

**IMPLEMENTATION OF ADVANCED NUMERICAL METHODS
FOR SOLVING NONLINEAR FISHER EQUATION**

Richa Kumari and Vikash Vimal*

Department of Physics,
Patna University,
Ashok Rajpath, Patna - 800005, Bihar, INDIA

E-mail : kricha061@gmail.com

*Department of Mathematics,
Nalanda University,
Rajgir, Nalanda - 803116, Bihar, INDIA

E-mail : vikashvimalnitjsr@gmail.com

(Received: Apr. 29, 2025 Accepted: Jul. 31, 2026 Published: Aug. 30, 2026)

Abstract: This paper presents higher order numerical methods for solving nonlinear Fisher equations. These types of equation arise in various fields of sciences and engineering, the main application of this equation has been found in biomedical sciences. The solution of this equation helps determine the size of the brain tumor. In this paper, we have constructed the numerical method based on the method of lines and higher order strong stability preserving schemes of order four and eight. These schemes are explicit in nature and easy to implement specially to solve the nonlinear problems. Due to the stability preserving nature of the scheme, the restriction on time step is very mild. These schemes are very practical to use and produce very accurate results. Various test problems are considered to validate the scheme along with a comparison of l_2 and l_∞ errors with the exact solution, resulting in high accuracy. The scheme is found to be better compared to existing schemes with less computational effort.

Keywords and Phrases: Fisher's problems, Method of lines, Finite difference methods, Strong stability preserving Runge-Kutta methods.

2020 Mathematics Subject Classification: 65M20, 92D25, 65L06, 65L07.

1. Introduction

We consider the well-known 1-D nonlinear Fisher equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u(1 - u), \quad 0 \leq x \leq 1, \quad t > 0, \quad (1.1)$$

where $u(x, t)$ population density.

The Fisher equation was initially formulated in 1937, and equation (1.1) is referenced from [9]. The details of the analysis and explanation of equation (1.1) can be found in [16]. Due to this it is called the Fisher- Kolmogorov-Petrovsky-Piscounov (Fisher-KPP) equation. However it is widely known as Fisher equation. Over time, the Fisher equation has become really important for studying how things spread out, like helpful genes, populations, and how stuff moves in nature and biology. Many scientists are studying different things using the Fisher equation [2, 3, 6, 7, 10, 25, 30, 31]. The Fisher equation's math properties and discussions, along with numerous of numerical methods, are explained in the literature with references for more information [1, 4, 12, 15, 17, 18, 19, 20, 23, 32, 36], References like [11, 22, 26, 27, 29] provide useful information about the background and uses of this equation in different scientific areas.

In mathematical science, we develop a method for obtaining numerical solutions of a one-dimensional nonlinear reaction-diffusion equation using the SSPRK-84 technique. We achieve this by transforming the PDEs into an ODEs in time through the application of the method of lines [21]. The method of lines, which is a technique for finding numerical solutions of partial differential equations, plays an essential role in preserving the accuracy and stability of the developing solution. The ordinary differential equations resulting from the discretization of the Navier-Stokes equations are integrated using an implicit method, Adams-Moulton, which is integrated with the widely recognized ODE solver Hindmarsh, Alan C [14]. The modified form of a nonlinear Fisher's reaction-diffusion equation solved by RBFs based on DQMs [13].

In this paper, we introduce a numerical approach for solving Fisher's equation. We combine the method of lines (MOL) in spatial dimensions with the strong stability preserving Runge-Kutta method (SSP-RK84) in time dimensions. SSPRK-84 method is a advanced techniques for solving high-order time discretization methods. In Section 2, we introduce Fisher equation in one dimension with initial and boundary conditions. In Section 3, involves semi-discretizing the derived equation in the spatial dimension using MOL and fully discretizing it by implementing the SSP-RK84 method on the resulting ODE system. In Section 4, we describe numerical experiments of test examples and compare the numerical solutions with a

few existing methods. Our method demonstrates higher accuracy compared to the existing methods. In Section 5, conclusion.

This paper presents an efficient method for addressing Fisher's equation, providing valuable perspectives for various scientific and engineering applications.

2. Problem statement

We consider the well-known one-dimensional nonlinear Fisher equation.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u(1 - u), \quad 0 \leq x \leq 1, \quad t > 0, \quad (2.1)$$

with initial condition

$$u(x, 0) = u_0(x), \quad 0 \leq x \leq 1,$$

and the boundary conditions

$$\begin{aligned} u(0, t) &= f_1(t), \quad 0 \leq t \leq T, \\ u(1, t) &= f_2(t), \quad 0 \leq t \leq T. \end{aligned}$$

In this context, α represents the reactive factor, T signifies the final time, and $u_0(x)$, f_1 , and f_2 denote given functions that are sufficiently smooth, collectively defining initial and boundary conditions for the mathematical model.

3. Numerical Scheme

The method of lines (MOL) is a widely recognized technique used for solving time-dependent partial differential equations. Initially, the PDEs are converted into ODEs using the method of lines (MOL). Then the set of ordinary differential equations is solved by applying the SSP-RK84 scheme for integration. To discretize the solution domain for equation (2.1), we apply a uniform mesh approach. The spatial interval $[0, 1]$ is divided into M equal sub-intervals, each with a width of Δx , where Δx is calculated as $\Delta x = \frac{1}{M}$. We then define spatial points x_m as $x_m = m\Delta x$ for m is ranging from 0 to M .

3.1. Semi-discretization: Method of Lines (MOL)

In his article [24], Rothe first introduced the method of lines (MOL), and in a subsequent works Bonkile [5], Parambu [22] and his team utilized MOL to transform the PDEs into a set of ODEs, effectively addressing the Burger's equation and Stefan problem. Unsteady nonlinear partial differential equation undergoes spatial discretization to create a semi-discrete method of lines (MOL) scheme. This includes discretizing the reaction term $\frac{\partial u}{\partial t}$ with a second-order central method and

using central difference to discretize the diffusion term $\frac{\partial^2 u}{\partial x^2}$.

$$\frac{\partial u}{\partial x} = \frac{u_{m+1}(t) - u_{m-1}(t)}{2h}, \quad (3.1)$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{m+1}(t) - 2u_m(t) + u_{m-1}(t)}{h^2}, \quad h = \Delta x, \quad (3.2)$$

$$\frac{du_m}{dt} = \left(\frac{u_{m+1}(t) - 2u_m(t) + u_{m-1}(t)}{h^2} \right) + (\alpha u_m(1 - u_m)), \quad (3.3)$$

where $m = 1, 2, 3, \dots, M - 1$.

Can be written in a discrete operator form, the right-hand side of the above equation (3.3).

$$\frac{du_m}{dt} = L(u_m). \quad (3.4)$$

Where m is ranging from 1 to $M - 1$, and L is basically a nonlinear difference operator.

