Hindawi
International Journal of Mathematics and Mathematical Sciences
Volume 2024, Article ID 5541066, 12 pages
https://doi.org/10.1155/2024/5541066

Research Article

Numerical Solution of Two-Dimensional Nonlinear Unsteady Advection-Diffusion-Reaction Equations with Variable Coefficients

Endalew Getnet Tsega

Department of Mathematics, College of Science, Bahir Dar University, Bahir Dar, Ethiopia

Correspondence should be addressed to Endalew Getnet Tsega; endalebdumath2016@gmail.com

Received 29 January 2024; Revised 12 March 2024; Accepted 21 March 2024; Published 31 March 2024

Academic Editor: Saranya Shekar

The advection-diffusion-reaction (ADR) equation is a fundamental mathematical model used to describe various processes in many different areas of science and engineering. Due to wide applicability of the ADR equation, finding accurate solution is very important to better understand a physical phenomenon represented by the equation. In this study, a numerical scheme for solving two-dimensional unsteady ADR equations with spatially varying velocity and diffusion coefficients is presented. The equations include nonlinear reaction terms. To discretize the ADR equations, the Crank–Nicolson finite difference method is employed with a uniform grid. The resulting nonlinear system of equations is solved using Newton's method. At each iteration of Newton's method, the Gauss–Seidel iterative method with sparse matrix computation is utilized to solve the block tridiagonal system and obtain the error correction vector. The consistency and stability of the numerical scheme are investigated. MATLAB codes are developed to implement this combined numerical approach. The validation of the scheme is verified by solving a two-dimensional advection-diffusion equation without reaction term. Numerical tests are provided to show the good performances of the proposed numerical scheme in simulation of ADR problems. The numerical scheme gives accurate results. The obtained numerical solutions are presented graphically. The result of this study may provide insights to apply numerical methods in solving comprehensive models of physical phenomena that capture the underlying situations.

1. Introduction

The ADR equation is a partial differential equation used to represent various mathematical models in science and engineering. The equation describes transport processes driven by advection, diffusion, and reaction. Advection involves the collective displacement of species propelled by a flow field. Diffusion, characterized by the movement of species driven by concentration gradients, serves to homogenize concentration distributions over time. Reaction process occurring within the system may either generate or deplete the transported species. In practical scenarios, the coupling of advection, diffusion, and reaction processes results in intricate interactions, leading to temporal and spatial alterations within the ADR system. The ADR equation is used in modeling contaminant transport [1, 2], fluid flow [3], mass and heat transfer [4], chemical reaction process [5], diffusion across a cell membrane [6], tumor cell expansion [7], population dynamics [8], and so on.

Because of its importance in many fields, obtaining accurate solution for ADR equation has been the interest of numerous researchers. There have been a large number of solutions for ADR equations presented in literature using analytic [1, 9, 10] and numerical [11–15] techniques. Due to the intricate nature of many advection-diffusion-reaction problems, numerical techniques are commonly employed to obtain solutions for the ADR equation. Various numerical techniques have been employed in the literature to compute numerical solutions for two-dimensional nonlinear ADR equations. For instance, Mesgarani et al. [16] used radial basis functions to solve time-dependent nonlinear ADR equations with variable coefficients. They transformed the

ADR equation into system ordinary differential equations and used the fourth-order Runge–Kutta method to compute solution of the system. Similarly, Ali et al. [17] applied a finite difference method incorporating with Lucas and Fibonacci polynomials to obtain the solution of one- and two-dimensional nonlinear ADR equations. Ngondiep [18] employed a six-level time-split Leap-Frog/Crank–Nicolson approach to approximate solutions of two-dimensional nonlinear time-dependent ADR equations. He deduced that the numerical scheme is fast, spatial fourth-order convergent, and temporal second-order accurate. Zhang et al. [19] used fourth-order fully implicit finite difference scheme to solve unsteady nonlinear ADR equations. Tambue [20] presented a finite volume method combined with exponential time differencing to solve the ADR equation involving the nonlinear reaction term.

Numerous researchers have solved ADR equations that model real-realistic problems. Yu et al. [21] utilized the homotopy analysis method (HAM) to simulate a contaminant transport model governed by ADR equations, presenting contaminant concentration profiles at various time points based on simulation outcomes. Para et al. [12] investigated a water pollution scenario represented by ADR equations, employing a finite volume method. Their numerical solution was compared with results obtained using a finite difference method, revealing satisfactory agreement between the solutions. Gurarslan et al. [22] obtained a numerical solution for contaminant transport described by a one-dimensional advection-diffusion equation, employing a finite difference scheme in space and a Runge–Kutta method in time. The integrated scheme yielded accurate solutions. Putri et al. [23] conducted numerical simulations of advection-diffusion equations representing oxygen demand concentration in waste stabilization ponds using a finite difference method. Pananu et al. [24] utilized a finite difference method scheme to investigate a water pollutant dispersion problem.

A Crank–Nicolson method is an implicit finite difference method for solving partial differential equations involving time and space derivatives [25]. In solving partial differential equations using the Crank–Nicolson method, both the time and space derivatives are approximated by central differences. The method is unconditionally stable and second-order accurate both in space and time variables for solving the diffusion equation. However, the method requires more computing time to solve the resulting system of algebraic equations. Specially, when the method is applied to solve nonlinear and multidimensional problems, linearization techniques like Newton's method [26] are required. Solving the system of nonlinear equations introduced through discretization demands much extra time. Any method solving system of linear equations (matrix-inversion method, Gauss–Seidel method, Lu decomposition method, and so on) can be used to solve the block tridiagonal system to obtain the error correction vector required in Newton's method.

The objective of this paper is to devise a numerical scheme based on finite difference approximation to solve ADR equations with all of the following properties: (i) unsteady, (ii) two-dimensional, (iii) nonlinear reaction term, and (iv) spatially varying velocity and diffusion coefficients. The intention is to employ the developed numerical scheme to simulate the distribution of concentration of a contaminant within a flowing fluid.

2. Problem Description

In this study, the ADR equation of the following form is considered [27]:

$$\frac{\partial}{\partial t}C(x,y,t) + \nabla\cdot(\mathbf{V}(x,y)C(x,y,t)) = \nabla\cdot(\mathbf{D}(x,y)\nabla C(x,y,t)) + R(C(x,y,t)),\quad (x,y)\in\Omega\subseteq\mathbb{R}^2, 0<t\leq T, \tag{1}$$

with initial condition

$$C(x,y,0) = C_0(x,y),\quad (x,y)\in\Omega, \tag{2}$$

and boundary condition

$$C(x,y,t) = g(x,y,t),\quad (x,y)\in\partial\Omega, t>0, \tag{3}$$

where $C(x,y,t)$ is contaminant concentration or temperature at position *(x, y)* and time t, $\mathbf{V}(x,y)$ is the advection velocity vector, $\mathbf{D}(x,y)$ is a tensor expressing the diffusivity of C, $R(C(x,y,t))$ is the reaction term assumed to be nonlinear function of C making the ADR equation nonlinear, and $\partial\Omega$ is the boundary of a bounded domain Ω in $\mathbb{R}^2$. We assume contaminant is released in an incompressible fluid so that the continuity equation $\nabla\cdot\mathbf{V} = u_x + u_y = 0$ is satisfied. If we consider $\mathbf{V}(x,y) = (u(x,y), v(x,y))$ and $\mathbf{D}(x,y) = \text{diag}(Dx(x,y), Dy(x,y))$, where u, v, Dx, and Dy are velocity components and diffusion coefficients in x- and y-directions, the ADR equation in (1) can be rewritten as

$$\frac{\partial C}{\partial t} + \left(u - \frac{\partial Dx}{\partial x}\right)\frac{\partial C}{\partial x} + \left(v - \frac{\partial Dy}{\partial y}\right)\frac{\partial C}{\partial y} - Dx\frac{\partial^2 C}{\partial x^2} - Dy\frac{\partial^2 C}{\partial y^2} - R(C) = 0. \tag{4}$$

For this work, the spatial domain $\Omega = [0, 1] \times [0, 1]$ and time domain $[0, T]$ are used.

