja'afar AL-Sadiq University, Baghdad, Iraq.*
[3]*Department of Mathematics, College of Education, Akre University for Applied Sciences, Kurdistan Region, Iraq.*

Received 8 October 2023; revised 19 April 2024;
accepted 20 April 2024; available online 28 May 2024

DOI: 10.24271/PSR.2024.419907.1404

ABSTRACT

This study aims to develop a very accurate method for solving reaction-diffusion systems with Numman boundary conditions. Our approach combines an explicit Runge-Kutta method for the temporal aspect with a 4th-order compact finite difference (CFD) technique for spatial dimension. Moreover, we employ a 4th-order precise CFD technique to discretize the boundary points. At the core of our methodology lies the utilization of the method of lines (MOL). This approach strategically enables us to effectively incorporate explicit Runge-Kutta methods, known for their fifth-order precision in the temporal domain. Consequently, our approach achieves high-order precision in the temporal and spatial domains, which leads to a noteworthy reduction in the computational costs associated with the scheme. The combination of explicit Runge-Kutta methods and compact finite difference approaches yields a dependable and accurate solution to solve the reaction-diffusion system, according to numerical experiments.

Keywords: Explicit Runge-Kutta method, Compact finite difference method, Reaction-Diffusion System.

1. Introduction

The higher-order compact finite difference (HOCFD) method stands out as a widely employed and highly flexible technique for addressing partial differential equations (PDEs). Specifically, in the realm of linear and non-linear parabolic problems with Dirichlet boundary conditions, there has been a noticeable surge in interest, this has led to a substantial body of literature, including references[1], that comprehensively explains and makes accessible the method's implementation.

However, there has been comparatively less progress in the development of high-order numerical techniques to solve problems involving Neumann boundary conditions[1–3]. A fourth-order CFD technique was presented by Cao et al.[1] in order to solve the convection-diffusion system. Their approach utilized a 4th-order CFD method to discretize both boundary and interior points. Yao et al.[2] proposed a 4th-order CFD method to solve a simulated moving bed's model equation. They offered two distinct approaches, namely the pseudo grid point technique and the direct technique, to effectively handle the challenges posed by boundary conditions. A high-order exponential approach was used by Fu et al.[3] to solve the convection-diffusion equation. They developed a 4th-order compact exponential difference method.

Our aim is to propose a cutting-edge numerical approach to solve system (1) at 4th-order spatial and 5th-order temporal accuracy with particular components. Our approach involves discretizing the time integration within the spatial derivative with high-order precision using an explicit Runge-Kutta scheme and CFD approximation. This approach results in the formulation of a non-linear system of ODEs.

Even though the diffusive Lotka-Volterra System has been solved numerically before, as references[4] and[5] show, previous efforts mainly focused on achieving only 1st-order or 2nd-order precision at the boundary locations. The primary issues in our work are constructing a high order compact finite difference (HOCFD) approach to the boundary points and effectively resolving the non-linear reaction term. We use methods introduced by reference[1] to address boundary points and the ERK (explicit Runge-Kutta) method to solve the non-linear component to get over these challenges. It will be important to highlight that our adoption of the IMEX-RK (Implicit-Explicit Runge-Kutta) technique is motivated by the need to employ implicit techniques to solve the explicit methods for the non-linear term and linear term in the differential equations. This

* Corresponding author
E-mail address: mardan.ameen@koyauniversity.org(Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

approach enhances the overall stability and efficiency of our numerical method.

Because of their intricate nature, solving systems of ODEs can be difficult because analytical solutions are frequently unachievable. Moreover, these systems frequently involve multiple time scales that evolve simultaneously. Consequently, researchers have dedicated substantial effort to address these issues, resulting in the development of a range of numerical techniques over the years. Among these techniques, the Explicit Runge-Kutta method (ERK) has garnered significant attention for its effectiveness in solving problems expressed within a system of differential equations, such as equation (1). For a more comprehensive exploration of these numerical methods and their applications, please refer to[6]. Among these techniques, the Explicit Runge-Kutta method (ERK) has garnered significant attention for its effectiveness in solving problems expressed within a system of differential equations, such as equation (1). For a more comprehensive exploration of these numerical methods and their applications, please refer to[7-8].

The following outline will describe how the remaining sections of this work will be arranged: We present the model problem and describe a 4th-order CFD approach that can be applied to both interior and boundary points in the second section. In the third section, we delve into the exposition of the ERK technique. Section 4 is dedicated to showcasing practical numerical experiments. Lastly, Section 5 encapsulates our conclusions and summarizes the key findings derived from this study.

A groundbreaking theory regarding morphogenesis, the process through which structures develop within an organism throughout its life, was introduced in a significant publication by Turing in 1952[9]. From a mathematical standpoint, Turing's concept immediately opens the door to the creation of reaction-diffusion systems (RDSs) that go beyond individual equations. These systems are capable of demonstrating what is referred to as Turing instability, as exemplified in[10]. Contemporary non-linear models are employed to represent a broad spectrum of phenomena across medicine, ecology, chemistry, biology, physics, and other fields. Among these models, non-linear RDSs serve as the foundational equations for many widely recognized non-linear models. This study aims to develop an exact method for solving reaction-diffusion systems with Numman boundary conditions. Our approach combines explicit Runge-Kutta and 4th-order compact finite difference techniques for temporal and spatial aspects, respectively.

Consider the reaction-diffusion system (RDS):

$$u_t - \alpha u_{xx} = f\big(t, x, u(x,t), v(x,t)\big),$$

$$v_t - \beta v_{xx} = g\big(t, x, u(x,t), v(x,t)\big),$$

$$a \le x \le b, and\ 0 \le\ t \le T. \quad \text{.....1}$$

Depending on the following conditions:

The initial condition at t = 0:

$$\begin{aligned} u(x,0) &= u_0(x) = \bar{\psi}(x), \\ v(x,0) &= v_0(x) = \bar{\bar{\psi}}(x). \end{aligned} \quad \text{.....2a}$$

The Dirichlet boundary conditions at $x = a$ and $x = b$:

$$\begin{aligned} u(a,t) &= \bar{\emptyset}_a(t), \qquad u(b,t) = \bar{\emptyset}_b(t), \\ v(a,t) &= \bar{\bar{\emptyset}}_a(t), \qquad v(b,t) = \bar{\bar{\emptyset}}_b(t). \end{aligned} \quad \text{.....2b}$$

Moreover, the Neumann boundary conditions

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{x=a} &= \breve{u}(t), \qquad \left.\frac{\partial u}{\partial x}\right|_{x=b} = \hat{u}(t), \\ \left.\frac{\partial v}{\partial x}\right|_{x=a} &= \breve{v}(t), \qquad \left.\frac{\partial v}{\partial x}\right|_{x=b} = \hat{v}(t), \end{aligned} \quad \text{.....2c}$$

where $u(x,t)$ and $v(x,t)$ represent the concentrations or values of the first and second species at position x and time t, respectively. Also, α and β are parameters representing the diffusion coefficients for u and v, respectively. Furthermore, $f(u,v)$ and $g(u,v)$ are the reaction terms that describe how u and v interact with each other. These terms may involve non-linear functions of u and v, and they govern how the concentrations of u and v change over time.