3.2. SSP-RK84

The purpose of SSP-RK84 is to achieve high-order accuracy in time integration while maintaining strong stability properties. The SSP-RK84 scheme utilizes seven stages ($s = 8$) and achieves a fourth-order accuracy ($k = 4$). The primary characteristic of SSP methods is that the number of stages ($s = 8$) exceeds the order ($k = 4$) of accuracy in terms of the time variable for the method [11]. Let's express an s-stage explicit Runge-Kutta method in the following manner,

$$U^{(0)} = U^n \quad (3.5)$$

$$U^{(i)} = \sum_{k=0}^{i-1} (a_{i,k} U^{(k)} + \Delta t b_{i,k} L(U^{(k)})), \quad i = 1, 2, 3, \dots, s \quad (3.6)$$

$$U^{n+1} = U^{(m)} \quad (3.7)$$

The SSP-RK84 scheme is characterized by coefficients $a_{i,k}$ that satisfy the conditions $a_{i,k} \geq 0$ and $a_{i,k} = 0$ only if $b_{i,k} = 0$ [27]. This scheme also possesses a Courant-Friedrichs-Lewy (CFL) coefficient of 4.14593344751963. Additionally, it's required that the sum of coefficients $\sum_{k=0}^{i-1} a_{i,k} = 1$ holds for $i = 1, 2, 3, \dots, s$.

To divide the time interval $[0, T]$ into N equal parts with a consistent size, we use $\Delta t = 1/N = k$, where $t_n = n\Delta t$. We perform the integration of equation (3.4) from t_n to $t_n + \Delta t$ using the following steps for $n = 0, 1, 2, \dots, N$, resulting in the complete determination of the solution $u(x, t)$ at a specific time level. Table (1) gives the values of a_{ik} and b_{ik} coefficients.

$$u_m^{(0)} = u_m^n, \quad (3.8)$$

where u_m^n is a initial condition.

$$\begin{aligned} u_m^{(1)} &= a_{10}u_m^{(0)} + \Delta t b_{10}L(u_m^{(0)}) \\ u_m^{(1)} &= u_m^{(0)} + \Delta t(0.24120020561311)L(u_m^{(0)}) \end{aligned} \quad (3.9)$$

$$\begin{aligned} u_m^{(1)} &= u_m^{(0)} + k(0.24120020561311) \left[\frac{u_{m+1}^{(0)} - 2u_m^{(0)} + u_{m-1}^{(0)}}{h^2} + \alpha u_m^{(0)}(1 - u_m^{(0)}) \right] \\ u_m^{(2)} &= \sum_{k=0}^1 (a_{2k}u_m^{(k)} + \Delta t b_{2k}L(u_m^{(k)})) \\ u_m^{(2)} &= (0.10645325745007)u_m^{(0)} + (0.89354674254993)u_m^{(1)} + \Delta t(0.21552365802797)L(u_m^{(1)}) \end{aligned} \quad (3.10)$$

$$\begin{aligned} u_m^{(2)} &= (0.10645325745007)u_m^{(0)} + (0.89354674254993)u_m^{(1)} \\ &\quad + k(0.21552365802797) \left[\frac{u_{m+1}^{(1)} - 2u_m^{(1)} + u_{m-1}^{(1)}}{h^2} + \alpha u_m^{(1)}(1 - u_m^{(1)}) \right] \\ u_m^{(3)} &= \sum_{k=0}^2 (a_{3k}u_m^{(k)} + \Delta t b_{3k}L(u_m^{(k)})) \\ u_m^{(3)} &= u_m^{(2)} + \Delta t \cdot (0.24120020561311)L(u_m^{(2)}) \end{aligned} \quad (3.11)$$

$$u_m^{(3)} = u_m^{(2)} + k \cdot (0.24120020561311) \left(\frac{u_{m+1}^{(2)} - 2u_m^{(2)} + u_{m-1}^{(2)}}{h^2} + \alpha u_m^{(2)}(1 - u_m^{(2)}) \right)$$

$$\begin{aligned} u_m^{(4)} &= \sum_{k=0}^3 (a_{4k}u_m^{(k)} + \Delta t b_{4k}L(u_m^{(k)})) \\ u_m^{(4)} &= (0.57175518477257)u_m^{(0)} + (0.42824481522743)u_m^{(3)} + \Delta t(0.10329273748560)L(u_m^{(3)}) \end{aligned} \quad (3.12)$$

$$\begin{aligned} u_m^{(4)} &= (0.57175518477257)u_m^{(0)} + (0.42824481522743)u_m^{(3)} + \\ &\quad k(0.10329273748560) \left[\frac{u_{m+1}^{(3)} - 2u_m^{(3)} + u_{m-1}^{(3)}}{h^2} + \alpha u_m^{(3)}(1 - u_m^{(3)}) \right] \end{aligned}$$

$$\begin{aligned} u_m^{(5)} &= \sum_{k=0}^4 (a_{5k}u_m^{(k)} + \Delta t b_{5k}L(u_m^{(k)})) \\ u_m^{(5)} &= (.19161667219044)u_m^{(0)} + (0.80838332780956)u_m^{(4)} + \Delta t(0.19498222488188)L(u_m^{(4)}) \end{aligned} \quad (3.13)$$

$$\begin{aligned}
u_m^{(5)} &= (.19161667219044)u_m^{(0)} + (0.80838332780956)u_m^{(4)} + \\
&\quad k(0.19498222488188) \left[\frac{u_{m+1}^{(4)} - 2u_m^{(4)} + u_{m-1}^{(4)}}{h^2} + \alpha u_m^{(4)}(1 - u_m^{(4)}) \right] \\
u_m^{(6)} &= \sum_{k=0}^5 (a_{6k}u_m^{(k)} + \Delta t b_{6k}L(u_m^{(k)})) \\
u_m^{(6)} &= u_m^{(5)} + \Delta t(0.24120020561311)L(u_m^{(5)})
\end{aligned} \tag{3.14}$$

$$\begin{aligned}
u_m^{(6)} &= u_m^{(5)} + k(0.24120020561311) \left[\frac{u_{m+1}^{(5)} - 2u_m^{(5)} + u_{m-1}^{(5)}}{h^2} + \alpha u_m^{(5)}(1 - u_m^{(5)}) \right] \\
u_m^{(7)} &= \sum_{k=0}^6 (a_{7k}u_m^{(k)} + \Delta t b_{7k}L(u_m^{(k)})) \\
u_m^{(7)} &= u_m^{(6)} + \Delta t(0.24120020561311)L(u_m^{(6)})
\end{aligned} \tag{3.15}$$