3. Discretization of ADR Equation

To apply finite difference discretization of (4), we divide the spatial domain into uniform meshes of step sizes Δx and Δy in x- and y-directions, as shown in Figure 1, and the time domain with step size Δt. The grid points are defined as

$$\begin{aligned} x_i &= (i-1)\Delta x, \quad i = 1, 2, 3, \ldots, N_x, \Delta x = \frac{1-0}{N_x - 1}, \\ y_j &= (j-1)\Delta y, \quad j = 1, 2, 3, \ldots, N_y, \Delta y = \frac{1-0}{N_y - 1}, \\ t_k &= (k-1)\Delta t, \quad k = 1, 2, 3, \ldots, N_t, \Delta t = \frac{T-0}{N_t - 1}. \end{aligned} \tag{5}$$

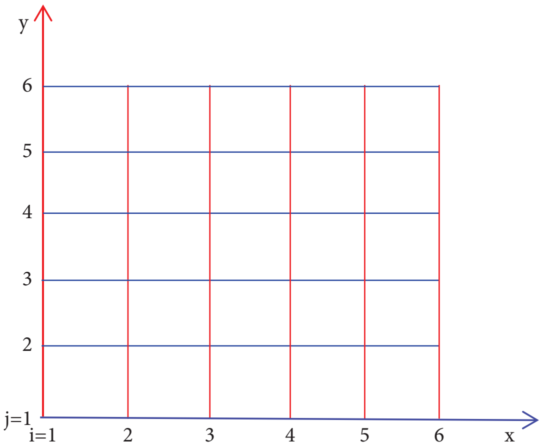


Figure 1: 2D computational mesh.

In equation (4), approximating the time derivative by central difference at point $(i, j, k + (1/2))$, the derivatives of the diffusion coefficients by central difference, the derivatives of C by the averages of central difference at k^{th} and $(k+1)^{\text{th}}$ time levels, and reaction term using the average values of R at k^{th} and $(k+1)^{\text{th}}$ time levels, we get

$$\begin{aligned} &\frac{C^{k+1}_{i,j} - C^{k+1}_{i,j}}{dt} + \left[u_{i,j} - \frac{Dx_{i+1,j} - Dx_{i-1,j}}{2\Delta x}\right]\left[\frac{1}{2}\left(\frac{C^{k}_{i+1,j} - C^{k}_{i-1,j}}{2\Delta x} + \frac{C^{k+1}_{i+1,j} - C^{k+1}_{i-1,j}}{2\Delta x}\right)\right] \\ &+ \left[v_{i,j} - \frac{Dy_{i,j+1} - Dx_{i,j-1}}{2\Delta y}\right]\left[\frac{1}{2}\left(\frac{C^{k}_{i,j+1} - C^{k}_{i,j-1}}{2\Delta y} + \frac{C^{k+1}_{i,j+1} - C^{k+1}_{i,j-1}}{2\Delta y}\right)\right] \\ &- Dx_{i,j}\left[\frac{C^{k}_{i+1,j} - 2C^{k}_{i,j} + C^{k}_{i-1,j}}{\Delta x^2} + \frac{C^{k+1}_{i+1,j} - 2C^{k+1}_{i,j} + C^{k+1}_{i-1,j}}{\Delta x^2}\right] \\ &- Dy_{i,j}\left[\frac{C^{k}_{i,j+1} - 2C^{k}_{i,j} + C^{k}_{i,j-1}}{\Delta y^2} + \frac{C^{k+1}_{i,j+1} - 2C^{k+1}_{i,j} + C^{k+1}_{i,j-1}}{\Delta y^2}\right] - \frac{1}{2}\left(R\left(C^{k}_{i,j}\right) + R\left(C^{k+1}_{i,j}\right)\right) = 0. \end{aligned} \tag{6}$$

Note that in discretization (6), the Crank–Nicolson Method is used. Rearranging equation (6), we obtain

$$\begin{aligned} &\left(-P_{i,j} - S_{i,j}\right)C^{k+1}_{i-1,j} + \left(2 + S_{i,j} + T_{i,j}\right)C^{k+1}_{i,j} - R\left(C^{k+1}_{i,j}\right) + \left(P_{i,j} - S_{i,j}\right)C^{k+1}_{i+1,j} \\ &\left(-Q_{i,j} - T_{i,j}\right)C^{k+1}_{i,j-1} + \left(Q_{i,j} - T_{i,j}\right)C^{k+1}_{i,j+1} + \left(-P_{i,j} - S_{i,j}\right)C^{k}_{i-1,j} + \left(-2 + S_{i,j} + T_{i,j}\right)C^{k}_{i,j} \\ &- R\left(C^{k}_{i,j}\right) + \left(P_{i,j} - S_{i,j}\right)C^{k}_{i+1,j} + \left(-Q_{i,j} - T_{i,j}\right)C^{k}_{i,j-1} + \left(Q_{i,j} - T_{i,j}\right)C^{k}_{i,j+1} = 0, \end{aligned} \tag{7}$$

where $P_{i,j} = (dt/2\Delta x)(u_{i,j} - (Dx_{i+1,j} - Dx_{i-1,j})/(2\Delta x)), Q_{i,j} = (dt/2\Delta y)(v_{i,j} - (Dy_{i,j+1} - Dy_{i,j-1})/(2\Delta y))$, and

$$S_{i,j} = \frac{dtDx_{i,j}}{\Delta x^2}, T_{i,j} = \frac{dtDy_{i,j}}{\Delta y^2}. \tag{8}$$

For $j = 2$ and $i = 2, 3, \ldots, N_x - 2$, $j = 3$ and $i = 2, 3, \ldots, N_x - 2, \ldots, j = N_y - 2$ and $i = 2, 3, \ldots, N_x - 2$, equation (7) yields a system of nonlinear equations with $(N_x - 2) \times (N_y - 2)$ unknowns $C_{2,2}^{k+1}, C_{3,2}^{k+1}, \ldots, C_{N_x-1,2}^{k+1}, C_{3,2}^{k+1}, C_{3,3}^{k+1}, \ldots, C_{N_x-1,3}^{k+1}, \ldots, C_{N_x-1,2}^{k+1}, C_{N_x-1,3}^{k+1}, \ldots, C_{N_x-1,N_x-1}^{k+1}$.

4. Newton's Method

For $p = 2, 3, \ldots, (N_x - 2) \times (N_y - 2)$, define equations

$$f_p\left(C_{i-1,j}^{k+1}, C_{i,j}^{k+1}, C_{i+1,j}^{k+1}, C_{i,j-1}^{k+1}, C_{i,j+1}^{k+1}\right) = 0, \qquad (9)$$

where the functions f_p include variable terms from $(k+1)^{\text{th}}$ time level values and reaction terms and constant terms from the boundary values and k^{th} time level values. Let $\mathbf{f} = (f_1, f_2, f_3, \ldots, f_{(N_x-2)\times(N_y-2)})^T$ be a vector function whose components are the functions in (9). Let $X = (C_{2,2}^{k+1}, C_{3,2}^{k+1}, \ldots, C_{N_x-1,2}^{k+1}, C_{3,2}^{k+1}, C_{3,3}^{k+1}, \ldots, C_{N_x-1,3}^{k+1}, \ldots, C_{N_x-1,2}^{k+1}, C_{N_x-1,3}^{k+1}, \ldots, C_{N_x-1,N_x-1}^{k+1})^{\mathrm{T}}$ be the vector of unknowns in (7). Let X_0 be the initial approximation of X. Using Newton's method, an error correction vector $h = (h_1, h_2, h_3, \ldots, h_{(N_x-2)\times(N_x-2)})^T$ such that $X = X_0 + h$ can be obtained iteratively by solving the transformed linear system:

$$Ah = -\mathbf{f}(X_0), \qquad (10)$$

where $A_{i,j} = (\partial f_i(X_0)/\partial X_j)$. The iterative process for solving h in (10) at each $(k+1)^{\text{th}}$ time level continues until we obtain X at this time level with the required accuracy. The computation is performed until $\mathbf{f}(X) < \text{Tol}$, where Tol is the tolerance of approximation for Newton's method. The matrix A in (10) is a block tridiagonal matrix. For instance, for sample mesh indicated in Figure 1 with 6×6 grid points, A has the following form:

$$A = \left[\begin{array}{cccc|cccc|cccc|cccc}
d_1 & q_1 & 0 & 0 & t_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
p_1 & d_2 & q_2 & 0 & 0 & t_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & p_2 & d_3 & q_3 & 0 & 0 & t_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & p_3 & d_4 & 0 & 0 & 0 & t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
s_1 & 0 & 0 & 0 & d_5 & q_4 & 0 & 0 & t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & s_2 & 0 & 0 & p_4 & d_6 & q_5 & 0 & 0 & t_6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & s_3 & 0 & 0 & p_5 & d_7 & q_6 & 0 & 0 & t_7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & s_4 & 0 & 0 & p_6 & d_8 & 0 & 0 & 0 & t_8 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & s_5 & 0 & 0 & 0 & d_9 & q_7 & 0 & 0 & t_9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & s_6 & 0 & 0 & p_7 & d_{10} & q_8 & 0 & 0 & t_{10} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & s_7 & 0 & 0 & p_8 & d_{11} & q_9 & 0 & 0 & t_{11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & s_8 & 0 & 0 & p_9 & d_{12} & 0 & 0 & 0 & t_{12} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_9 & 0 & 0 & 0 & d_{13} & q_{10} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{10} & 0 & 0 & p_{10} & d_{14} & q_{11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{11} & 0 & 0 & p_{11} & d_{15} & q_{12} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{12} & 0 & 0 & p_{12} & d_{16}
\end{array}\right]$$

To avoid computation with zero entries in matrix A, the Gauss–Seidel iterative method with sparse matrix computation is used to solve the system in (10). A termination criterion $\|h^{\text{iter}+1} - h^{\text{iter}}\|_\infty < \text{tol}$ is used in the Gauss–Seidel iterative method, where *tol* is a prescribed number to bound the error. The values of X at $k+2, k+3, \ldots, N_t$ time levels are solved similarly.

5. Convergence and Accuracy of the Numerical Scheme

Let L and L_D be the differential operator and the finite difference operator representing equations (4) and (6), respectively. If the diffusion coefficient is constant, the truncation error *T.E.* of the finite difference scheme in (6) is

$$
\begin{aligned}
\mathrm{TE} &= L(C) - L_D\left(C_{i,j}^{k}\right) \\
&= -\left(\frac{\Delta t^2}{24}\frac{\partial^3 C(x,y,t+(\Delta t/2))}{\partial t^3} + \frac{\Delta t^4}{1920}\frac{\partial^5 C(x,y,t+(\Delta t/2))}{\partial t^5} + \frac{\Delta t^6}{322560}\frac{\partial^7 C(x,y,t+(\Delta t/2))}{\partial t^7} + \cdots\right) \\
&\quad - \frac{u(x,y)}{2}\left(\frac{\Delta x^2}{6}\frac{\partial^3 C(x,y,t)}{\partial x^3} + \frac{\Delta x^4}{120}\frac{\partial^5 C(x,y,t)}{\partial x^5} + \frac{\Delta x^6}{5040}\frac{\partial^7 C(x,y,t)}{\partial x^7} + \cdots\right) \\
&\quad - \frac{u(x,y)}{2}\left(\frac{\Delta x^2}{6}\frac{\partial^3 C(x,y,t+\Delta t)}{\partial x^3} + \frac{\Delta x^4}{120}\frac{\partial^5 C(x,y,t+\Delta t)}{\partial x^5} + \frac{\Delta x^6}{5040}\frac{\partial^7 C(x,y,t+\Delta t)}{\partial x^7} + \cdots\right) \\
&\quad - \frac{v(x,y)}{2}\left(\frac{\Delta y^2}{6}\frac{\partial^3 C(x,y,t)}{\partial y^3} + \frac{\Delta y^4}{120}\frac{\partial^5 C(x,y,t)}{\partial y^5} + \frac{\Delta y^6}{5040}\frac{\partial^7 C(x,y,t)}{\partial y^7} + \cdots\right) \\
&\quad - \frac{v(x,y)}{2}\left(\frac{\Delta y^2}{6}\frac{\partial^3 C(x,y,t+\Delta t)}{\partial y^3} + \frac{\Delta y^4}{120}\frac{\partial^5 C(x,y,t+\Delta t)}{\partial y^5} + \frac{\Delta y^6}{5040}\frac{\partial^7 C(x,y,t+\Delta t)}{\partial y^7} + \cdots\right) \\
&\quad + \frac{Dx}{2}\left(\frac{\Delta x^2}{12}\frac{\partial^4 C(x,y,t)}{\partial x^4} + \frac{\Delta x^4}{360}\frac{\partial^6 C(x,y,t)}{\partial x^6} + \frac{\Delta x^6}{20160}\frac{\partial^8 C(x,y,t)}{\partial x^8} + \cdots\right) \\
&\quad + \frac{Dx}{2}\left(\frac{\Delta x^2}{12}\frac{\partial^4 C(x,y,t+\Delta t)}{\partial x^4} + \frac{\Delta x^4}{360}\frac{\partial^6 C(x,y,t+\Delta t)}{\partial x^6} + \frac{\Delta x^6}{20160}\frac{\partial^8 C(x,y,t+\Delta t)}{\partial x^8} + \cdots\right) \\
&\quad + \frac{Dy}{2}\left(\frac{\Delta y^2}{12}\frac{\partial^4 C(x,y,t)}{\partial y^4} + \frac{\Delta y^4}{360}\frac{\partial^6 C(x,y,t)}{\partial y^6} + \frac{\Delta y^6}{20160}\frac{\partial^8 C(x,y,t)}{\partial y^8} + \cdots\right) \\
&\quad + \frac{Dy}{2}\left(\frac{\Delta y^2}{12}\frac{\partial^4 C(x,y,t+\Delta t)}{\partial y^4} + \frac{\Delta y^4}{360}\frac{\partial^6 C(x,y,t+\Delta t)}{\partial y^6} + \frac{\Delta y^6}{20160}\frac{\partial^8 C(x,y,t+\Delta t)}{\partial y^8} + \cdots\right) \\
&= O\left(\Delta t^2\right) + O\left(\Delta x^2\right) + O\left(\Delta y^2\right).
\end{aligned}
\tag{11}
$$

The truncation error vanishes as $(\Delta t, \Delta x, \Delta y) \longrightarrow (0,0,0)$. Hence, the finite difference scheme is consistent with the partial differential equation.