2. Creation of Compact Finite Differencing Algorithms of Fourth-Order

2.1 The spatial interior points

Consider the system of differential equations

$$\begin{aligned} \alpha \frac{d^2U}{dx^2} + f &= l(x), \\ \beta \frac{d^2V}{dx^2} + g &= m(x), \end{aligned} \quad \text{.....3}$$

We designate δ_x^2 as the widely recognized central difference scheme for computing the second derivative of U and V at the location x_i within a uniform grid characterized by mesh sizes h. We will get the following discrete equations:

$$\alpha\delta_x^2 U_i + f - \tau_i^1 = l_i, \quad \text{.....4}$$

$$\beta\delta_x^2 V_i + g - \tau_i^2 = m_i, \quad \text{.....5}$$

where

$$\delta_x^2 U_i = \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + \frac{h^2}{12}\left[\frac{d^4U}{dx^4}\right]_i + O(h^4),$$

$$\delta_x^2 V_i = \frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} + \frac{h^2}{12}\left[\frac{d^4V}{dx^4}\right]_i + O(h^4), \quad \text{.....6}$$

$$\tau_i^1 = \frac{\alpha h^2}{12}\left[\frac{d^4U}{dx^4}\right]_i + O(h^4), \text{and } \tau_i^2 = \frac{\beta h^2}{12}\left[\frac{d^4V}{dx^4}\right]_i + O(h^4).$$

Our objective is to estimate the primary term in equation (6) and incorporate it into the difference formulation to achieve a method with $O(h^4)$ accuracy. Assuming the solution exhibits adequate regularity, we can attain this by taking the derivative of the system (3), resulting in:

$$\left.\frac{d^3U}{dx^3}\right|_i = \frac{1}{\alpha}\left[-\frac{df}{dx} + \frac{dl}{dx}\right]_i, \quad \text{.....7}$$

$$\left.\frac{d^4U}{dx^4}\right|_i = \frac{1}{\alpha}\left[-\frac{d^2f}{dx^2} + \frac{d^2l}{dx^2}\right]_i, \quad \text{.....8}$$

$$\left.\frac{d^3V}{dx^3}\right|_i = \frac{1}{\beta}\left[-\frac{dg}{dx}+\frac{dm}{dx}\right]_i, \quad \text{.....9}$$

$$\left.\frac{d^4V}{dx^4}\right|_i = \frac{1}{\beta}\left[-\frac{d^2g}{dx^2}+\frac{d^2m}{dx^2}\right]_i \quad \text{.....10}$$

By substituting (7) and (8) into the truncation error (τ_i^1), we will get the following:

$$\tau_i^1 = \frac{h^2}{12}\left[-\frac{d^2f}{dx^2}+\frac{d^2l}{dx^2}\right]_i + O(h^4) \quad \text{.....11}$$

Put the value of τ_i^1 in (4), it yields:

$$\alpha\delta_x^2 U_i + f - \frac{h^2}{12}[-\delta_x^2 f + \delta_x^2 l]_i = l_i + O(h^4), \quad \text{.....12}$$

Similarly, for the Eq. (4), we have:

$$\beta\delta_x^2 V_i + g - \frac{h^2}{12}[-\delta_x^2 g + \delta_x^2 m]_i = m_i + O(h^4), \quad \text{.....13}$$

which we can use (12) and (13) to increase the accuracy of our approximation (4) and (5). Our method produces a fourth-order compact scheme as a result. The problem's solution domain $[a,b]\times[0,T]$ is overlaid with a grid mesh of lines.

$$x_i = ih, i = 0,1,2,\cdots,n,$$

$$t_j = j\tau, j = 0,1,2,\cdots,m,$$

In the context of spatial and temporal coordinates, we establish parallel axes. To approximate the value $U(x_i,t_j)$, we perform calculations at the intersection point of these axes, denoted as (i,j) grid-point. The spatial and temporal grid spacings remain constant, where h represents the spatial grid spacing defined as $h=(b-a)/n$, and τ represents the temporal grid spacing defined as $\tau = T/m$. The discretization of the system (3) spatially, at the point through the equation (12), followed by subsequent simplification and substitution of l and m by u_t and v_t respectively, at the grid points, yields the following expression:

$$\left.\frac{dU(t)}{dt}\right|_{i-1} + (12\alpha-2)\left.\frac{dU(t)}{dt}\right|_i + \left.\frac{dU(t)}{dt}\right|_{i+1}$$

$$= \frac{12\alpha^2}{h^2}U_{i-1}(t) - \frac{24\alpha^2}{h^2}U_i(t) + \frac{12\alpha^2}{h^2}U_{i+1}(t) + f_{i-1} + (12\alpha - 2)f_i + f_{i+1}, \quad \text{.....14}$$

Also for the equation (13)

$$\left.\frac{dV(t)}{dt}\right|_{i-1} + (12\beta-2)\left.\frac{dV(t)}{dt}\right|_i + \left.\frac{dV(t)}{dt}\right|_{i+1}$$

$$= \frac{12\beta^2}{h^2}V_{i-1}(t) - \frac{24\beta^2}{h^2}V_i(t) + \frac{12\beta^2}{h^2}V_{i+1}(t) + g_{i-1} + (12\beta - 2)g_i + g_{i+1}, \quad \text{.....15}$$

2.2 The boundary spatial points

When the left and right boundaries were first deduced, the Neumann boundary conditions could be applied. For the left boundary conditions of $U(x,t)$, with the Taylor expansions of $U(x_1,t)$ and $U(x_2,t)$, the first derivative of $U(x,t)$ at x_0 can be expressed as:

$$\left.\frac{dU(t)}{dx}\right|_{i=0} = \left.\frac{-3U_i(t)+4U_{i+1}(t)-U_{i+2}(t)}{2h}\right|_{i=0} + \left.\frac{h^2}{3}\frac{d^3U(t)}{dx^3}\right|_{i=0} + \left.\frac{h^3}{4}\frac{d^4U(t)}{dx^4}\right|_{i=0} + O(h^4) \quad \text{.....16}$$

Also, for the right boundary conditions of $u(x,t)$, we have:

$$\left.\frac{dU(t)}{dx}\right|_{i=n} = \left.\frac{U_{i-2}(t)-4U_{i-1}(t)+3U_i(t)}{2h}\right|_{i=n} + \left.\frac{h^2}{3}\frac{d^3U(t)}{dx^3}\right|_{i=n} + \left.\frac{h^3}{4}\frac{d^4U(t)}{dx^4}\right|_{i=n} + O(h^4). \quad \text{.....17}$$

Substituting Eqs. (7) and (8) and the Neumann boundary conditions into (16)

$$6\left.\frac{dU(t)}{dt}\right|_{i=0} - 4\left.\frac{dU(t)}{dt}\right|_{i=1} - 2\left.\frac{dU(t)}{dt}\right|_{i=2}$$

$$= -\frac{36\alpha}{h^2}U_0(t) + \frac{48\alpha}{h^2}U_1(t) - \frac{12\alpha}{h^2}U_2(t) + 6f_0(t) - 4f_1(t) - 2f_2(t) - \frac{24\alpha}{h}\breve{u}(t). \quad \text{.....18}$$

Substituting Eqs. (9) and (10) and the Neumann boundary conditions into Eq. (16)

$$6\left.\frac{dV(t)}{dt}\right|_{i=0} - 4\left.\frac{dV(t)}{dt}\right|_{i=1} - 2\left.\frac{dV(t)}{dt}\right|_{i=2}$$

$$= -\frac{36\beta}{h^2}V_0(t) + \frac{48\beta}{h^2}V_1(t) - \frac{12\beta}{h^2}V_2(t) + 6g_0(t) - 4g_1(t) - 2g_2(t) - \frac{24\beta}{h}\breve{v}(t). \quad \text{.....19}$$

Similarly, one can obtain the right boundary conditions.