$$\begin{aligned}
u_m^{(7)} &= u_m^{(6)} + k(0.24120020561311) \left[\frac{u_{m+1}^{(6)} - 2u_m^{(6)} + u_{m-1}^{(6)}}{h^2} + \alpha u_m^{(6)}(1 - u_m^{(6)}) \right] \\
u_m^{(n+1)} &= \sum_{k=0}^7 (a_{8k}u_m^{(k)} + \Delta t b_{8k}L(u_m^{(k)}))
\end{aligned}$$

$$\begin{aligned}
u_m^{(n+1)} &= (0.02580435327923)u_m^{(0)} + (0.03629901341774)u_m^{(2)} + (0.31859181340256)u_m^{(3)} + \\
&\quad (0.05186768980103)u_m^{(4)} + (0.03944076217320)u_m^{(5)} + \\
&\quad (0.00511633747411)u_m^{(6)} + (0.52288003045213)u_m^{(7)} + \\
&\quad \Delta t((0.00875532949991)kL(u_m^{(2)}) + (0.06195575835101)kL(u_m^{(3)}) + \\
&\quad (0.00951311994571)kL(u_m^{(5)}) + (0.12611877085604)kL(u_m^{(7)}))
\end{aligned} \tag{3.16}$$

$$\begin{aligned}
u_m^{(n+1)} &= (0.02580435327923)u_m^{(0)} + (0.03629901341774)u_m^{(2)} + (0.31859181340256)u_m^{(3)} + \\
&\quad (0.05186768980103)u_m^{(4)} + (0.03944076217320)u_m^{(5)} + (0.00511633747411)u_m^{(6)} + \\
&\quad (0.52288003045213)u_m^{(7)} +
\end{aligned}$$

$$\begin{aligned}
& k(0.00875532949991)[\alpha(t_n)\frac{u_{m+1}^{(2)} - 2u_m^{(2)} + u_{m-1}^{(2)}}{h^2} + \alpha u_m^{(2)}(1 - u_m^{(2)})] + \\
& k(0.06195575835101)[\alpha(t_n)\frac{u_{m+1}^{(3)} - 2u_m^{(3)} + u_{m-1}^{(3)}}{h^2} + \alpha u_m^{(3)}(1 - u_m^{(3)})] \\
& k(0.00951311994571)[\alpha(t_n)\frac{u_{m+1}^{(5)} - 2u_m^{(5)} + u_{m-1}^{(5)}}{h^2} + \alpha u_m^{(5)}(1 - u_m^{(5)})] \\
& k(0.12611877085604)[\alpha(t_n)\frac{u_{m+1}^{(7)} - 2u_m^{(7)} + u_{m-1}^{(7)}}{h^2} + \alpha u_m^{(7)}(1 - u_m^{(7)})]
\end{aligned}$$

For $m = 1, 2, 3, \dots, M-1$. In the succeeding iteration, we have utilized $u_m^{(0)} = u_m^{n+1}$, with n ranging from 0 to $N-1$.

4. Numerical Experiment

Presented are the numerical results of the SSP-RK84 method applied to various instances of the Fisher equation (1.1) using MATLAB. With the help of the exact solution, we measured the accuracy of the numerical method. Assess the accuracy and efficiency of the proposed method by evaluating the l_2 and l_∞ error norms.

$$l_2 = \left[\frac{1}{M} \sum_{m=0}^M (U_m - u_m)^2 \right]^{1/2}, \quad l_\infty = \max_{0 \leq m \leq M} |U_m - u_m|.$$

Here, u_m is numerical solution and U_m as the similarity solution corresponding to the node at position x_m .

Example 4.1. Consider the Fisher Equation

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

subject to the initial condition

$$u(x, 0) = \frac{1}{(1 + e^{\sqrt{\frac{\alpha}{6}}x})^2},$$

where the exact solution is presented in [14] given by

$$u(x, t) = \frac{1}{(1 + e^{\sqrt{\frac{\alpha}{6}}x - \frac{5}{6}\alpha t})^2}.$$

Example 4.2. Consider the following generalized Fisher's equation in domain $[0, 1]$:

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

with initial condition

$$u(x, 0) = \left\{ \frac{1}{2} \tanh \left(-\frac{\alpha}{2\sqrt{2\alpha+4}} x \right) + \frac{1}{2} \right\}^{\frac{2}{\alpha}}.$$

The exact solution is presented in [14] by

$$u(x, 0) = \left\{ \frac{1}{2} \tanh \left(-\frac{\alpha}{2\sqrt{2\alpha+4}} \left(x - \frac{\alpha+4}{\sqrt{2\alpha+4}} t \right) \right) + \frac{1}{2} \right\}^{\frac{2}{\alpha}}.$$

Table 2 and 3 provide numerical and exact values at different times for various values of ' α ' for examples one and two. We compare the numerical and exact values for $\alpha = 6$ and 1 using at two distinct time steps, namely $\Delta t = 0.0001$ and $\Delta t = 0.00005$, with a specific focus on examples (4.1) and (4.2). Figures 1, 2 and 5, 6 represent numerical values compared to the exact values, while Figures 3, 4 and 7, 8 display the absolute error graphs in relation to the exact values for $\alpha = 6$ and $\alpha = 1$, with Δt values of 0.0005 and 0.000001. Similarly Table 3 and Figures 9, 10 at different values. Our proposed numerical approaches, given in Tables 5 and 6, perform better results. Tables 12 and 13 provide the l_2 and l_∞ errors for examples one and two. From these tables and graphs, we observed that the proposed method yields more accurate values for all time steps.

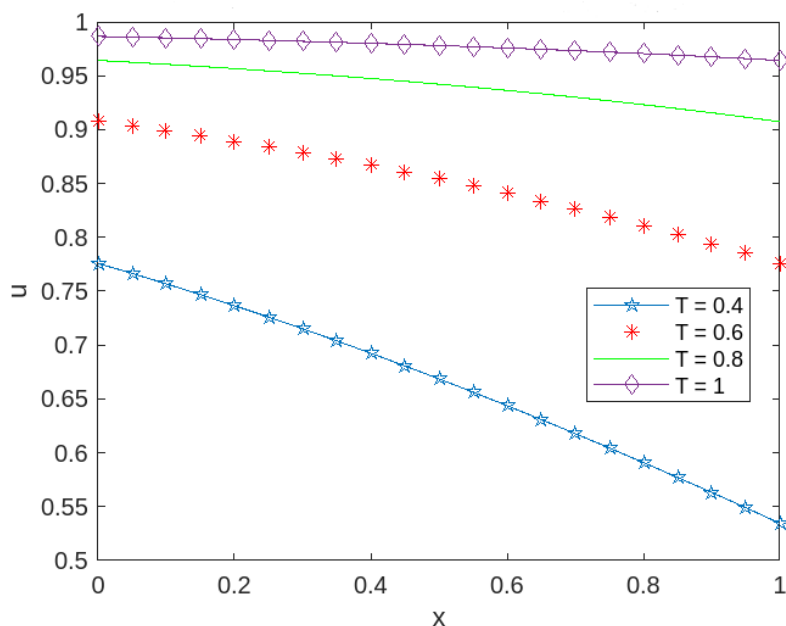


Figure 1: Numerical value at $\alpha = 6$, $\Delta t = 0.0005$, and $N = 20$ for Example (4.1).