Let us analyse the stability of the finite difference scheme (6) using the Von Neumann analysis method for some cases [25]. Assume velocity and diffusion coefficients are constants and the reaction term is zero. Suppose the discretization of C in (6) at grid point (i, j, k) has the form [13]

$$C_{i,j}^{k} = A\xi^{k} e^{I\beta_1 i\Delta x} e^{I\beta_2 j\Delta x}, \tag{12}$$

where $I = \sqrt{-1}$, A, β_1, and β_2 are a constants, and ξ is the amplification factor. Substituting (12) in (6) and dividing both sides of equations by $A\xi^{k} e^{I\beta_1 i\Delta x} e^{I\beta_2 j\Delta x}$, we get

$$
\begin{aligned}
&\frac{\xi - 1}{\Delta t} + \frac{1}{2}u\left[\frac{e^{I\beta_1\Delta x} - e^{-I\beta_1\Delta x}}{2\Delta x} + \frac{\xi e^{I\beta_1\Delta x} - \xi e^{-I\beta_1\Delta x}}{2\Delta x}\right] + \frac{1}{2}v\left[\frac{e^{I\beta_2\Delta y} - e^{-I\beta_2\Delta y}}{2\Delta y} + \frac{\xi e^{I\beta_2\Delta y} - \xi e^{-I\beta_2\Delta y}}{2\Delta y}\right] \\
&\quad - \frac{1}{2}Dx\left[\frac{e^{I\beta_1\Delta x} - 2 + e^{-I\beta_1\Delta x}}{\Delta x^2} + \frac{\xi e^{I\beta_1 i\Delta x} - 2\xi + \xi e^{-I\beta_1\Delta x}}{\Delta x^2}\right] \\
&\quad - \frac{1}{2}Dy\left[\frac{e^{I\beta_2\Delta y} - 2 + e^{-I\beta_2\Delta y}}{\Delta y^2} + \frac{\xi e^{I\beta_2\Delta y} - 2\xi + \xi e^{-I\beta_2\Delta y}}{\Delta y^2}\right] = 0.
\end{aligned}
\tag{13}
$$

Using the facts that $e^{I\theta} = \cos\theta + \mathrm{I}\sin\theta$ and $e^{-I\theta} = \cos\theta - \mathrm{I}\sin\theta$ and solving for ξ from (13), we have

$$\xi = \frac{1-R}{1+R}, \tag{14}$$

where

$$R = \frac{2\alpha\Delta t\sin^2(\beta_1\Delta x/2)}{\Delta x^2} + \frac{2\alpha\Delta t\sin^2(\beta_2\Delta y/2)}{\Delta y^2} + \frac{b_1\Delta tI\sin(\beta_1\Delta x)}{2\Delta x} + \frac{b_2\Delta tI\sin(\beta_2\Delta y)}{2\Delta y}. \tag{15}$$

For the stability, we must have $|\xi| \le 1$. From (14), $|\xi| = |(1-R)/(1+R)| \le 1$.

Hence, the finite difference scheme (6) is unconditionally stable. According to Lax's equivalence theorem, the finite difference scheme (6) gives a convergent solution for the considered cases.

In the proposed numerical scheme, the linearized system of equations at each iteration is solved using the Gauss–Seidel Iterative method to obtain error correction vector. Thus, if A is diagonally dominant, the Gauss–Seidel iterative method is convergent. A is diagonally dominant if the following inequality is satisfied for each row in the matrix:

$$\left|2 + 2S_{i,j} + 2T_{i,j} - \frac{\partial R(C)}{\partial C}\bigg|_{x=X_0(m)}\right| > \left|-P_{i,j} - S_{i,j}\right| + \left|P_{i,j} - S_{i,j}\right| + \left|-Q_{i,j} - T_{i,j}\right| + \left|Q_{i,j} - T_{i,j}\right|, \tag{16}$$

where X_0 is the initial guess to be taken for $(k+1)^{\text{th}}$ iteration and m indicates the m^{th} component of X_0. In (16), the following indexes have to be used: $2 \le i \le N_x - 1$ for all terms in the inequality; $2 \le j \le N_y - 1$ for the term in the left side and for the first and second terms in the right side; $2 \le j \le N_y - 2$ for the third term; and $3 \le j \le N_y - 1$ for the fourth term in the right side.

The accuracy of the numerical scheme is calculated by the absolute maximum error formula given by [28]

$$E = \max_{1\le i\le N_x, 1\le j\le N_y, 1\le k\le N_t}\left|C_{i,j}^k - C\left(x_i, y_j, t_k\right)\right|, \tag{17}$$

where $C_{i,j}^k$ and $C(x_i, y_j, t_k)$ are the approximate and exact solutions, respectively. The rate of convergence for the scheme is calculated using the formula as follows:

$$r = \frac{\ln(E_1/E_2)}{\ln(2)}. \tag{18}$$

Here, E_1 and E_2 are maximum absolute errors with sets of number of grid points $S_1 = \{N_x, Ny, N_t\}$ and $S_2 = \{2N_x, 2Ny, 2N_t\}$, respectively, where the set S_2 is obtained using half of the spatial and temporal step sizes used in S_1. If the exact solution is not known in (17), the absolute maximum error can be estimated using two approximate solutions as

$$E = \max_{1\le i\le N_x, 1\le j\le N_y, 1\le k\le N_t}\left|\left(C_{i,j}^k\right)^{S_1} - \left(C_{2i-1,2j-1}^{2k-1}\right)^{S_2}\right|, \tag{19}$$

and the rate of convergence can be computed using (19) and (18) accordingly.

6. Validation

To validate the convergent, accuracy, and computational efficiency of the proposed numerical scheme, the advection-diffusion equation in [29] with constant diffusion coefficient $Dx = Dy = 1$ and velocity components $u = \cos\pi y$ and $v = -\cos\pi x$ is taken. The equation is

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} - \frac{\partial^2 C}{\partial x^2} - \frac{\partial^2 C}{\partial y^2} = 0, \quad 0 < t \le T, \Omega = [0,1]\times[0,1], \tag{20}$$

with initial condition

$$C(x, y, 0) = \sin(\pi x) + \sin(\pi x), \tag{21}$$

and boundary conditions

$$\begin{aligned} C(x, 0, t) &= C(x, 1, t) = \sin(\pi x)e^{-\pi^2 t}, \\ C(0, y, t) &= C(1, y, t) = \sin(\pi y)e^{-\pi^2 t}. \end{aligned} \tag{22}$$

An approximate solution of (20) is obtained by the numerical scheme with spatial step sizes $\Delta x = \Delta y = 0.02$ and time step size $\Delta t = 0.005$ and $t = 0.5$. A tolerance of 0.000001 is employed for both Newton's method and the Gauss–Seidel iterative method. Figure 2 illustrates the comparison between the approximate and exact solutions. In this computation, the absolute maximum error of the results obtained using the numerical scheme for $0 \le x, y \le 1$ and $0 \le t \le 0.5$ is 7.5275×10^{-5}. Furthermore, Figure 3 presents a comparison of the exact and approximate solutions for $0 \le t \le 1$ at $x = 0.5$ and $y = 0.5$ and for $0 \le x \le 1$ at $y = 0.5$ and $t = 0.5$. Notably, there is a strong agreement between the numerical and exact solutions, as evidenced by the figures. In addition, Table 1 shows the maximum absolute errors, convergence rates, and CPU time. The validation process demonstrates that the proposed numerical scheme yields accurate results and can be utilized for addressing similar partial differential equations. The scheme requires increased computational time when employing very small spatial step sizes.

