Communications in Mathematics and Applications
Vol. 15, No. 3, pp. 997–1010, 2024
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v15i3.2806

http://www.rgnpublications.com

Research Article

Higher-Order Numerical Technique Based on Strong Stability Preserving Method for Solving Nonlinear Fisher Equation

Vikash Vimal*[1], Richa Kumari[2] and Rajesh Kumar Sinha[1]

[1] *Department of Mathematics, National Institute of Technology, Patna 800005, India*
[2] *Department of Physics, Patna University, Ashok Rajpath, Patna 800005, India*
***Corresponding author:** vikashvimalnitjsr@gmail.com

Received: July 4, 2024 **Accepted:** October 8, 2024

Abstract. This paper presents higher-order numerical methods for solving nonlinear Fisher equations. These types of equations arise in various fields of sciences and engineering, the main application of this equation has been found in the biomedical sciences. The solution of this equation helps to 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 three and four. 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 steps 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. Fisher's problems, Method of lines, Finite difference methods, Strong stability preserving Runge-Kutta methods

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

1. Introduction

We introduced the one-dimensional nonlinear Fisher equation

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

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

The Fisher equation was initially formulated in 1937, and eq. (1.1) is referenced from [10]. The details of the analysis and explanation of eq. (1.1) can be found in Kolmogorov *et al.* [16]. As a result, it is termed the Fisher-Kolmogorov-Petrovsky-Piscounov (Fisher-KPP) equation, although it is more commonly recognized as the Fisher equation. The first solutions for eq. (1.1) were found by Ablowitz and Zeppetella [2]. After that, eq. (1.1) was expressed multiple times in various form by Korpusov *et al.* [17], Kudryashov [18], and Polyanin and Zhurov [25]. This equation finds numerous uses in the fields of science and engineering (see, Baker [4], Chandraker *et al.* [8], Dattoli *et al.* [9], Gunzburger *et al.* [11], Ma and Fuchssteiner [20], Polyanin and Zhurov [25]). The researchers explored some important variations of this equation (see, Abd-Elhameed *et al.* [1], Agbavon *et al.* [3], Bastani and Salkuyeh [5], Branco *et al.* [7], Hussen and Mebrate [13], Jiwrai and Mittal [14], Macías-Díaz *et al.* [21], Mittal and Jiwari [22], Verma *et al.* [29], Vimal *et al.* [30, 31], and Wang [34]). The literature contains detailed information regarding the mathematical properties of Fisher's equation, along with extensive discussions (see, Kawahara and Tanaka [15], Larson [19], and Tyson and Brazhnik [28]).

In mathematical science, we develop a method for obtaining numerical solutions of a one-dimensional nonlinear reaction-diffusion equation using the SSPRK-43 technique. We perform this task by transforming the PDE into ODE over time through the application of the method of lines (Oymak and Selçuk [23]). The method of lines, which is a technique for finding numerical solutions of PDE, plays an essential role in preserving the accuracy and stability of the developing solution. The ODEs 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 [12]).

In this paper, we introduce a numerical approach for solving Fisher's equation. We use the method of lines in the spatial domain and employ the *strong stability preserving Runge-Kutta* (SSPRK) method in the time domain for Fisher's equation. In Section 2, we illustrate Fisher's equation in one dimension, including 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-RK43 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 is given.

2. Problem Statement

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

with initial condition

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

and the boundary conditions

$$u(0,t) = f_3(t), \quad 0 \le t \le T,$$

$$u(1,t) = f_4(t), \quad 0 \le t \le T,$$

where α is the reactive factor and $u_0(x)$, f_3, and f_4, are given functions of the variables which are sufficiently smooth.