Table 2: Numerical and exact value of Example (4.1) at $\alpha = 6$ and $N = 20$.

x	T	$\Delta t = 0.0005$		$\Delta t = 0.000001$	
		Numerical value	Exact value	Numerical value	Exact value
0.25	0.4	0.726023	0.725824	0.7258260	0.7258236
	0.6	0.883538	0.883437	0.8834382	0.8834371
	0.8	0.954614	0.954573	0.9545731	0.9545732
	1.0	0.982934	0.982919	0.9829184	0.9829186
0.5	0.4	0.668653	0.668428	0.6684312	0.6684280
	0.6	0.854161	0.854038	0.8540399	0.8540381
	0.8	0.942286	0.942235	0.9422347	0.9422347
	1.0	0.978167	0.978147	0.9781466	0.9781468
0.75	0.4	0.604445	0.604195	0.6041977	0.6041951
	0.6	0.818540	0.818393	0.8183943	0.8183926
	0.8	0.926804	0.926740	0.9267395	0.9267395
	1.0	0.972096	0.972071	0.9720703	0.9720705

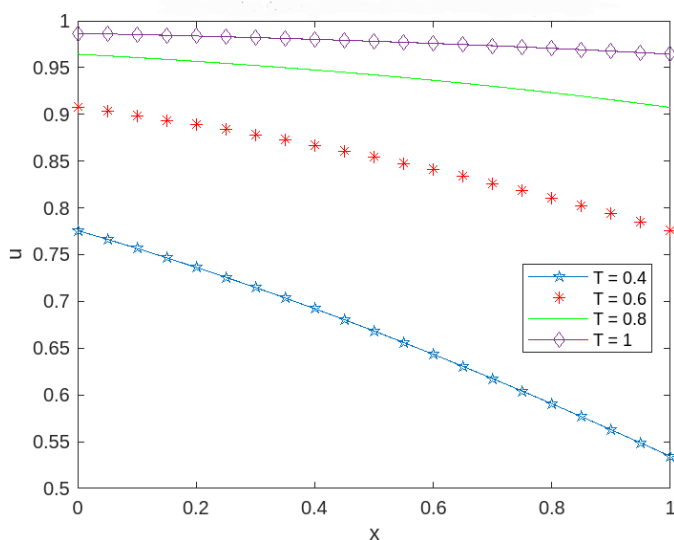
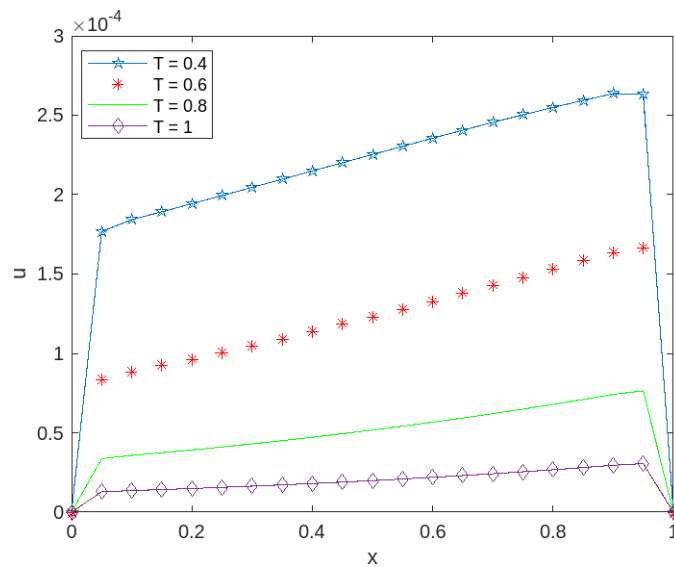
Figure 2: Numerical value at $\alpha = 6$, $\Delta t = 0.000001$, and $N = 20$ of Example (4.1).

Table 3: Numerical and exact value of Example (4.2) at $\alpha = 1$ and $N = 20$.

x	T	$\Delta t = 0.0005$		$\Delta t = 0.000001$	
		Numerical value	Exact value	Numerical value	Exact value
0.25	0.4	0.310917	0.310875	0.31087477	0.31087480
	0.6	0.357879	0.357834	0.35783382	0.35783382
	0.8	0.406474	0.406428	0.40642834	0.40642829
	1.0	0.455785	0.455739	0.45573875	0.45573865
0.5	0.4	0.283338	0.283298	0.28329788	0.28329797
	0.6	0.328870	0.328827	0.32882654	0.32882658
	0.8	0.376572	0.376526	0.37652604	0.37652602
	1.0	0.425555	0.425509	0.42550859	0.42550850
0.75	0.4	0.256871	0.256832	0.25683236	0.25683242
	0.6	0.300678	0.300635	0.30063543	0.30063545
	0.8	0.347150	0.347106	0.34710568	0.34710566
	1.0	0.395457	0.395411	0.39541150	0.39541142

Figure 3: Absolute errors at $\alpha = 6$, $\Delta t = 0.0005$, and $N = 20$ of Example (4.1).

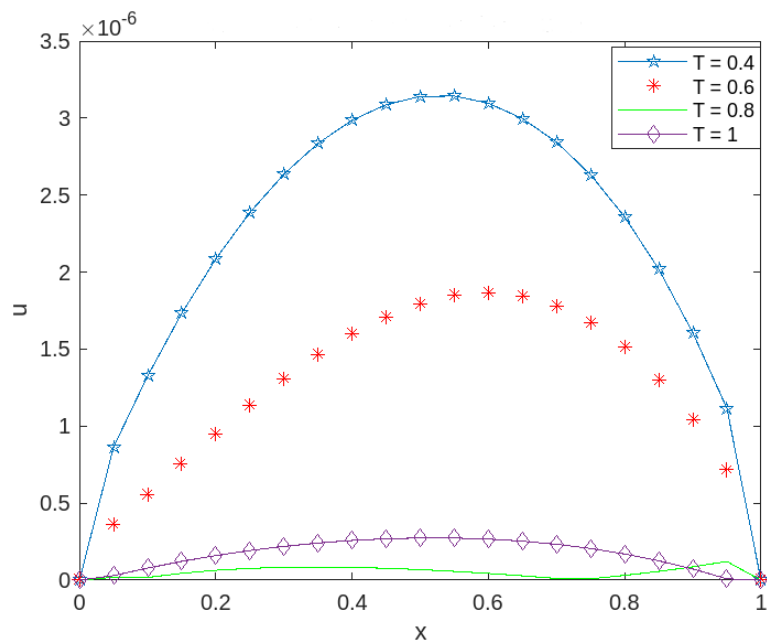


Figure 4: Absolute errors at $\alpha = 6$, $\Delta t = 0.000001$, and $N = 20$ of Example (4.1).