$$-10\left.\frac{dU(t)}{dt}\right|_{i=n-2} + 28\left.\frac{dU(t)}{dt}\right|_{i=n-1} - 18\left.\frac{dU(t)}{dt}\right|_{i=n}$$

$$= \frac{12\alpha}{h^2}U_{n-2}(t) - \frac{48\alpha}{h^2}U_{n-1}(t) + \frac{36\alpha}{h^2}U_n(t) - 10f_{n-2}(t) + 28f_{n-1}(t) - 18f_n(t) - \frac{24\alpha}{h}\hat{u}(t), \quad \text{.....20}$$

and

$$-10\left.\frac{dV(t)}{dt}\right|_{i=n-2} + 28\left.\frac{dV(t)}{dt}\right|_{i=n-1} - 18\left.\frac{dV(t)}{dt}\right|_{i=n}$$

$$= \frac{12\beta}{h^2}V_{n-2}(t) - \frac{48\beta}{h^2}V_{n-1}(t) + \frac{36\beta}{h^2}V_n(t) - 10g_{n-2}(t) + 28g_{n-1}(t) - 18g_n(t) - \frac{24\beta}{h}\hat{v}(t). \quad \text{.....21}$$

Therefore, by utilizing equation (14) along with the given boundary conditions, we derive a linear system of equations, which can be expressed in matrix form as a tri-diagonal linear system, as outlined below:

$$\boldsymbol{A}_1\frac{dU(t)}{dt} = \boldsymbol{B}_1\boldsymbol{U}(t) + \boldsymbol{F}(t) + \boldsymbol{H}_1(t), \quad \text{.....22}$$

where

$$\boldsymbol{A_1}=\begin{bmatrix}6 & -4 & -2 & & \\ 1 & (12\alpha-2) & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & (12\alpha-2) & 1 \\ & & -10 & 28 & -18\end{bmatrix},\boldsymbol{U}(t)=\begin{bmatrix}U_0(t)\\U_1(t)\\U_2(t)\\\vdots\\U_n(t)\end{bmatrix},\boldsymbol{B_1}=\begin{bmatrix}-\frac{36\alpha}{h^2} & \frac{48\alpha}{h^2} & -\frac{12\alpha}{h^2} & & \\ \frac{12\alpha^2}{h^2} & -\frac{24\alpha^2}{h^2} & \frac{12\alpha^2}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{12\alpha^2}{h^2} & -\frac{24\alpha^2}{h^2} & \frac{12\alpha^2}{h^2} \\ & & \frac{12\alpha}{h^2} & -\frac{48\alpha}{h^2} & \frac{36\alpha}{h^2}\end{bmatrix},$$

$$\boldsymbol{H}_1(t)=\begin{bmatrix}-\frac{24\alpha}{h}\check{u}(t)\\0\\0\\\vdots\\0\\-\frac{24\alpha}{h}\hat{u}(t)\end{bmatrix},\boldsymbol{F}(t)=\begin{bmatrix}6f_0(t) & -4f_1(t) & -2f_2(t) & & \\ f_0(t) & (12\alpha-2)f_1(t) & f_2(t) & & \\ & \ddots & \ddots & \ddots & \\ & & f_{n-2}(t) & (12\alpha-2)f_{n-1}(t) & f_n(t) \\ & & -10f_{n-2}(t) & 28f_{n-1}(t) & -18f_n(t)\end{bmatrix}.$$

Similarly, by applying equation (15) in conjunction with the provided boundary conditions, we deduce a linear system of equations that can be represented in matrix form as a multi-diagonal linear system, as described below:

$$\boldsymbol{A}_2\frac{d\boldsymbol{V}(t)}{dt}=\boldsymbol{B}_2\boldsymbol{V}(t)+\boldsymbol{G}(t)+\boldsymbol{H}_2(t),\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 23$$

where

$$\boldsymbol{A_2}=\begin{bmatrix}6 & -4 & -2 & & \\ 1 & (12\beta-2) & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & (12\beta-2) & 1 \\ & & -10 & 28 & -18\end{bmatrix},\boldsymbol{V}(t)=\begin{bmatrix}V_0(t)\\V_1(t)\\V_2(t)\\\vdots\\V_n(t)\end{bmatrix},\boldsymbol{B_2}=\begin{bmatrix}-\frac{36\beta}{h^2} & \frac{48\beta}{h^2} & -\frac{12\beta}{h^2} & & \\ \frac{12\beta^2}{h^2} & -\frac{24\beta^2}{h^2} & \frac{12\beta^2}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{12\beta^2}{h^2} & -\frac{24\beta^2}{h^2} & \frac{12\beta^2}{h^2} \\ & & \frac{12\beta}{h^2} & -\frac{48\beta}{h^2} & \frac{36\beta}{h^2}\end{bmatrix},$$

$$\boldsymbol{H}_2(t)=\begin{bmatrix}-\frac{24\alpha}{h}\check{v}(t)\\0\\0\\\vdots\\0\\-\frac{24\alpha}{h}\hat{v}(t)\end{bmatrix},\boldsymbol{G}(t)=\begin{bmatrix}6g_0(t) & -4g_1(t) & -2g_2(t) & & \\ g_0(t) & (12\beta-2)g_1(t) & g_2(t) & & \\ & \ddots & \ddots & \ddots & \\ & & g_{n-2}(t) & (12\beta-2)g_{n-1}(t) & g_n(t) \\ & & -10g_{n-2}(t) & 28g_{n-1}(t) & -18g_n(t)\end{bmatrix}.$$

For more simplicity, we can write the ordinary differential equations (22) and (23) as follows:

$$\begin{cases}\frac{d\boldsymbol{U}(t)}{dt}=\boldsymbol{A_1^{-1}B_1}\boldsymbol{U}(t)+\boldsymbol{A_1^{-1}H}_1(t)+\boldsymbol{A_1^{-1}F}(t)\\ \frac{d\boldsymbol{V}(t)}{dt}=\boldsymbol{A_2^{-1}B}_2\boldsymbol{V}(t)+\boldsymbol{A_2^{-1}H}_2(t)+\boldsymbol{A_2^{-1}G}(t)\end{cases}\ldots\ldots\ldots\ldots\ldots\ldots 24$$

3. Explicit Runge-Kutta Method Algorithms

The conventional sequence of steps in the explicit Runge-Kutta algorithm corresponds to an approximation of the initial terms present in an infinite Taylor series. This specific series is utilized for calculating the path taken by a moving point. Shampine and Gordon extensively investigated this concept in their research[10]. The local truncation error estimator (LTEE) refers to the portion of the infinite sum that remains after removal. These predictive forecasting techniques are known as explicit Runge-Kutta (ERK) algorithms. In essence, they forecast a point's future location independent of phase information from the past. Because of this characteristic, they require only a minimal amount of input data, making them straightforward to use and implement. The solution to an initial value problem is determined by utilizing the explicit Runge-Kutta (ERK) method, which incorporates p stages:

$$\begin{cases}\frac{d\boldsymbol{U}(t)}{dt}=\boldsymbol{\psi}(t,\boldsymbol{U}(t),\boldsymbol{V}(t)\,)\\ \frac{d\boldsymbol{V}(t)}{dt}=\boldsymbol{\phi}(t,\boldsymbol{U}(t),\boldsymbol{V}(t)\,)\end{cases}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 25$$

where $\boldsymbol{\psi}(t,\boldsymbol{U}(t),\boldsymbol{V}(t)\,)=\boldsymbol{A_1^{-1}B_1}\boldsymbol{U}(t)+\boldsymbol{A_1^{-1}H}_1(t)+\boldsymbol{A_1^{-1}F}(t)$ and $\boldsymbol{\phi}(t,\boldsymbol{U}(t),\boldsymbol{V}(t)\,)=\boldsymbol{A_2^{-1}B}_2\boldsymbol{V}(t)+\boldsymbol{A_2^{-1}H}_2(t)+\boldsymbol{A_2^{-1}G}(t)$ with the initial conditions are $\boldsymbol{U}(t_0)=\boldsymbol{u}_0$ and $\boldsymbol{V}(t_0)=\boldsymbol{v}_0$, will be determined as follows:

$$\begin{cases}\boldsymbol{U}_{n+1}=\boldsymbol{U}_n+\tau\sum_{i=1}^{p}b_i\boldsymbol{k}_i,\\ \boldsymbol{V}_{n+1}=\boldsymbol{V}_n+\tau\sum_{i=1}^{p}b_i\boldsymbol{k}\boldsymbol{k}_i,\end{cases}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 26$$

where $\boldsymbol{k}_i=\boldsymbol{\psi}\big(t_n+c_ih,\boldsymbol{U}+\tau\sum_{j=1}^{p}a_{ij}\boldsymbol{k}_j\,,\boldsymbol{V}+h\sum_{j=1}^{p}a_{ij}\boldsymbol{k}\boldsymbol{k}_j\big)$, $\boldsymbol{k}\boldsymbol{k}_i=\boldsymbol{\phi}\big(t_n+c_ih,\boldsymbol{U}+h\sum_{j=1}^{p}a_{ij}\boldsymbol{k}_j\,,\boldsymbol{V}+h\sum_{j=1}^{p}a_{ij}\boldsymbol{k}\boldsymbol{k}_j\big)$ and $c_i=\sum_{j=1}^{p}a_{ij},\ i=1,2,\ldots,p.$

Considering vectors c and b, both having p dimensions, along with matrix $A\big(a_{ij}\big)$ of size $p\times p$. However, smaller values of τ generally lead to more accurate results because they allow for more accurate approximations of the solution at each step. The

structure of the ERK algorithm for the system (24) can be described as follows:

$$\boldsymbol{k}_1 = \tau\boldsymbol{\psi}(t_n, \boldsymbol{U}_n, \boldsymbol{V}_n),$$
$$\boldsymbol{kk}_1 = \tau\boldsymbol{\phi}(t_n, \boldsymbol{U}_n, \boldsymbol{V}_n),$$
$$\boldsymbol{k}_2 = \tau\boldsymbol{\psi}\left(t_n + \frac{h}{4}, \boldsymbol{U}_n + \frac{\boldsymbol{k}_1}{4}, \boldsymbol{V}_n + \frac{\boldsymbol{kk}_1}{4}\right),$$
$$\boldsymbol{kk}_2 = \tau\boldsymbol{\phi}\left(t_n + \frac{h}{4}, \boldsymbol{U}_n + \frac{\boldsymbol{k}_1}{4}, \boldsymbol{V}_n + \frac{\boldsymbol{kk}_1}{4}\right),$$
$$\boldsymbol{k}_3 = \tau\boldsymbol{\psi}\left(t_n + \frac{h}{4}, \boldsymbol{U}_n + \frac{\boldsymbol{k}_1}{8} + \frac{\boldsymbol{k}_2}{8}, \boldsymbol{V}_n + \frac{\boldsymbol{kk}_1}{8} + \frac{\boldsymbol{kk}_2}{8}\right),$$
$$\boldsymbol{kk}_3 = \tau\boldsymbol{\phi}\left(t_n + \frac{h}{4}, \boldsymbol{U}_n + \frac{\boldsymbol{k}_1}{8} + \frac{\boldsymbol{k}_2}{8}, \boldsymbol{V}_n + \frac{\boldsymbol{kk}_1}{8} + \frac{\boldsymbol{kk}_2}{8}\right),$$
$$\boldsymbol{k}_4 = \tau\boldsymbol{\psi}\left(t_n + \frac{h}{2}, \boldsymbol{U}_n - \frac{\boldsymbol{k}_2}{2} + \boldsymbol{k}_3, \boldsymbol{V}_n - \frac{\boldsymbol{kk}_2}{2} + \boldsymbol{kk}_3\right),$$
$$\boldsymbol{kk}_4 = \tau\boldsymbol{\phi}\left(t_n + \frac{h}{2}, \boldsymbol{U}_n - \frac{\boldsymbol{k}_2}{2} + \boldsymbol{k}_3, \boldsymbol{V}_n - \frac{\boldsymbol{kk}_2}{2} + \boldsymbol{kk}_3\right),$$
$$\boldsymbol{k}_5 = \tau\boldsymbol{\psi}\left(t_n + \frac{3h}{4}, \boldsymbol{U}_n + \frac{3\boldsymbol{k}_1}{16} + \frac{9\boldsymbol{k}_4}{16}, \boldsymbol{V}_n + \frac{3\boldsymbol{kk}_1}{16} + \frac{9\boldsymbol{kk}_4}{16}\right),$$
$$\boldsymbol{kk}_5 = \tau\boldsymbol{\phi}\left(t_n + \frac{3h}{4}, \boldsymbol{U}_n + \frac{3\boldsymbol{k}_1}{16} + \frac{9\boldsymbol{k}_4}{16}, \boldsymbol{V}_n + \frac{3\boldsymbol{kk}_1}{16} + \frac{9\boldsymbol{kk}_4}{16}\right),$$
$$\boldsymbol{k}_6 = \tau\boldsymbol{\psi}\left(t_n + h, \boldsymbol{U}_n - \frac{3\boldsymbol{k}_1}{7} + \frac{2\boldsymbol{k}_2}{7} + \frac{12\boldsymbol{k}_3}{7} - \frac{12\boldsymbol{k}_4}{7} + \frac{8\boldsymbol{k}_5}{7}, \boldsymbol{V}_n - \frac{3\boldsymbol{kk}_1}{7} + \frac{2\boldsymbol{kk}_2}{7} + \frac{12\boldsymbol{kk}_3}{7} - \frac{12\boldsymbol{kk}_4}{7} + \frac{8\boldsymbol{kk}_5}{7}\right),$$
$$\boldsymbol{kk}_6 = \tau\boldsymbol{\phi}\left(t_n + h, \boldsymbol{U}_n - \frac{3\boldsymbol{k}_1}{7} + \frac{2\boldsymbol{k}_2}{7} + \frac{12\boldsymbol{k}_3}{7} - \frac{12\boldsymbol{k}_4}{7} + \frac{8\boldsymbol{k}_5}{7}, \boldsymbol{V}_n - \frac{3\boldsymbol{kk}_1}{7} + \frac{2\boldsymbol{kk}_2}{7} + \frac{12\boldsymbol{kk}_3}{7} - \frac{12\boldsymbol{kk}_4}{7} + \frac{8\boldsymbol{kk}_5}{7}\right).$$

Predictor using a 5th-order Explicit Runge-Kutta method with six stages:

$$\begin{cases} \boldsymbol{U}_{n+1} = \boldsymbol{U}_n + \frac{1}{90}(7\boldsymbol{k}_1 + 32\boldsymbol{k}_3 + 12\boldsymbol{k}_4 + 32\boldsymbol{k}_5 + 7\boldsymbol{k}_6), \\ \boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \frac{1}{90}(7\boldsymbol{kk}_1 + 32\boldsymbol{kk}_3 + 12\boldsymbol{kk}_4 + 32\boldsymbol{k}k_5 + 7\boldsymbol{kk}_6). \end{cases} \quad \ldots\ldots 27$$

The accuracy of the methods is assessed by calculating the absolute error, denoted as L_{abs}, which is defined as follows:

$$L_{\text{abs}}(u) = \left|u_{\text{Exact}}(x_i, t) - U_{\text{Approximate}}(x_i, t)\right|, \quad i = 1,2,\ldots,n, \quad \ldots\ldots 28$$

$$L_{\text{abs}}(v) = \left|v_{\text{Exact}}(x_i, t) - V_{\text{Approximate}}(x_i, t)\right|, \quad i = 1,2,\ldots,n. \quad \ldots\ldots 29$$