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. Then the set of ODEs is solved by applying the SSP-RK43 scheme for integration. To discretize the solution domain for eq. (1.1), 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 Method of Lines (MOL)

In [26], Rothe first introduced the *Method of Lines* (MOL), and in subsequent works of Bonkile *et al.* [6], and Parambu *et al.* [24] utilized the MOL to transform the PDEs into a set of ODEs, effectively addressing the Burger's equation and Stefan problem. An unsteady linear 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 the central difference to discretize the diffusion term $\frac{\partial^2 u}{\partial x^2}$,

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

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

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

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

The right-hand side of eq. (3.3) can be expressed using a discrete operator form:

$$\frac{du_m}{dt} = L(u_m), \tag{3.4}$$

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

3.2 Time Integration

In this method, we establish a system of ordinary differential equations that are solved with a four-stage, third-order time-stepping Runge-Kutta (SSP-RK43) scheme designed to ensure strong stability preservation. In this case, the SSP method used exhibits the property where the number of stages ($s = 4$) exceeds the method's order ($k = 3$).

3.2.1 SSP-RK43

The purpose of SSP-RK43 is to achieve high-order accuracy in time integration while maintaining strong stability properties. Let us express an s-stage explicit Runge-Kutta method in the following manner,

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

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

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

The SSP-RK43 scheme is characterized by coefficients $a_{i,k}$ that satisfy the conditions $a_{i,k} \geq 0$ and $a_{i,k} = 0$ only if $b_{i,k} = 0$ (Shu [27]). This scheme also possesses a *Courant-Friedrichs-Lewy* (CFL) coefficient is 2. Additionally, it's required that the sum of coefficients $\sum_{k=0}^{i-1} a_{i,k} = 1$ holds for $i = 1,2,3,\ldots,s$.

To descretize the temporal domain $[0,T]$ into N equivalent sub-intervals with a uniform mesh size, we assume $\Delta t = 1/N = k$ and utilize $t_n = n\Delta t$. We perform the integration of eq. (3.4) from t_n to $t_n + \Delta t$ using the following steps for $n = 0,1,2\ldots,N$, 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.

Table 1. Butcher tableau of SSP-RK43 scheme (Shu [27])

$a_{i,k}$				$b_{i,k}$			
1				$\frac{1}{2}$			
0	1			0	$\frac{1}{2}$		
$\frac{2}{3}$	0	$\frac{1}{3}$		0	0	$\frac{1}{6}$	
0	0	0	1	0	0	0	$\frac{1}{2}$

$$\left.\begin{aligned} u_m^{(0)} &= u_m^n \\ u_m^{(0)} &= u_m^n \end{aligned}\right\}, \tag{3.8}$$

$$\left.\begin{aligned} u_m^{(1)} &= u_m^{(0)} + \Delta t\left(\frac{1}{2}\right)L(u_m^{(0)}) \\ u_m^{(1)} &= u_m^{(0)} + \left(\frac{k}{2}\right)\left[\frac{u_{m+1}^{(0)} - 2u_m^{(0)} + u_{m-1}^{(0)}}{h^2} + \alpha u_m^{(0)}(1 - u_m^{(0)})\right] \end{aligned}\right\}, \tag{3.9}$$

$$\left.\begin{aligned} u_m^{(2)} &= u_m^{(1)} + \Delta t\left(\frac{1}{2}\right)L(u_m^{(1)}) \\ u_m^{(2)} &= u_m^{(1)} + \left(\frac{k}{2}\right)\left[\frac{u_{m+1}^{(1)} - 2u_m^{(1)} + u_{m-1}^{(1)}}{h^2} + \alpha u_m^{(1)}(1 - u_m^{(1)})\right] \end{aligned}\right\}, \tag{3.10}$$

$$\left.\begin{aligned} u_m^{(3)} &= \left(\frac{2}{3}\right)u_m^{(0)} + \left(\frac{1}{3}\right)u_m^{(2)} + \Delta t\left(\frac{1}{6}\right)L(u_m^{(2)}) \\ u_m^{(3)} &= \left(\frac{2}{3}\right)u_m^{(0)} + \left(\frac{1}{3}\right)u_m^{(2)} + \left(\frac{k}{6}\right)\left[\frac{u_{m+1}^{(2)} - 2u_m^{(2)} + u_{m-1}^{(2)}}{h^2} + \alpha u_m^{(2)}(1 - u_m^{(2)})\right] \end{aligned}\right\}, \tag{3.11}$$

$$\left.\begin{aligned} u_m^{(n+1)} &= u_m^{(3)} + \Delta t\left(\frac{1}{2}\right)L(u_m^{(3)}) \\ u_m^{(n+1)} &= u_m^{(0)} + \left(\frac{k}{2}\right)\left[\frac{u_{m+1}^{(3)} - 2u_m^{(3)} + u_{m-1}^{(3)}}{h^2} + \alpha u_m^{(3)}(1 - u_m^{(3)})\right] \end{aligned}\right\}, \tag{3.12}$$

where $\Delta t = k$, $m = 2,3,\ldots,M$ for the next iteration $u_m^{(0)} = u_m^{n+1}$.