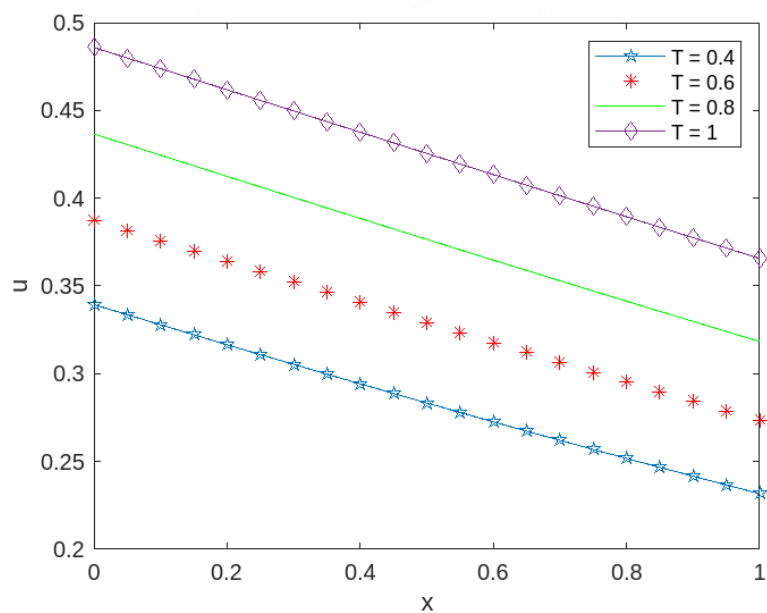


Figure 5: Solution at $\alpha = 1$, $\Delta t = 0.0005$, and $N = 20$ for Example (4.2).

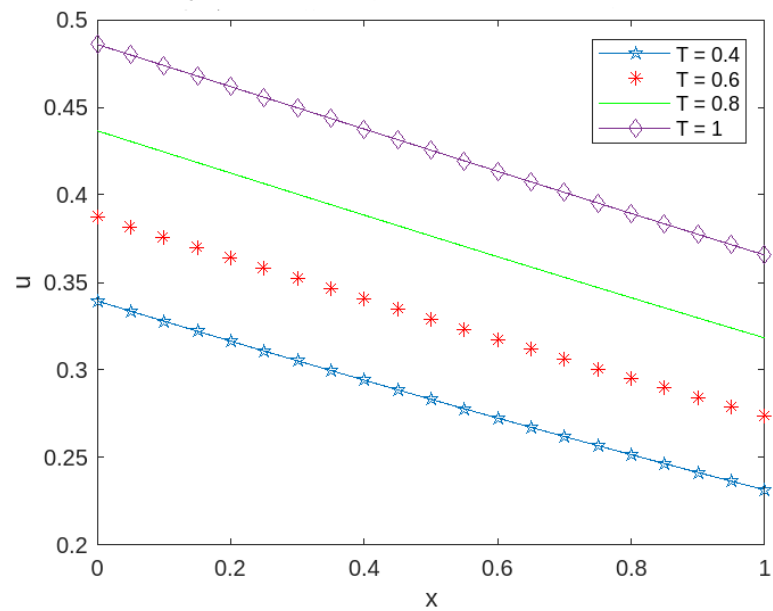


Figure 6: Numerical value at $\alpha = 1$, $\Delta t = 0.000001$, and $N = 20$ of Example (4.2).

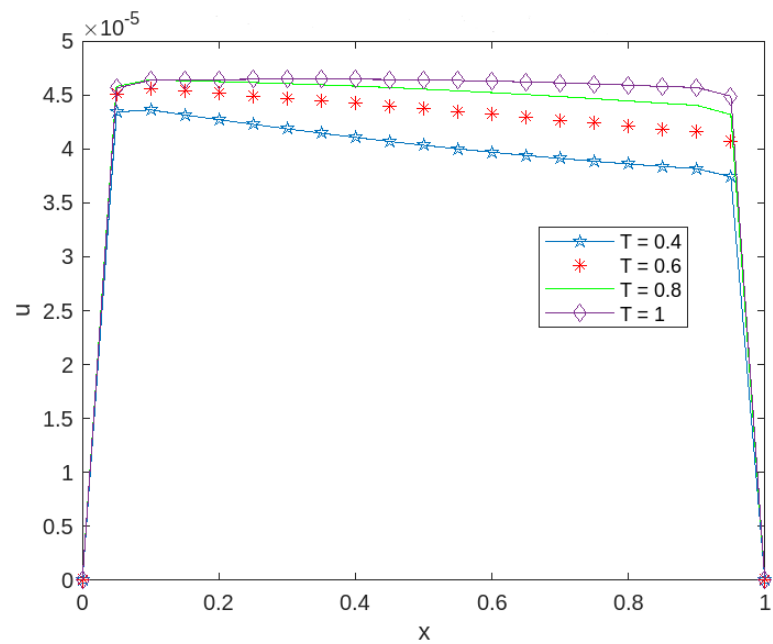


Figure 7: Absolute errors at $\alpha = 1$, $\Delta t = 0.0005$, and $N = 20$ of Example (4.2).

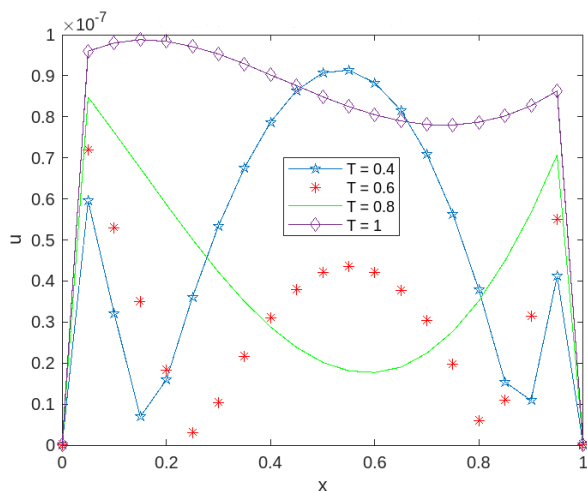


Figure 8: Absolute errors at $\alpha = 1$, $\Delta t = 0.000001$, and $N = 20$ of Example (4.2).

Table 4: Numerical and exact value of Example (4.1) at $\Delta t = 0.001$, $N = 20$ and $T = 1$.