When exact solutions are unavailable, we rely on convergence studies, known solutions for simplified cases, error estimation, and physical reasoning for validation. The arrangement of the Butcher array in equation (26) assumes the subsequent configuration:

0						
$\frac{1}{4}$	$\frac{1}{4}$					
$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$				
$\frac{1}{2}$	0	$-\frac{1}{2}$	1			
$\frac{3}{4}$	$\frac{3}{16}$	0	0	$\frac{9}{16}$		
0	$-\frac{3}{7}$	$\frac{2}{7}$	$\frac{12}{7}$	$-\frac{12}{7}$	$\frac{8}{7}$	
	$\frac{7}{90}$	0	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$

4. Numerical Results

We will now demonstrate the effectiveness of the higher order compact finite difference (HOCFD) method by implementing it using Matlab programming. Various functions for $f(t, x, u(x,t), v(x,t))$, and $g(t, x, u(x,t), v(x,t))$ have been chosen in this study.

Example 4.1: If we take $f(t, x, u(x,t), v(x,t)) = a_1u - bu^2 - cvu$ and $g(t, x, u(x,t), v(x,t)) = a_2v - buv - cv^2$, then the RDS (1) will become:

$$\begin{cases} u_t - \alpha u_{xx} = a_1u - bu^2 - cvu, \\ v_t - \beta u_{xx} = a_2v - buv - cv^2, \end{cases} \quad \ldots\ldots 30$$

with the exact solution is given in the reference[11]:

$$\begin{cases} u(x,t) = \frac{a_1e^{a_1t}}{2+be^{a_1t}+7e^{a_2t}}\left(1 + \frac{1}{2}\exp\left(\frac{\alpha(a_2-a_1)}{\alpha-\beta}\right)\sin\left(\sqrt{\frac{a_1-a_2}{\alpha-\beta}}x\right)\right), \\ v(x,t) = \frac{1}{c}\frac{a_1be^{a_1t}+7a_2e^{a_2t}}{be^{a_1t}+7e^{a_2t}+2} - \frac{b}{c}u(x,t). \end{cases} \quad \ldots\ldots 31$$

Example 4.2: If we take $f(t, x, u(x,t), v(x,t)) = a_1u - b_1u^2 - cvu$ and $g(t, x, u(x,t), v(x,t)) = -a_2v + b_2vu - 3cv^2$, then the RDS (1) will become:

$$\begin{cases} u_t - \alpha u_{xx} = a_1u - b_1u^2 - cvu, \\ v_t - \beta u_{xx} = -a_2v + b_2vu - 3cv^2, \end{cases} \quad \ldots\ldots 32$$

with the exact solution is given in the reference[11]:

$$\begin{cases} u(x,t) = \frac{3a_1+a_2}{2(3b_1+b_2)}\left(1 + \tanh\left(\sqrt{\frac{a_1b_2-a_2b_1}{8(3b_1+b_2)}}(x-\gamma t)\right)\right), \\ v(x,t) = \frac{a_1b_2-a_2b_1}{4c(3b_1+b_2)}\left(1 + \tanh\left(\sqrt{\frac{a_1b_2-a_2b_1}{8(3b_1+b_2)}}(x-\gamma t)\right)\right)^2, \end{cases} \quad \ldots 33$$

where $\beta = \frac{a_2b_1-3a_1(2b_1+b_2)}{a_2(5b_1+b_2)-2a_1b_2} > 0$, and $\gamma = \frac{a_2b_1-3a_1(2b_1+b_2)}{\sqrt{2(3b_1+b_2)(a_1b_2-a_2b_1)}}$.

Example 4.3: If we take $f(t, x, u(x,t), v(x,t)) = \alpha(a_1u - bu^2 - cvu)$ and $g(t, x, u(x,t), v(x,t)) = \beta(a_2v - bvu - cv^2)$, then the RDS (1) will become:

$$\begin{cases} u_t - \alpha u_{xx} = \alpha(a_1u - bu^2 - cvu), \\ v_t - \beta u_{xx} = \beta(a_2v - bvu - cv^2), \end{cases} \quad (34)$$

with the exact solution is given in the reference [11]:

$$\begin{cases} u(x,t) = \frac{a_1}{b} - \frac{1}{2(a_1-a_2)b}\sin\left(\sqrt{-\frac{\gamma}{\alpha}}x\right)e^{\gamma t}, \\ v(x,t) = -\frac{1}{2(a_2-a_1)c}\sin\left(\sqrt{-\frac{\gamma}{\alpha}}x\right)e^{\gamma t}, \end{cases} \quad (35)$$

where $\gamma = \frac{(a_1-a_2)\alpha\beta}{\beta-\alpha} < 0$.

Table 1: L_{abs} errors between the solutions (27) and (31) for system (30) using the Higher Order Compact Finite Difference (HOCFD) method at t=1, where α =3/2, β =1, a_1=2, a_2=1, b=3, and c=5.

x	$u(x,t)$ Exact	$U(x,t)$ HOCFD	$L_{abs}(u)$	$v(x,t)$ Exact	$V(x,t)$ HOCFD	$L_{abs}(v)$
-5	0.3361	0.3361	0	0.0917	0.0917	0
-4	0.3471	0.3471	1.1295e-5	0.0851	0.0851	2.9512e-5
-3	0.3497	0.3498	1.6255e-5	0.0835	0.0835	4.2834e-5
-2	0.3395	0.3395	5.8029e-6	0.0897	0.0897	1.6608e-5
-1	0.3337	0.3337	1.9242e-5	0.0931	0.0932	5.2365e-5
0	0.3421	0.3421	2.8472e-7	0.0881	0.0881	2.0024e-6
1	0.3505	0.3506	1.7324e-5	0.0831	0.0830	4.5400e-5
2	0.3447	0.3448	5.1121e-6	0.0865	0.0865	1.2125e-5
3	0.3345	0.3345	1.8010e-5	0.0927	0.0927	4.9977e-5
4	0.3371	0.3371	1.2394e-5	0.0911	0.0911	3.5790e-5
5	0.3482	0.3482	0	0.0845	0.0845	0