7. Numerical Examples and Discussion

In this section, numerical demonstrations are provided to illustrate the practical utilization of the proposed numerical scheme in solving ADR problems. The computed results are illustrated in graphs, and the estimated maximum absolute errors are presented in tabular form. For all examples,

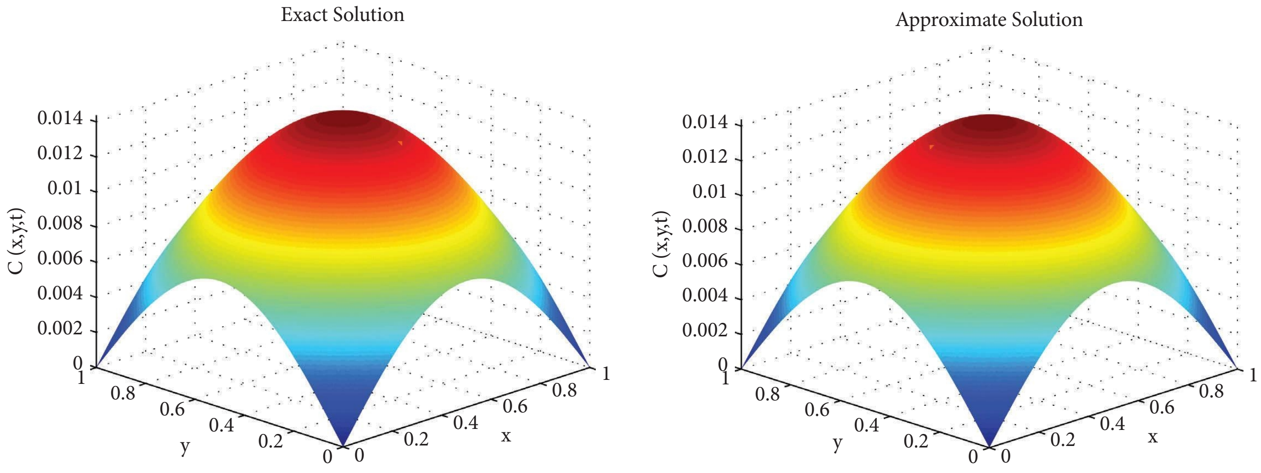
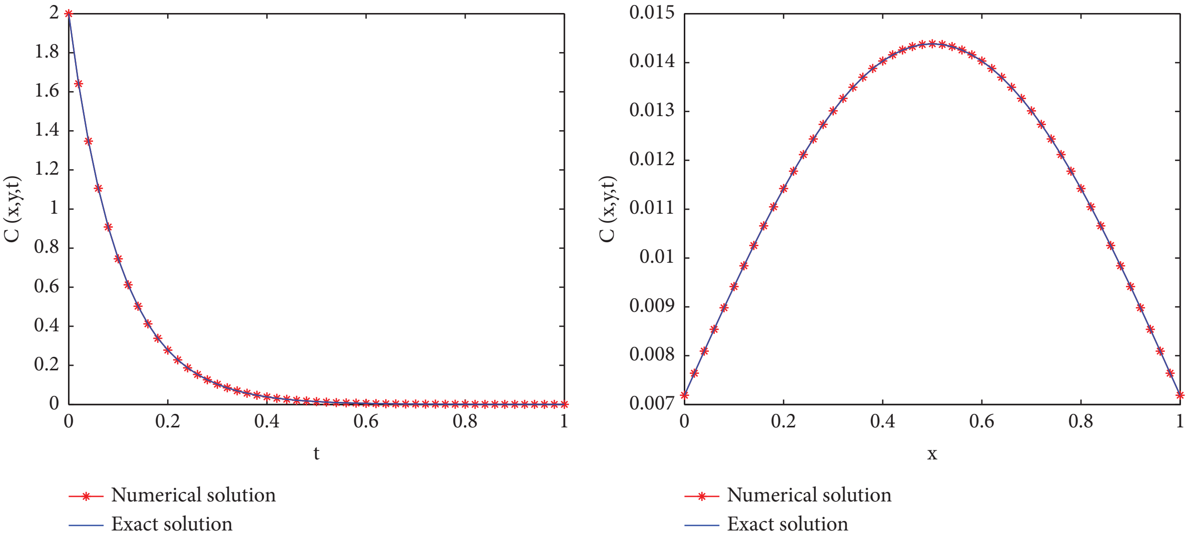


Figure 2: Space-time graph of exact and approximate solutions of (16) with $\Delta x = \Delta y = 0.02$ and $\Delta t = 0.005$ at $t = 0.5$.



(a)

(b)

Figure 3: Comparison of numerical and exact solutions of (16) for (a) $0 \le t \le 1$ at $x = 0.5$ and $y = 0.5$ and (b) $0 \le x \le 1$ at $y = 0.5$ and $t = 0.5$.

Table 1: The maximum absolute errors, convergence rate, and CPU time for solving (16) ($0 \le x, y \le 1$, $0 \le t \le 1$).

$\Delta x, \Delta y$	N_x, N_y	Δt	N_t	E	r	CPU time (in sec)
0.1	11	0.02	51	0.00298999882737	2.0055	0.286093
0.05	21	0.01	101	0.00074466360037	2.0017	3.422437
0.025	41	0.005	201	0.00018594563787	2.0052	96.72552
0.0125	81	0.0025	401	0.00004631927399	—	3531.15991

a tolerance of 0.000001 is used for Newton's method and Gauss–Seidel iterative method in the numerical scheme. All units of quantities in examples are in SI units. Exact solutions are not available for the considered examples.

Example 1. Consider the ADR problem in (4) with $u(x, y) = 0.3 + x$, $v(x, y) = 0.4 - y$ (see velocity streamlines in Figure 4), $Dx = 0.1x^2$, $Dy = 0.1y^2$ [1] (see graphs of Dx and Dy in Figure 5), $R(C) = C(1 - C)$ [30], initial condition

$$C(x, y, 0) = 0, \quad 0 < x, y < 1, \tag{23}$$

and boundary condition for $x, y \in \partial\Omega$ and $t > 0$

$$C(x, y, t) = \begin{cases} 1, & \text{for } x = 0 \text{ and } 0.4 \le y \le 0.6, \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Figure 6 displays the solution of Example 1 using the numerical scheme with spatial step sizes $\Delta x = \Delta y = 0.02$ and time step size $\Delta t = 0.001$ and at $t = 1$ and $t = 2$. In

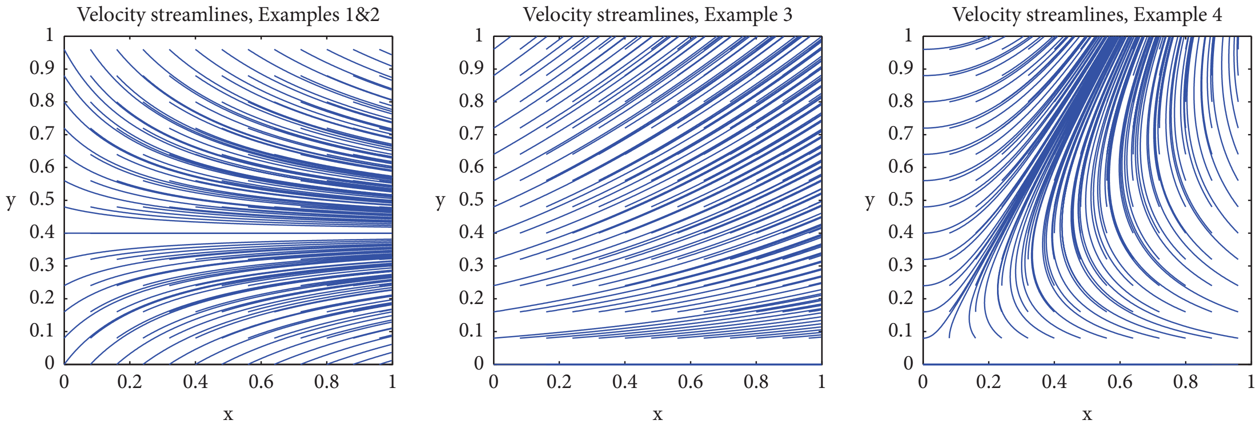


FIGURE 4: Velocity streamlines for velocities used in examples.