α	x	Numerical value	Exact value	α	Numerical value	Exact value
0.1	0.2	0.26458908	0.26458102	1	0.46187485	0.46178202
	0.4	0.25799346	0.25798550		0.43768942	0.43759651
	0.6	0.25147843	0.25147057		0.41353432	0.41344174
	0.8	0.24504623	0.24503847		0.38951863	0.38942680
0.3	0.2	0.30378117	0.30375552	10	0.99937984	0.99937793
	0.4	0.29168281	0.29165758		0.99919724	0.99919478
	0.6	0.27978818	0.27976339		0.99896095	0.99895775
	0.8	0.26811129	0.26808697		0.99865519	0.99865102
0.5	0.2	0.34670869	0.34666400	100	0.99999983	0.99999983
	0.4	0.33036437	0.33032033		0.99999976	0.99999976
	0.6	0.31427811	0.31423483		0.99999966	0.99999965
	0.8	0.29848464	0.29844219		0.99999951	0.99999950

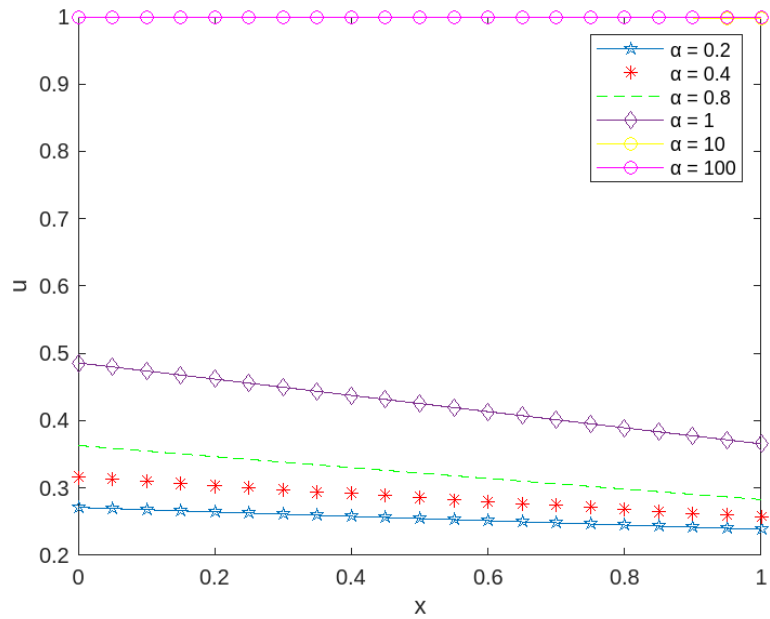


Figure 9: Numerical value at $\Delta t = 0.001$, $T = 1$ and $N = 20$ for Example (4.1).

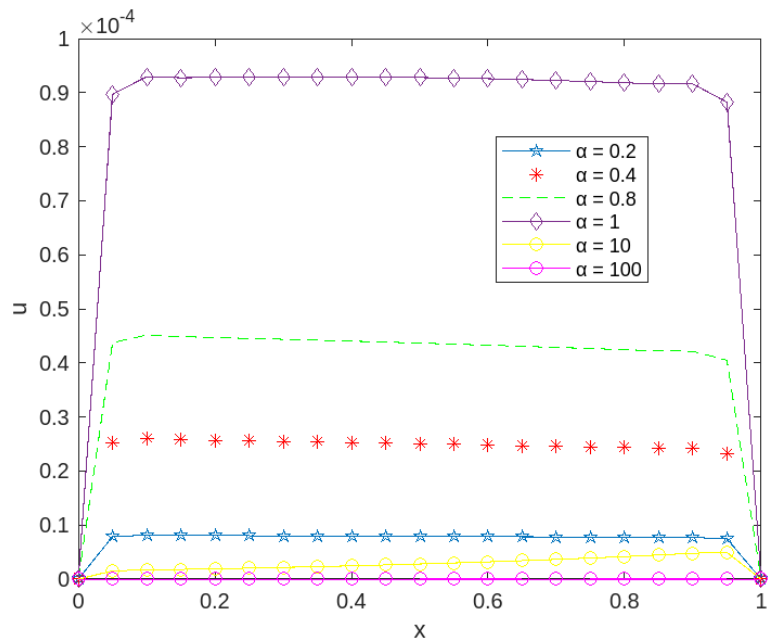


Figure 10: Absolute error at $\Delta t = 0.001$, $T = 1$ and $N = 20$ for Example (4.1).

Table 5: Present and existing numerical value comparison with exact value at $\alpha = 6$, $N = 20$ for Example (4.1).

x	T	$\Delta t = 0.0001$		$\Delta t = 0.00005$		
		R.Mittal [19]	Present value	R.Mittal [19]	Present value	Exact value
0.25	0.5	0.81847	0.818424	0.81843	0.818409	0.8183926
	1.0	0.98293	0.982922	0.98292	0.982920	0.9829186
	2.0	0.99988	0.999883	0.99988	0.999883	0.9998834
	5.0	1.00000	1.000000	1.00000	1.000000	1.0000000
0.5	0.5	0.77590	0.775841	0.77580	0.775824	0.7758035
	1.0	0.97816	0.978151	0.97814	0.978149	0.9781468
	2.0	0.99985	0.999850	0.99985	0.999850	0.9998503
	5.0	1.00000	1.000000	1.00000	1.0000000	1.0000000
0.75	0.5	0.72594	0.725866	0.72588	0.725846	0.7258236
	1.0	0.97209	0.972075	0.92208	0.972073	0.9720705
	2.0	0.99981	0.999808	0.99981	0.999808	0.9998078
	5.0	1.00000	1.000000	1.00000	1.000000	1.0000000

Table 6: Present and existing numerical value comparison with exact value at $\alpha = 1$, $N = 20$ for Example (4.2).

x	T	$\Delta t = 0.0001$		$\Delta t = 0.00005$		
		R.Mittal [19]	Present value	R.Mittal [19]	Present value	Exact value
0.25	0.5	0.33412	0.334103	0.33409	0.334098	0.334094
	1.0	0.45576	0.455748	0.45574	0.455743	0.455739
	2.0	0.68397	0.683958	0.68394	0.683954	0.683951
	5.0	0.96653	0.966526	0.96653	0.966526	0.966525
0.5	0.5	0.30576	0.305747	0.30574	0.305743	0.305739
	1.0	0.42553	0.425518	0.42551	0.425513	0.425509
	2.0	0.65924	0.659224	0.65921	0.659220	0.659216
	5.0	0.96303	0.963030	0.96303	0.963029	0.963028
0.75	0.5	0.27838	0.278361	0.27835	0.278357	0.278353
	1.0	0.39544	0.395421	0.39542	0.395416	0.395411
	2.0	0.63338	0.633366	0.63336	0.633362	0.633358
	5.0	0.95918	0.959179	0.95918	0.959179	0.959178

Table 7: Comparison of numerical solution for Example (4.1) at $\Delta t = 0.000005$, $T = 0.1$ and $\alpha = 6$.