4. Numerical Results

Numerical results for the Fisher equation (1.1) using the SSP-RK43 method in MATLAB are presented, and the method's accuracy is assessed by comparing it with the exact solution. Assess the accuracy and efficiency of the proposed method by evaluating the L_2 and L_∞ error norms,

$$L_2 = \left[\frac{1}{M}\sum_{m=0}^{M}(U_m - u_m)^2\right]^{\frac{1}{2}}, \quad L_\infty = \max_{0\le m\le M}|U_m - u_m|.$$

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 [22] 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 Fisher equation

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

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 [22] by

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

Tables 2 and 3 present numerical and exact solutions at various time points for different values of 'α' and N is number of iteration in examples one and two. We compare the numerical and exact solutions for $\alpha = 6$ and 1 using at two distinct time steps, namely $\Delta t = 0.00001$ and $\Delta t = 0.000005$, with a specific focus on Examples 4.1 and 4.2. Figures 1, 2, 3, and 4 represent numerical solutions compared to the exact solution, while Figures 5, 6, 7, and 8 display the absolute error graphs about the exact solution for $\alpha = 6$ and $\alpha = 1$, with Δt values of 0.00001 and 0.000005. We have conducted a comparison with existing numerical methods with [13, 22, 30, 32]. Our proposed numerical approaches, given in Tables 4, 5, 6, , 7, 8 and 9, perform better than those presented in references [22, 30, 32]. Tables 10, 11, 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.

Table 2. Numerical and exact solution at $\alpha = 6$, $N = 20$ for Example 4.1

		$\Delta t = 0.00001$		$\Delta t = 0.000005$	
x	T	Numerical solution	Exact solution	Numerical solution	Exact solution
0.25	0.5	0.818397	0.818393	0.818396	0.818393
	1.0	0.982919	0.982919	0.982919	0.982919
	2.0	0.999883	0.999883	0.999883	0.999883
	5.0	1.000000	1.000000	1.000000	1.000000
0.5	0.5	0.775809	0.775803	0.775808	0.775803
	1.0	0.978147	0.978147	0.978147	0.978147
	2.0	0.999850	0.999850	0.999850	0.999850
	5.0	1.000000	1.000000	1.000000	1.000000
0.75	0.5	0.725830	0.725824	0.725828	0.725824
	1.0	0.972071	0.972071	0.972071	0.972071
	2.0	0.999808	0.999808	0.999808	0.999808
	5.0	1.000000	1.000000	1.000000	1.000000

Table 3. Numerical and exact solution at $\alpha = 1$, $N = 20$ for Example 4.2

		$\Delta t = 0.00001$		$\Delta t = 0.000005$	
x	T	Numerical solution	Exact solution	Numerical solution	Exact solution
0.25	0.5	0.334095	0.334094	0.334094	0.334094
	1.0	0.455739	0.455739	0.455739	0.455739
	2.0	0.683951	0.683951	0.683951	0.683951
	5.0	0.966525	0.966525	0.966525	0.966525
0.5	0.5	0.305739	0.305739	0.305739	0.305739
	1.0	0.425509	0.425509	0.425509	0.425509
	2.0	0.659217	0.659216	0.659217	0.659216
	5.0	0.963028	0.963028	0.963028	0.963028
0.75	0.5	0.278354	0.278353	0.278354	0.278353
	1.0	0.395412	0.395411	0.395412	0.395411
	2.0	0.633359	0.633358	0.633358	0.633358
	5.0	0.959178	0.959178	0.959178	0.959178