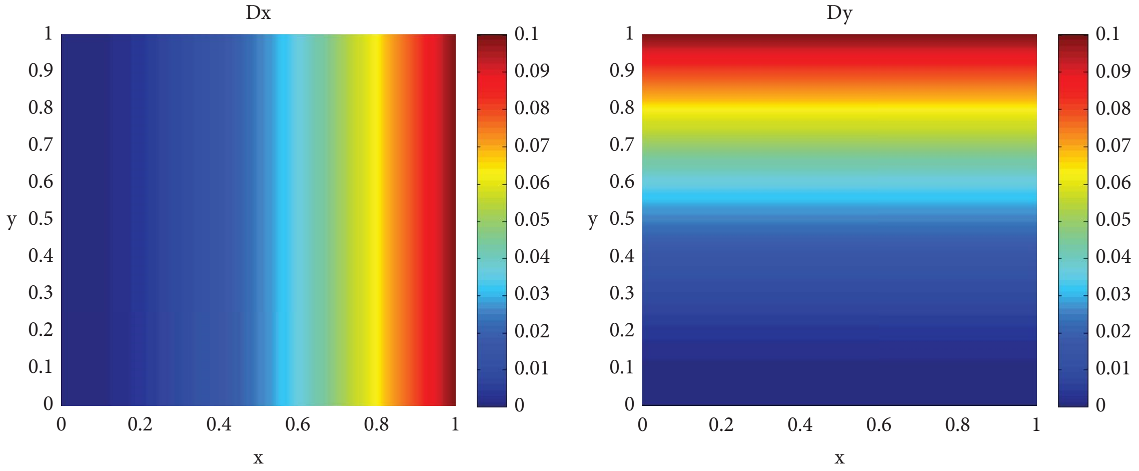


FIGURE 5: Graph of diffusion coefficients Dx and Dy used in Examples 1, 3, and 4.

solving the error correction vector h at each iteration, the matrix A for this problem is diagonally dominant and hence the numerical scheme yields convergent solution. The maximum absolute error obtained in the computation at $t=2$ using (16) is 0.1459. It is observed that as the maximum absolute error decreases, we take smaller step sizes. There is a rapid change of concentration distribution profile.

Example 2. Let us take the ADR problem in (4) with diffusion coefficient $Dx = 0.1(x + y - x^2 - y^2)$ and $Dy = 0.1(1 - xy)$ (see Graphs of Dx and Dy in Figure 7) having the same velocity components, reaction term, and initial and boundary conditions as Example 1.

The numerical solution of Example 2 at $t = 1$ and $t = 2$ is displayed in Figure 8. The same grid size is used as Example 1 for the computation. This example targets to see the effect of diffusion coefficient on the contaminant concentration distribution. As we can observe in Figure 8, higher values of the diffusion coefficient in the contamination transport give wider coverage area of concentration distribution for the same advection velocity which agrees with situation observed in [12]. The maximum value of the concentration at $t=2$ is attained near the entrance gate.

Example 3. Consider the ADR problem in (4) with $u(x, y) = e^{-x}\cosh y$ and $v(x, y) = e^{-x}\sinh y$ (see the velocity streamline in Figure 4) and having the same diffusion coefficients, reaction term, and initial and boundary conditions as Example 1.

The numerical solution of Example 3 at $t = 1$ and $t = 3$ is depicted in Figure 9. The same grid sizes are used as Example 1 to obtain the solution. For this problem, the matrix A that appears at each iteration in solving error correction vector h is also diagonally dominant. Hence, the numerical scheme is convergent. The maximum absolute error obtained in the computation is 0.2336. There is relatively slow change of concentration distribution profile between $t=1$ and $t=3$ as it can be observed from the graphs. The contaminant concentration distribution profile follows the advection velocity streamlines (see Figures 4 and 9).

Example 4. Let $u(x, y) = -x^2 + y^2$, $v(x, y) = 2xy$ (see the velocity streamlines in Figure 4), and $R(C) = C(1 - C)(C - 0.001)$ [29] in (4) having the same diffusion coefficient and initial and boundary conditions as Example 1.

In Example 4, the reaction term is a third degree polynomial. The numerical solution of this problem is

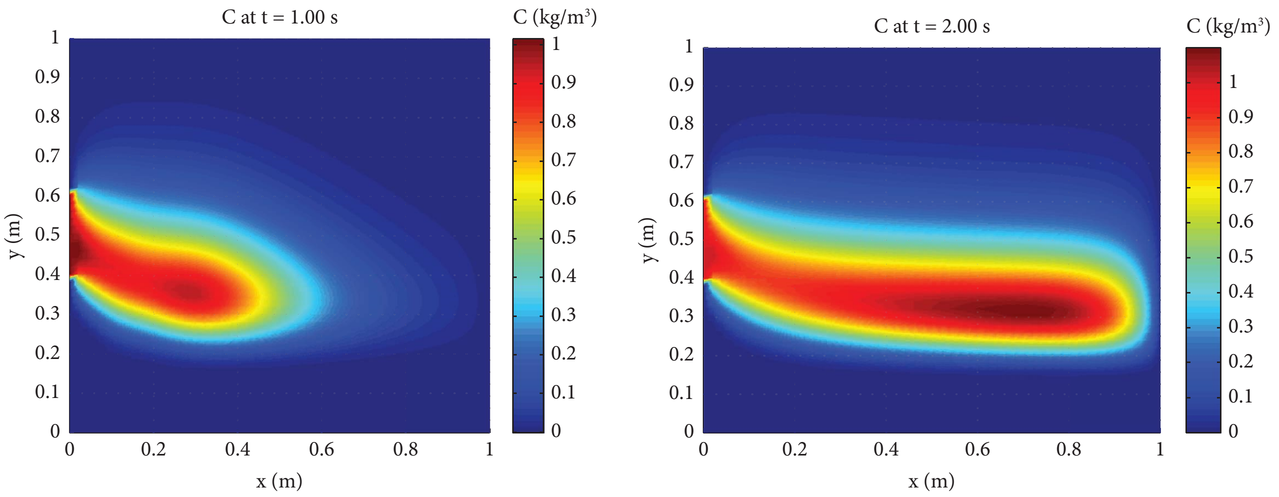


Figure 6: There is a rapid change of concentration distribution profile between $t = 1$ and $t = 2$. The maximum value of contaminate concentration at $t = 2$ is attained in the interior of the domain away from the entrance gate.

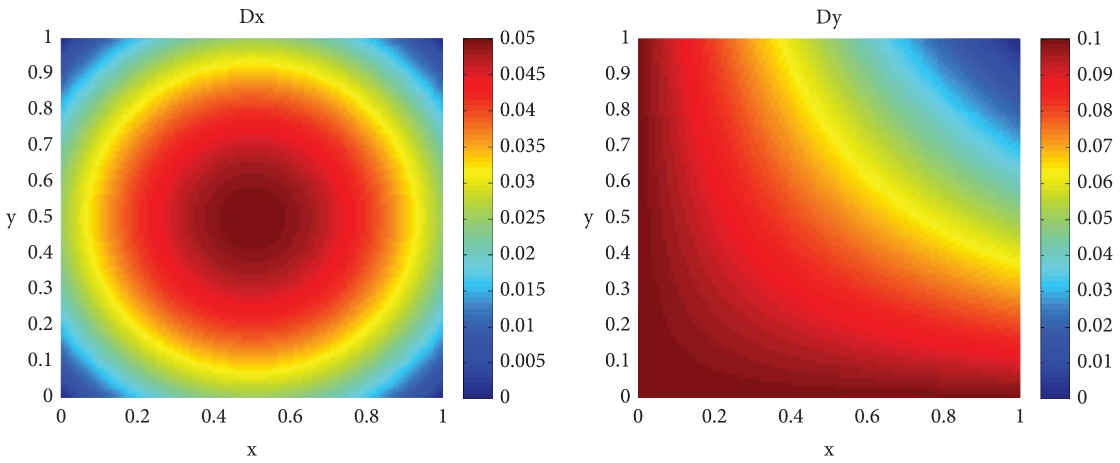


Figure 7: Graph of diffusion coefficients Dx and Dy used in Example 2.