x	BDF1 [33]	BDF2 [33]	Present solution	Exact solution
0.1	0.35841806	0.35842071	0.35842355	0.35842691
0.2	0.32997086	0.32997260	0.32997547	0.32998421
0.3	0.30230060	0.30230157	0.30230442	0.30231742
0.4	0.27558402	0.27558442	0.27558724	0.27560315
0.5	0.24997987	0.24997993	0.24998270	0.25000000
0.6	0.22562504	0.22562500	0.22562770	0.22564477
0.7	0.20263156	0.20263170	0.20263428	0.20264943
0.8	0.18108475	0.18108534	0.18108774	0.18109917
0.9	0.16104234	0.16104363	0.16104574	0.16105159

Table 8: Comparison of numerical solution for Example (4.1) at $\Delta t = 0.000005$, $T = 0.1$ and $\alpha = 1$.

x	BDF1 [33]	BDF2 [33]	Present solution	Exact solution
0.1	0.26073733	0.26073824	0.26073859	0.26073843
0.2	0.25042002	0.25042078	0.25042106	0.25042110
0.3	0.24031064	0.24031127	0.24031151	0.24031169
0.4	0.23041738	0.23041791	0.23041812	0.23041838
0.5	0.22074766	0.22074815	0.22074835	0.22074865
0.6	0.21130823	0.21130872	0.21130891	0.21130920
0.7	0.20210505	0.20210558	0.20210579	0.20210601
0.8	0.19314332	0.19314394	0.19314418	0.19314428
0.9	0.18442751	0.18442824	0.18442851	0.18442843

Table 9: Comparison of numerical solution for Example (4.1) at $\Delta t = 0.000005$, $T = 0.1$ and $\alpha = 6$.

x	BDF1 [34]	BDF2 [34]	Present solution	Exact solution
0.1	0.35840890	0.35842119	0.35842355	0.35842691
0.2	0.32996342	0.32997349	0.32997547	0.32998421
0.3	0.30229465	0.30230278	0.30230442	0.30231742
0.4	0.27557923	0.27558586	0.27558724	0.27560315
0.5	0.24997582	0.24998148	0.24998270	0.25000000
0.6	0.22562126	0.22562655	0.22562770	0.22564477
0.7	0.20262760	0.20263311	0.20263428	0.20264943
0.8	0.18108020	0.18108646	0.18108774	0.18109917
0.9	0.16103685	0.16104428	0.16104574	0.16105159

Table 10: Comparison of numerical solution for Example (4.1) at $\Delta t = 0.000005$, $T = 0.1$ and $\alpha = 6$.

x	BDF1 [34]	BDF2 [34]	ssprk43 [35]	Present solution	Exact solution
0.1	0.35840890	0.35842119	0.35842328	0.35842355	0.35842691
0.2	0.32996342	0.32997349	0.32997524	0.32997547	0.32998421
0.3	0.30229465	0.30230278	0.30230424	0.30230442	0.30231742
0.4	0.27557923	0.27558586	0.27558708	0.27558724	0.27560315
0.5	0.24997582	0.24998148	0.24998256	0.24998270	0.25000000
0.6	0.22562126	0.22562655	0.22562757	0.22562770	0.22564477
0.7	0.20262760	0.20263311	0.20263415	0.20263428	0.20264943
0.8	0.18108020	0.18108646	0.18108759	0.18108774	0.18109917
0.9	0.16103685	0.16104428	0.16104557	0.16104574	0.16105159

Table 11: Comparison of numerical solution for Example (4.2) at $\Delta t = 0.000005$, $T = 0.1$ and $\alpha = 1$.

x	BDF1 [33]	BDF2 [33]	BDF3 [33]	Present solution	Exact solution
0.1	0.260737784	0.260738230	0.260738056	0.260738590	0.260738428
0.2	0.250420391	0.250420750	0.250420905	0.250421064	0.250421096
0.3	0.240310950	0.240311240	0.240311656	0.240311514	0.240311688
0.4	0.230417634	0.230417876	0.230418460	0.230418121	0.230418385
0.5	0.220747899	0.220748116	0.220748764	0.220748347	0.220748648
0.6	0.211308466	0.211308683	0.211309286	0.211308913	0.211309201
0.7	0.202105305	0.202105547	0.202106001	0.202105789	0.202106010
0.8	0.193143625	0.193143913	0.193144130	0.193144177	0.193144276
0.9	0.184427868	0.184428219	0.184428133	0.184428511	0.184428430

Table 12: Errors of Example (4.1) at $\alpha = 6$.

$\Delta t = 0.0005$			$\Delta t = 0.000001$	
T	l_2	l_∞	l_2	l_∞
0.4	2.20E-04	2.64E-04	2.40627E-06	3.1436E-06
0.6	1.24E-04	1.66E-04	1.37048E-06	1.8643E-06
0.8	5.33868E-05	7.64021E-05	6.21653E-08	1.214E-07
1.0	2.08138E-05	3.06015E-05	1.95289E-07	2.742E-07

Table 13: Errors of Example (4.2) at $\alpha = 1$.

$\Delta t = 0.0005$			$\Delta t = 0.000001$	
T	l_2	l_∞	l_2	l_∞
0.4	3.95488E-05	4.36174E-05	5.91446E-08	9.14E-08
0.6	4.24829E-05	4.55655E-05	3.52465E-08	7.2E-08
0.8	4.41753E-05	4.64187E-05	4.5857E-08	8.48E-08
1.0	4.49747E-05	4.64647E-05	8.57457E-08	9.88E-08

5. Conclusion

This paper presents a new approach to solving nonlinear partial differential equations. The method of lines is used to discretize the equation, and the resulting system of ordinary differential equations is solved using the strong stability preserving time-stepping Runge-Kutta (SSP-RK84) method. To evaluate the accuracy and efficiency of the technique, two test examples involving the Fisher equation were examined. The numerical values were compared at various solutions of the parameter ' α ' within the system of ODEs. The norms l_2 and l_∞ were also used to measure the absolute errors of the numerical values. The results of this technique are more accurate and closer to the exact value. This technique can be applied to numerically solve higher-dimensional nonlinear partial differential equations. Future work will focus on extending the proposed method to two- and three-dimensional Fisher-type models, more general nonlinear reaction–diffusion systems, and problems with complex geometries or nonstandard boundary conditions.