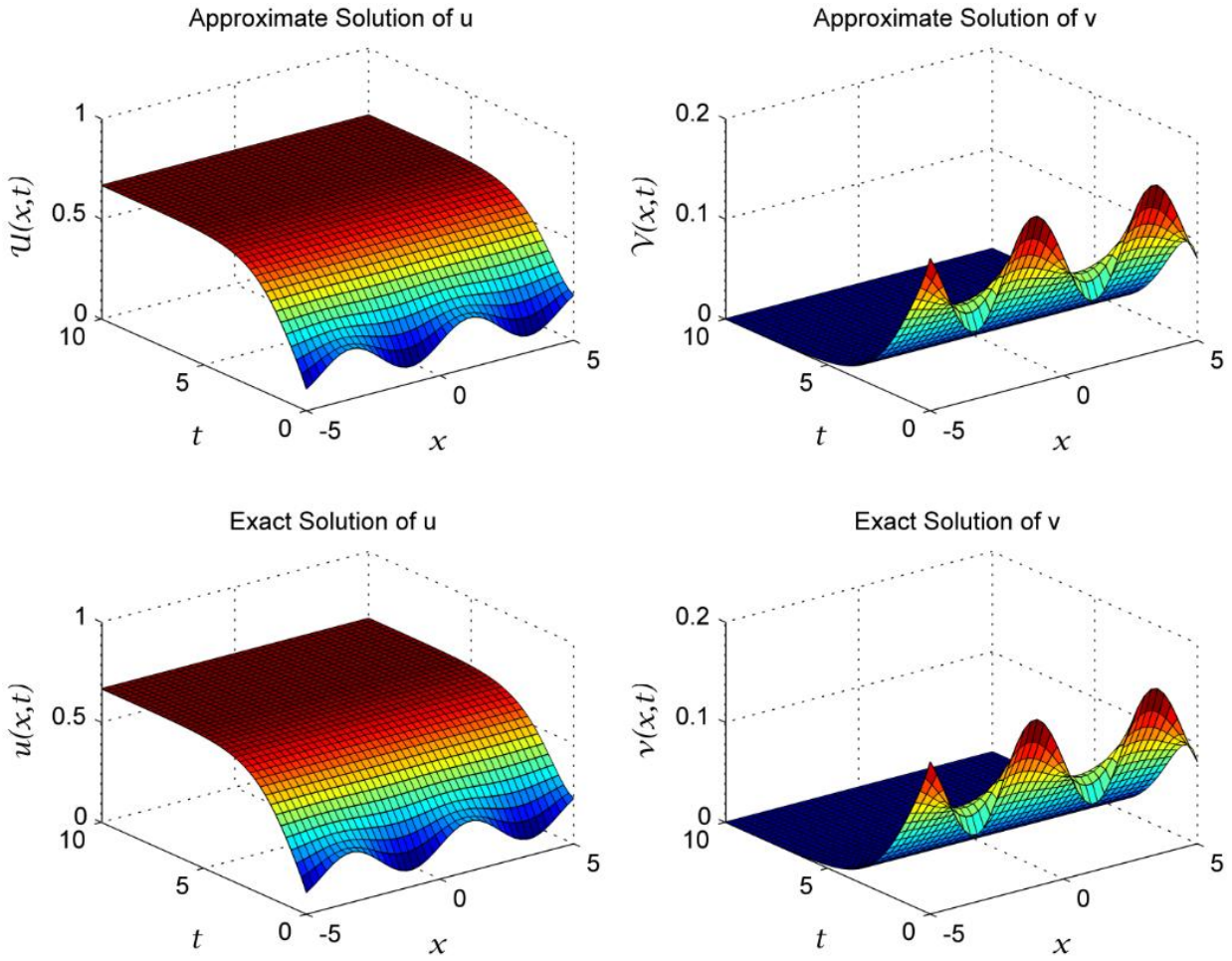

Figure 1: Representational surfaces depicting the constituents u and v within solution compositions (27) and (31) of the system (30), with α =3/2, β =1, $a_1 = 2$, $a_2 = 1$, b=3, and c=5.

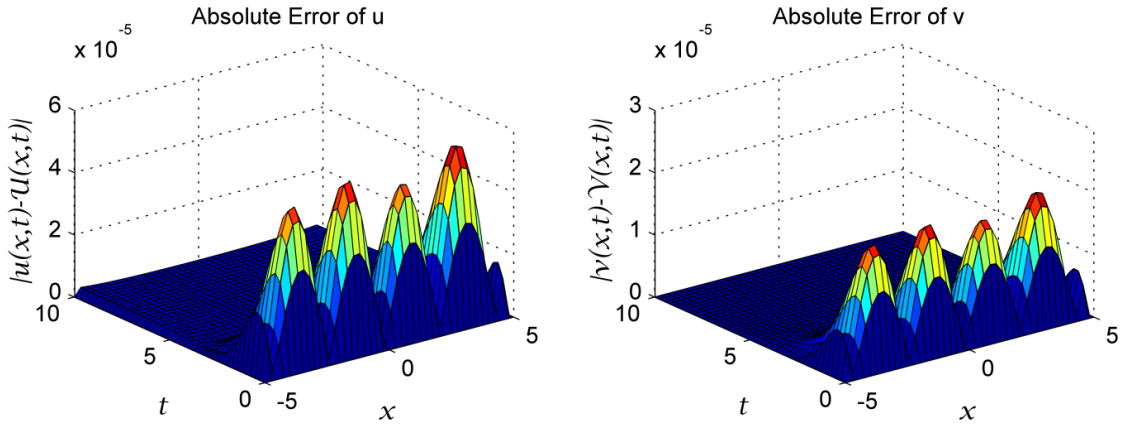

Figure 2: Illustration of the absolute error assessment between solutions (27) and (31) for system (30) using the Higher Order Compact Finite Difference (HOCFD) method.

Table 2: L_{abs} errors between the solutions (27) and (31) for system (30) using the Higher Order Compact Finite Difference (HOCFD) method at t = 2, where $a_1 = 1$,$b_1 = 1$, $a_2 = 0.1$, $b_2 = 10$, c = 0.5, and $\alpha = 1$.

x	$u(x,t)$ Exact	$U(x,t)$ HOCFD	$L_{abs}(u)$	$v(x,t)$ Exact	$V(x,t)$ HOCFD	$L_{abs}(v)$
-5	0.1001	0.0979	3.8322e-2	0.2684	0.2300	2.2221e-3
-4	0.1366	0.1361	5.8713e-2	0.4997	0.4410	4.4582e-4
-3	0.1700	0.1728	5.5515e-2	0.7744	0.7189	2.8027e-3
-2	0.1959	0.2013	3.5230e-2	1.0282	0.9930	5.3929e-3
-1	0.2135	0.2199	2.2534e-2	1.2204	1.1979	6.4757e-3
0	0.2243	0.2310	2.0088e-2	1.3474	1.3273	6.7445e-3
1	0.2306	0.2374	2.1181e-2	1.4243	1.4031	6.8095e-3
2	0.2342	0.2410	2.2415e-2	1.4686	1.4461	6.8616e-3
3	0.2361	0.2430	2.3166e-2	1.4933	1.4701	6.9131e-3
4	0.2372	0.2441	2.3664e-2	1.5069	1.4832	6.9433e-3
5	0.2378	0.2447	2.4357e-2	1.5143	1.4900	6.9243e-3