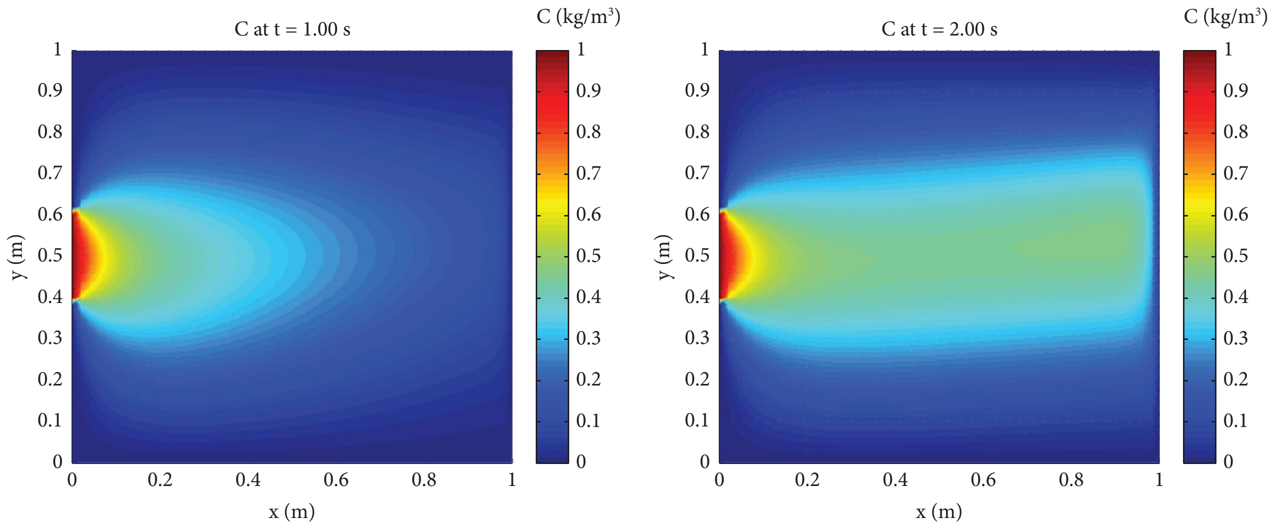


Figure 8: Numerical solution of Example 2 using $\Delta x = \Delta y = 0.02$ and $\Delta t = 0.005$ at $t = 1$ and $t = 2$.

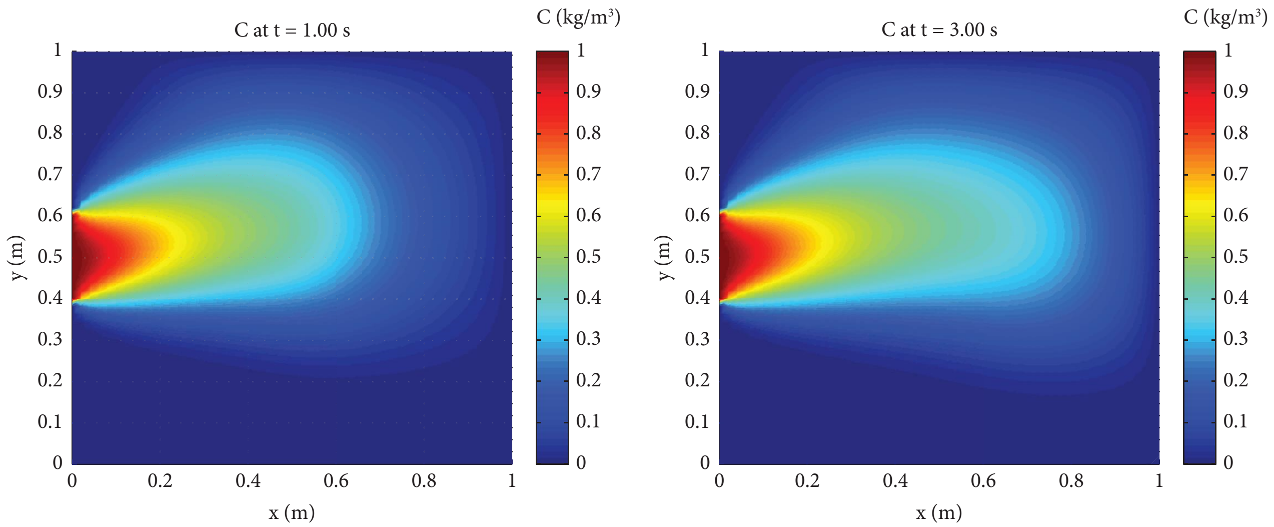


Figure 9: Numerical solution of Example 2 using $\Delta x = \Delta y = 0.02$ and $\Delta t = 0.005$ at $t = 1$ and $t = 3$.

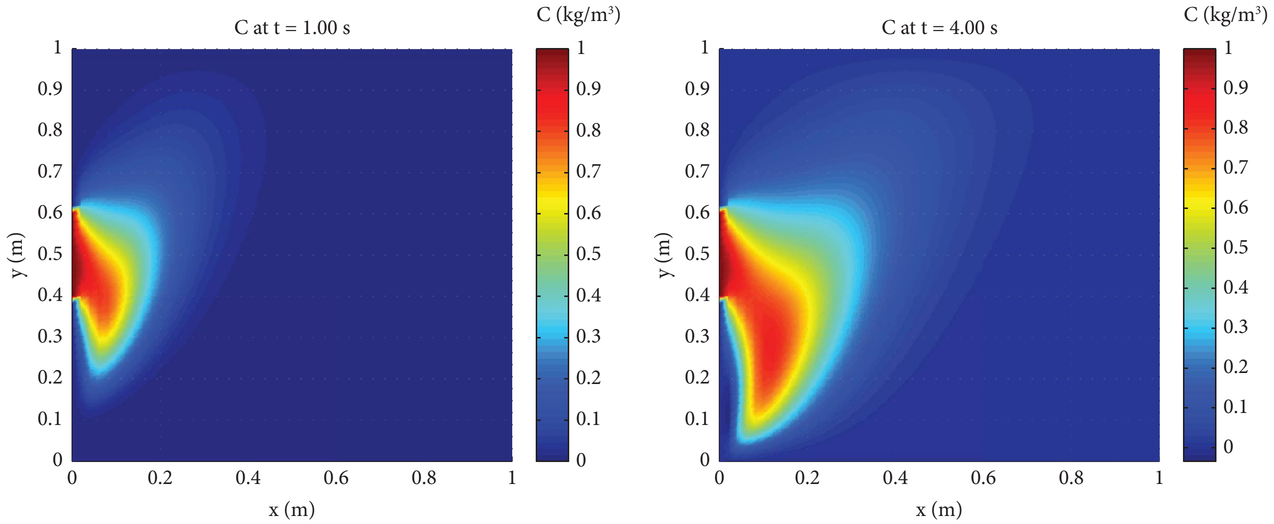


Figure 10: Numerical solution of Example 3 using $\Delta x = \Delta y = 0.02$ and $\Delta t = 0.005$ at $t = 1$ and $t = 4$.

presented in Figure 10 at $t = 1$ and $t = 4$. The same grid sizes are used as Example 1 in the computation. The numerical scheme yields a convergent solution. The maximum absolute error obtained in the computation is 0.1615.

For all examples, the maximum absolute errors are computed for the four sets of number of grid points $S_1 = \{11, 11, 126\}$, $S_2 = \{21, 21, 251\}$, $S_3 = \{41, 41, 501\}$, and $S_4 = \{81, 81, 1001\}$ for $0 \leq x, y \leq 1$ and $0 \leq t \leq 1$ using (16) as indicated in Table 1. For all problems, the maximum absolute error decreases as grid sizes decrease which shows the accuracy and the convergent of the proposed numerical scheme in solving the problems. Figure 4 shows the velocity stream lines of the three variable velocities in the examples. These streamlines are plotted using the streamline function of MATLAB. From the numerical results of problems in Examples 1, 3, and 4, it is observed that the contaminant concentration distribution profiles follow the fluid flow velocity streamlines which agreed with the condition realized in [1] for small diffusion coefficient. This reveals the applicability of the proposed numerical scheme in simulations of ADR problems.