References

- [1] Alqudah M. A., Ashraf R., Rashid S., Singh J., Hammouch Z., and Abdeljawad T., Novel numerical investigations of fuzzy cauchy reaction–diffusion models via generalized fuzzy fractional derivative operators, *Fractal and Fractional*, Vol. 5, No. 4 (2021), 151.
- [2] Alshammari B. S. and Mashat D. S., Numerical study of fisher's equation by finite difference schemes, *Applied Mathematics*, Vol. 8, No. 8 (2017), 1100–1116.
- [3] Ammerman A. J. and Cavalli-Sforza L. L., Measuring the rate of spread of early farming in Europe, *Man*, (1971), 674–688.
- [4] Bastani M. and Salkuyeh D. K., A highly accurate method to solve fisher's equation, *Pramana*, Vol. 78 (2012), 335–346.
- [5] Bonkile M., Awasthi A., and Jayaraj S., A numerical implementation of higher order time integration algorithm on unsteady burgers' equation, in *International conference on mathematical modeling and computer simulation*, (2014), 1–10.
- [6] Bramson M. D., Maximal displacement of branching brownian motion, *Communications on Pure and Applied Mathematics*, Vol. 31, No. 5 (1978), 531–581.

- [7] Canosa J., Diffusion in nonlinear multiplicative media, *Journal of Mathematical Physics*, Vol. 10, No. 10 (1969), 1862–1868.
- [8] Chandraker V., Awasthi A., and Jayaraj S., A numerical treatment of fisher equation, *Procedia Engineering*, Vol. 127 (2015), 1256–1262.
- [9] Fisher R. A., The wave of advance of advantageous genes, *Annals of eugenics*, Vol. 7, No. 4 (1937), 355–369.
- [10] Frank-Kamenetskii D., Diffusion and heat transfer in chemical kinetics, 2nd enlarged/revised ed, 1969.
- [11] Gottlieb S., Shu C.-W., and Tadmor E., Strong stability-preserving high-order time discretization methods, *SIAM review*, Vol. 43, No. 1 (2001), 89–112.
- [12] Hamou A. A., Azroul E. H., Hammouch Z., and Alaoui A. L., A monotone iterative technique combined to finite element method for solving reaction-diffusion problems pertaining to non-integer derivative, *Engineering with Computers*, Vol. 39, No. 4 (2023), 2515–2541.
- [13] Hanaç Duruk E., Koksall M. E., and Jiwari R., Analyzing similarity solution of modified fisher equation, *Journal of Mathematics*, Vol. 2022, (2022).
- [14] Hindmarsh A. C., Toward a systematized collection of ode solvers, tech. rep., Lawrence Livermore National Lab., CA (USA), 1982.
- [15] Jiwari R. and Mittal R., A higher order numerical scheme for singularly perturbed burger-huxley equation, *Journal of applied mathematics & informatics*, Vol. 29, No. 3-4 (2011), 813–829.
- [16] Kolmogorov A., Petrovskii I., and Piskunov N., Study of a diffusion equation that is related to the growth of a quality of matter, and its application to a biological problem, *Moscow University Mathematics Bulletin*, Vol. 1 (1937), 1-26.
- [17] Macías-Díaz J. E., Medina-Ramírez I., and Puri A., Numerical treatment of the spherically symmetric solutions of a generalized fisher–kolmogorov–petrovsky–piscounov equation, *Journal of computational and applied mathematics*, Vol. 231, No. 2 (2009), 851–868.

- [18] Mickens R. E., A best finite-difference scheme for the fisher equation, Numerical Methods for Partial Differential Equations, Vol. 10, No. 5 (1994), 581–585.
- [19] Mittal R. and Jiware R., Numerical study of fisher's equation by using differential quadrature method, Int. J. Inf. Syst. Sci, Vol. 5, No. 1 (2009), 143–160.
- [20] Mittal R. and Kumar S., Numerical study of fisher's equation by wavelet galerkin method, International Journal of Computer Mathematics, Vol. 83, No. 3 (2006), 287–298.
- [21] Oymak O. and Selçuk N., Method-of-lines solution of time-dependent two-dimensional navier-stokes equations, International journal for numerical methods in fluids, Vol. 23, No. 5 (1996), 455–466.
- [22] Parambu R. A. V., Awasthi A., Vimal V., and Jha N., A numerical implementation of higher-order time integration method for the transient heat conduction equation with a moving boundary based on boundary immobilization technique, in AIP Conference Proceedings, Vol. 2336, AIP Publishing, 2021.
- [23] Qiu Y. and Sloan D., Numerical solution of fisher's equation using a moving mesh method, Journal of Computational Physics, Vol. 146, No. 2 (1998), 726–746.
- [24] Rothe E., Zweidimensionale parabolische randwertaufgaben als grenzfall eindimensionaler randwertaufgaben, Mathematische Annalen, Vol. 102, No. 1 (1930), 650–670.
- [25] Shah N. A., An analytical view of fractional-order fisher's type equations within caputo operator, Mathematical Problems in Engineering, Vol. 2021 (2021), 1–10.
- [26] Shampine L., Ode solvers and the method of lines, Numerical Methods for Partial Differential Equations, Vol. 10, No. 6 (1994), 739–755.
- [27] Shu C.-W., Total-variation-diminishing time discretizations, SIAM Journal on Scientific and Statistical Computing, Vol. 9, No. 6 (1988), 1073–1084.

- [28] Spiteri R. J. and Ruuth S. J., Non-linear evolution using optimal fourth-order strong-stability-preserving runge-kutta methods, *Mathematics and Computers in Simulation*, Vol. 62, No. 1-2 (2003), 125–135.
- [29] Spiteri R. J. and Ruuth S. J., A new class of optimal high-order strongstability-preserving time discretization methods, *SIAM Journal on Numerical Analysis*, Vol. 40, No. 2 (2002), 469–491.
- [30] Tang S. and Weber R., Numerical study of fisher’s equation by a petrov-galerkin finite element method, *The ANZIAM Journal*, Vol. 33, No. 1 (1991), 27–38.
- [31] Tyson J. J. and Brazhnik P. K., On traveling wave solutions of fisher’s equation in two spatial dimensions, *SIAM Journal on Applied Mathematics*, Vol. 60, No. 2 (2000), 371–391.
- [32] Verma A., Jiwari R., and Koksai M. E., Analytic and numerical solutions of nonlinear diffusion equations via symmetry reductions, *Advances in Difference Equations*, Vol. 2014 (2014), 1–13.
- [33] Vimal V., Sinha R. K., and Pannikkal L., (si13-09) numerical methods for solving nonlinear fisher equation using backward differentiation formula, *Applications and Applied Mathematics: An International Journal (AAM)*, Vol. 19, No. 4 (2024), 4.
- [34] Vimal V., Sinha R. K., and Liju P., An unconditionally stable numerical scheme for solving nonlinear fisher equation, *Nonlinear Engineering*, Vol. 13, No. 1 (2024), 20240006.
- [35] Vimal V., Kumari R., and Sinha R. K., Higher-order numerical technique based on strong stability preserving method for solving nonlinear fisher equation, (2024).
- [36] Wang X., Exact and explicit solitary wave solutions for the generalised fisher equation, *Physics letters A*, Vol. 131, No. 4-5 (1988), 277–279.

This page intentionally left blank.