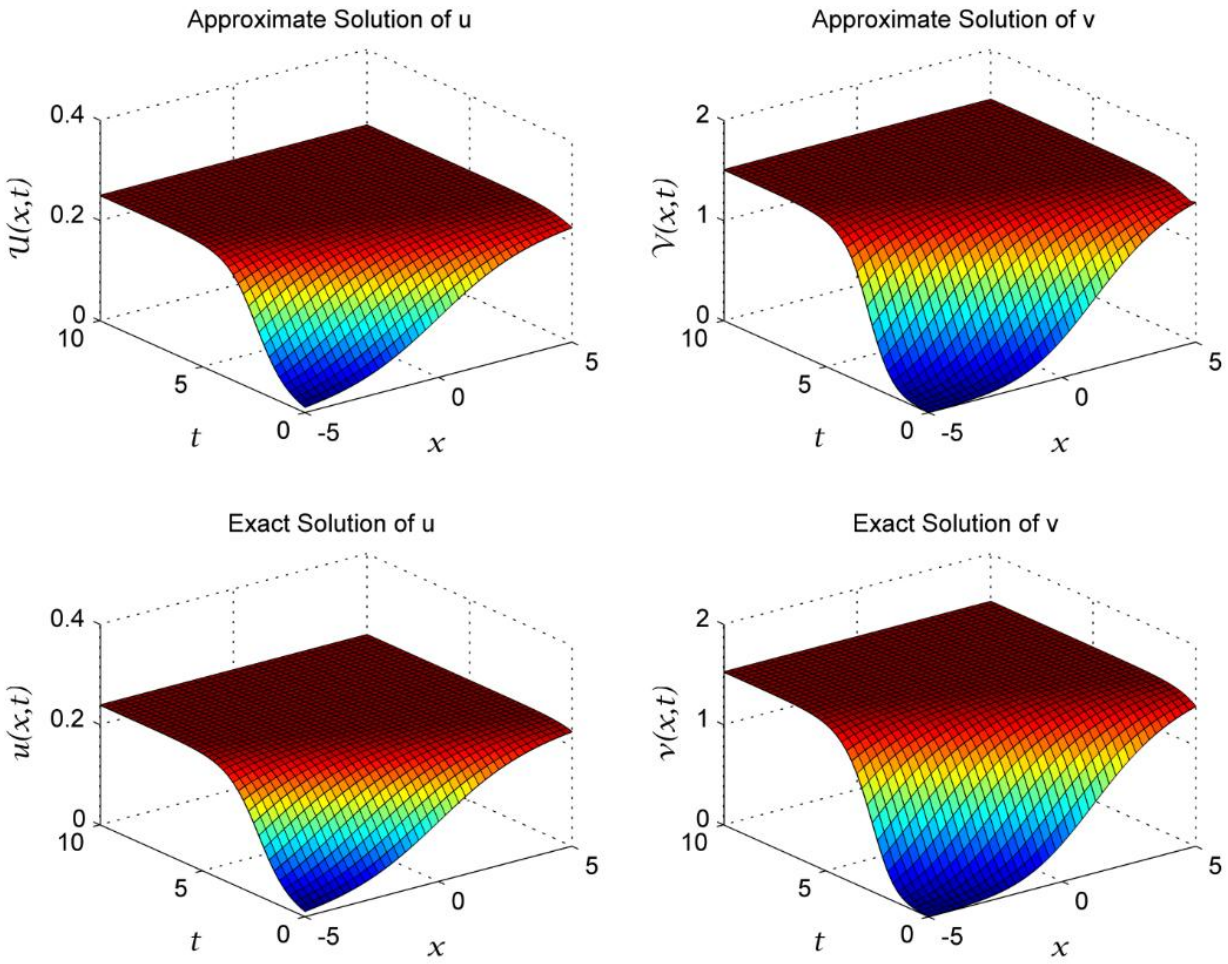

Figure 3: Representational surfaces depicting the constituents u and v within solution compositions (27) and (33) of the system (32), with $a_1 = 1, b_1 = 1$, $a_2 = 0.1$, $b_2 = 10$, c=0.5, and α=1.

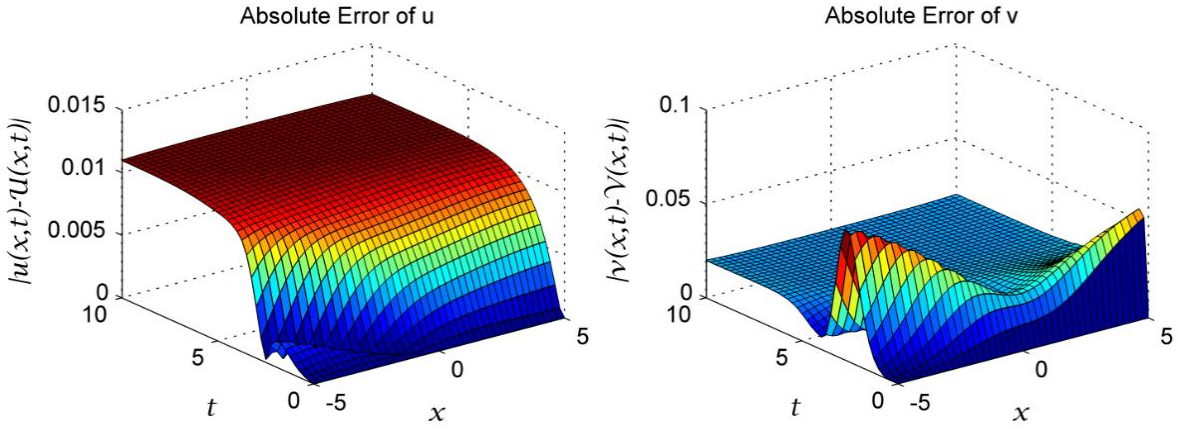

Figure 4: Illustration of the absolute error assessment between solutions (27) and (33) for system (32) using the Higher Order Compact Finite Difference (HOCFD) method.

Table 3: L_{abs} errors between the solutions (27) and (31) for system (30) using the Higher Order Compact Finite Difference (HOCFD) method at t=1, where α =3/2, β =1, $a_1 = 2$, $a_2 = 1$, $b = 3$, and $c = 5$.

x	$u(x,t)$ Exact	$U(x,t)$ HOCFD		$L_{abs}(u)$	$v(x,t)$ Exact	$V(x,t)$ HOCFD	$L_{abs}(v)$
0	5.5556	5.5556		0	0	0	0
0.2560	5.5428	5.5423		3.4262e-3	0.2294	0.2328	4.6466e-4
0.5120	5.5313	5.5306		6.4067e-3	0.4364	0.4428	6.8443e-4
0.7680	5.5222	5.5213		8.7856e-3	0.6009	0.6097	8.6077e-4
1.0240	5.5163	5.5153		1.0323e-2	0.7068	0.7171	9.7500e-4
1.2800	5.5142	5.5132		1.0860e-2	0.7438	0.7546	1.0150e-3
1.5360	5.5162	5.5152		1.0344e-2	0.7082	0.7186	9.7644e-4
1.7920	5.5220	5.5212		8.8269e-3	0.6036	0.6125	8.6347e-4
2.0480	5.5311	5.5304		6.4638e-3	0.4402	0.4466	6.8808e-4
2.3040	5.5426	5.5421		3.4938e-3	0.2338	0.2373	4.6880e-4
2.5600	5.5553	5.5553		0	0.0046	0.0046	0

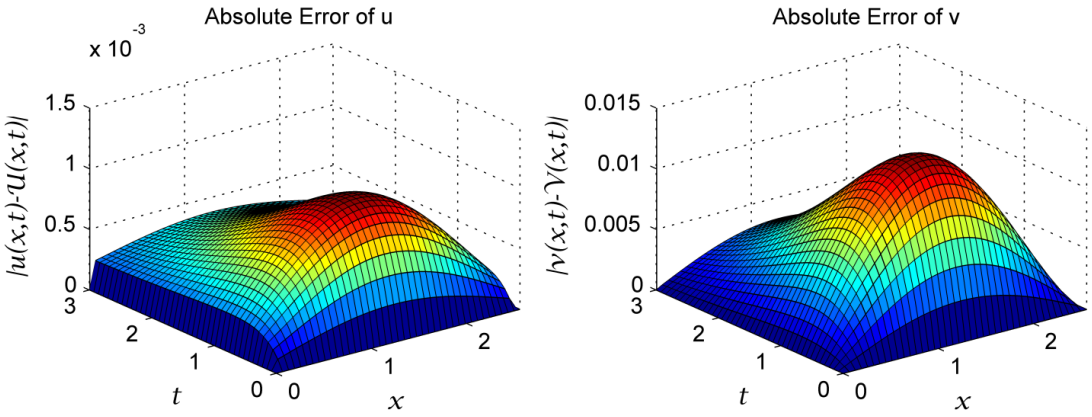

Figure 5: Illustration of the absolute error assessment between solutions (27) and (35) for system (34) using the Higher Order Compact Finite Difference (HOCFD) method.