8. Conclusions

In this study, a numerical scheme is employed to obtain the solution of two-dimensional unsteady nonlinear advection-diffusion-reaction equations with velocity and diffusion coefficients that vary spatially. By developing and implementing the comprehensive numerical scheme, which integrates the Crank–Nicolson Method, Newton's method, and the Gauss–Seidel iterative method, nonlinear ADR problems are successfully solved. The absolute maximum errors and convergence rate of the numerical scheme are estimated to investigate the accuracy and efficiency of the numerical scheme. The findings demonstrate the scheme's capability to accurately approximate solutions for ADR problems and its suitability for simulating related two-dimensional nonlinear partial differential equations. The scheme requires substantial CPU time during computation when a large number of spatial grid points are utilized.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] C. Zoppou and J. H. Knight, "Analytical solution of a spatially variable coefficient advection–diffusion equation in up to three dimensions," *Applied Mathematical Modelling*, vol. 23, no. 9, pp. 667–685, 1999.

[2] L. Li, Z. Jiang, and Z. Yin, "Fourth-order compact finite difference method for solving two-dimensional convection–diffusion equation," *Advances in Difference Equations*, vol. 2018, no. 1, p. 234, 2018.

[3] D. A. Garzón-Alvarado, C. H. Galeano, and J. M. Mantilla, "Computational examples of reaction–convection–diffusion equations solution under the influence of fluid flow: first example," *Applied Mathematical Modelling*, vol. 36, no. 10, pp. 5029–5045, 2012.

[4] M. Arshad, A. Hussain, A. Hassan et al., "Heat and mass transfer analysis above an unsteady infinite porous surface with chemical reaction," *Case Studies in Thermal Engineering*, vol. 36, Article ID 102140, 2022.

[5] M. R. Eid, "Chemical reaction effect on MHD boundary-layer flow of two-phase nanofluid model over an exponentially stretching sheet with a heat generation," *Journal of Molecular Liquids*, vol. 220, pp. 718–725, 2016.

[6] W. Stein, *Transport and Diffusion across Cell Membranes*, Elsevier, Amsterdam, Netherlands, 2012.

[7] N. Meghdadi, M. Soltani, H. Niroomand-Oscuii, and N. Yamani, "Personalized image-based tumor growth prediction in a convection–diffusion–reaction model," *Acta Neurologica Belgica*, vol. 120, no. 1, pp. 49–57, 2020.

[8] S. Komala Sheshachala and R. Codina, "Finite element modeling of nonlinear reaction–diffusion–advection systems of equations," *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 28, no. 11, pp. 2688–2715, 2018.

[9] K. Zamani and F. A. Bombardelli, "Analytical solutions of nonlinear and variable-parameter transport equations for verification of numerical solvers," *Environmental Fluid Mechanics*, vol. 14, no. 4, pp. 711–742, 2014.

[10] J. Pérez Guerrero, E. M. Pontedeiro, M. T. van Genuchten, and T. H. Skaggs, "Analytical solutions of the one-dimensional advection–dispersion solute transport equation subject to time-dependent boundary conditions," *Chemical Engineering Journal*, vol. 221, pp. 487–491, 2013.

[11] E. Ngondiep, "A novel three-level time-split approach for solving two-dimensional nonlinear unsteady convection-diffusion-reaction equation," *The Journal of Mathematics and Computer Science*, vol. 26, no. 3, pp. 222–248, 2021.

[12] K. Para, S. Sungnul, S. Sirisubtawee, and S. Phongthanapanich, "A comparison of numerical solutions for advection-diffusion-reaction equations between finite volume and finite difference methods," *Engineering Letters*, vol. 30, no. 2, 2022.

[13] J. Wei, Y. Ge, and Y. Wang, "High-order compact difference method for solving two and three-dimensional unsteady convection diffusion reaction equations," *Axioms*, vol. 11, no. 3, p. 111, 2022.

[14] A. Singh, S. Das, S. H. Ong, and H. Jafari, "Numerical solution of nonlinear reaction–advection–diffusion equation," *Journal of Computational and Nonlinear Dynamics*, vol. 14, no. 4, Article ID 041003, 2019.

[15] U. Ahmed, D. S. Mashat, and D. A. Maturi, "Finite volume method for a time-dependent convection-diffusion-reaction equation with small parameters," *International Journal of Differential Equations*, vol. 2022, Article ID 3476309, 15 pages, 2022.

[16] H. Mesgarani, M. Kermani, and M. Abbaszadeh, "Application of SPD-RBF method of lines for solving nonlinear advection–diffusion–reaction equation with variable coefficients," *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 32, no. 2, pp. 850–886, 2022.

[17] I. Ali, S. Haq, K. S. Nisar, and S. U. Arifeen, "Numerical study of 1D and 2D advection-diffusion-reaction equations using Lucas and Fibonacci polynomials," *Arabian Journal of Mathematics*, vol. 10, no. 3, pp. 513–526, 2021.

[18] E. Ngondiep, "A six-level time-split leap-frog/crank–nicolson approach for two-dimensional nonlinear time-dependent convection diffusion reaction equation," *International Journal of Computational Methods*, vol. 20, no. 8, Article ID 2250064, 2023.

[19] L. Zhang, Y. Ge, X. Yang, and Y. Zhang, "High accuracy compact difference and multigrid methods for two-dimensional time-dependent nonlinear advection-diffusion-reaction problems," *International Journal of Computer Mathematics*, vol. 100, no. 7, pp. 1552–1579, 2023.

[20] A. Tambue, "An exponential integrator for finite volume discretization of a reaction–advection–diffusion equation," *Computers and Mathematics with Applications*, vol. 71, no. 9, pp. 1875–1897, 2016.

[21] C. Yu, A. Deng, J. Ma, X. Cai, and C. Wen, "Semi-analytical solutions for two-dimensional convection–diffusion–reactive equations based on homotopy analysis method," *Environmental Science and Pollution Research*, vol. 25, no. 34, pp. 34720–34729, 2018.

[22] G. Gurarslan, H. Karahan, D. Alkaya, M. Sari, and M. Yasar, "Numerical solution of advection-diffusion equation using a sixth order compact finite difference method," *Mathematical Problems in Engineering*, vol. 2013, Article ID 672936, 7 pages, 2013.

[23] G. A. Putri, S. Sunarsih, and S. Hariyanto, "Numerical simulation of advection-diffusion on flow in waste stabilization ponds (1-dimension) with finite difference method forward time central space scheme," *Environmental Engineering Research*, vol. 23, no. 4, pp. 442–448, 2018.

[24] K. Pananu, S. Sungnul, S. Sirisubtawee, and S. Phongthanapanich, "Convergence and applications of the implicit finite difference method for advection-diffusion-reaction equations," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 645–663, 2020.

[25] R. S. Gupta, *Elements of Numerical Analysis*, Cambridge University Press, Cambridge, UK, 2nd edition, 2015.

[26] S. S. Ray, *Numerical Analysis with Algorithms and Programming*, Chapman and Hall/CRC, Nework, Boca Raton, FL, USA, 1st edition, 2016.

[27] B. McLaughlin, J. Peterson, and M. Ye, "Stabilized reduced order models for the advection–diffusion–reaction equation using operator splitting," *Computers and Mathematics with Applications*, vol. 71, no. 11, pp. 2407–2420, 2016.

[28] P. Roul and V. Rohil, "A high-order numerical scheme based on graded mesh and its analysis for the two-dimensional time-fractional convection-diffusion equation," *Computers and Mathematics with Applications*, vol. 126, pp. 1–13, 2022.

[29] S. A. Al-Bayati, *Boundary element analysis for convection-diffusion-reaction problems combining dual reciprocity and radial integration methods*, Ph.D. thesis, pp. 161–177, Brunel University, London, UK, 2018.

[30] V. S. Aswin and A. Awasthi, "A robust numerical scheme for the simulation of nonlinear convection–diffusion–reaction equation," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 20, no. 5, pp. 347–357, 2019.