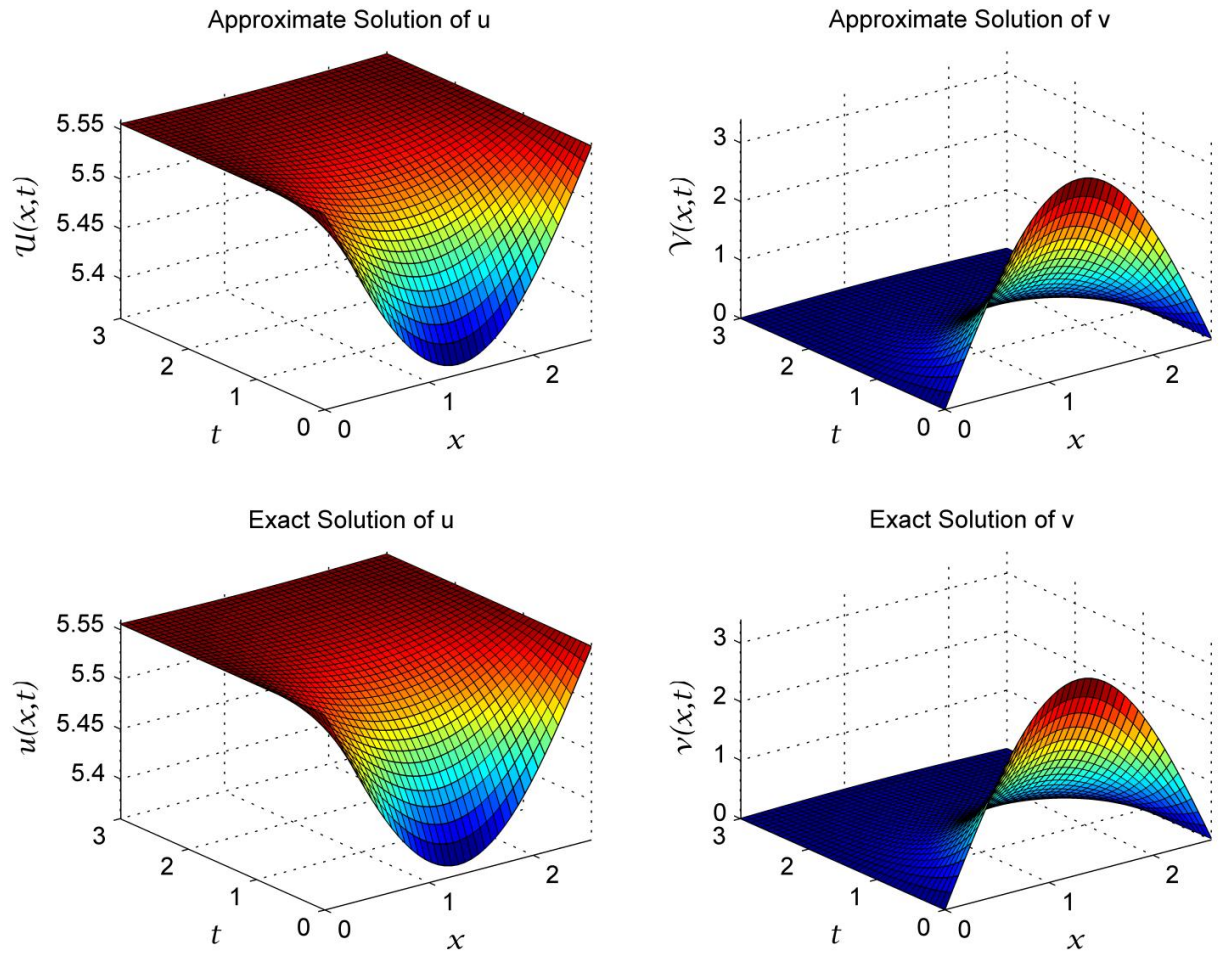

Figure 6: Representational surfaces depicting the constituents u and v within solution compositions (27) and (35) of the system (34), with α =1, β =1/3, $a_1 = 5$, $a_2 = 2$, b=0.9, c=0.05, and γ=-1.5

Conclusion

This work introduces a novel high-order compact finite difference (HOCFD) method designed to efficiently solve the challenges posed by the Reaction-Diffusion system. The central concept behind our approach is the integration of a 4th-order CFD method for spatial discretization and the utilization of an ERK scheme for time integration, resulting in a set of non-linear ordinary differential equations. Significantly, our suggested approach achieves a level of accuracy characterized by fourth-order precision in spatial computations and fifth-order precision in temporal calculations. A key objective of our technique is to minimize the number of iterations required, thus significantly reducing the scheme's computational expense. We have rigorously demonstrated that our proposed approach excels in terms of computational efficiency and stability. Extensive computational tests conducted using Matlab programming further affirm the reliability and efficiency of our method in solving Reaction-Diffusion systems. Consequently, this research contributes a valuable tool to solve the Reaction-Diffusion system, offering an effective means of achieving accurate results with reduced computational expenses.

Author Contribution

The authors contributed equally to this work, from the implementation and design of the research, the analysis of the results and to the writing of the manuscript.

Funding

The authors received no financial support for the research, authorship, and publication of this article.

Conflict of interests

None

References

1. Cao, H.H., Liu, L.B., Zhang, Y. and Fu, S.M., 2011. A fourth-order method of the convection–diffusion equations with Neumann boundary conditions. Applied Mathematics and Computation, 217(22), pp.9133-9141. https://doi.org/10.1016/j.amc.2011.03.141
2. Yao, C., Zhang, Y., Chen, J., Ling, X., Jing, K., Lu, Y. and Fan, E., 2020. Development of a fourth-order compact finite difference scheme for simulation of simulated-moving-bed process. Scientific reports, 10(1), p.7820. https://doi.org/10.1038/s41598-020-64562-8
3. Fu, Y., Tian, Z. and Liu, Y., 2018. A Compact Exponential Scheme for Solving 1D Unsteady Convection-Diffusion Equation with Neumann Boundary Conditions. arXiv preprint arXiv:1805.05728. https://doi.org/10.48550/arXiv.1805.05728

4. Nyaanga, P.P., 2004. The interaction of species in population biology. www.ii.uib.no/~pius/thesis3.pdf.

5. Manaa, S.A. and Qasem, A.F., 2007. Numerical solution of non-linear prey-predator system using finite elements method. AL-Rafidain Journal of Computer Sciences and Mathematics, 4(2), pp.113-133.

6. Turing, A.M. 1952. The chemical basis of morphogenesis. Phil. Trans. R. Soc. Lond. B 237, pp.37–72.

7. Rasool, H.M., Pirdawood, M.A., Sabawi, Y.A., Mahmood, R.H. and Khalil, P.A., 2022. Model reduction and analysis for ERK cell signalling pathway using implicit-explicit Rung-Kutta methods. Passer Journal of Basic and Applied Sciences, 4(Special issue), pp.160-178. https://doi.org/10.24271/psr.2022.161692

8. Pirdawood, M.A., Rasool, H.M. and Aziz, B.F., 2023. Integrating Sensitivity Analysis and Explicit Runge-Kutta Method for Modeling the Effect of Exposure Rate to Contaminated Water on Cholera Disease Spread. Passer Journal of Basic and Applied Sciences, 6(1), pp.33-41.

9. Murray, J.D., 1989. Mathematical biology, vol. 19 of Biomathematics. Springer, Berlin.

10. Shampine, L.F., 1975. Computer solution of ordinary differential equations. The initial value problem.

11. Cherniha, R., 2011. Conditional symmetries and exact solutions of the diffusive Lotka–Volterra system. Mathematical and computer modelling, 54(5-6), pp.1238-1251. https://doi.org/10.1016/j.mcm.2011.03.035