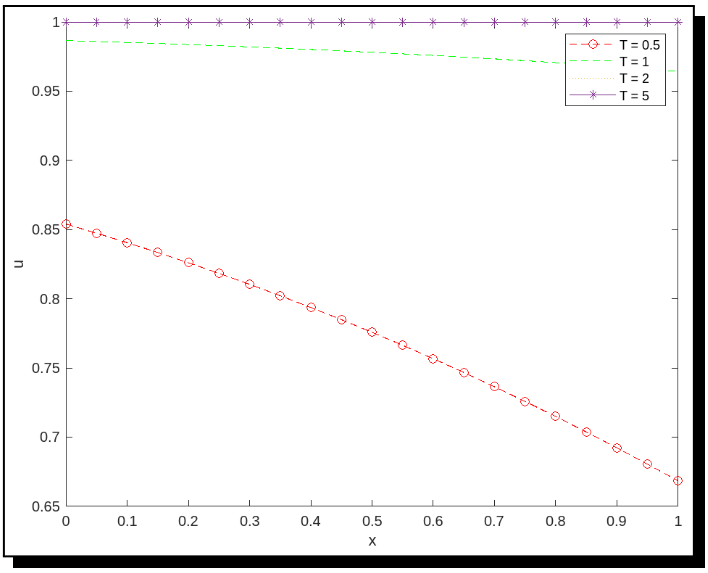

Figure 1. Solution at $\alpha = 6$, $\Delta t = 0.00001$, and $N = 20$ for Example 4.1

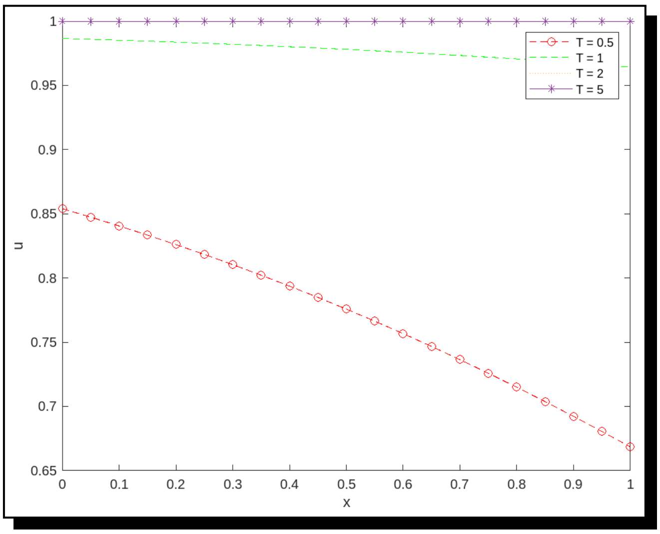

Figure 2. Solution at $\alpha = 6$, $\Delta t = 0.000005$, and $N = 20$ for Example 4.1

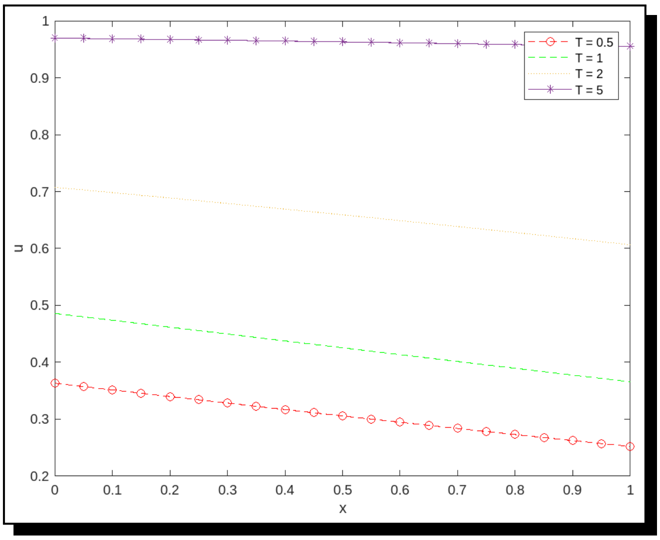

Figure 3. Solution at $\alpha = 1$, $\Delta t = 0.00001$, and $N = 20$ for Example 4.2

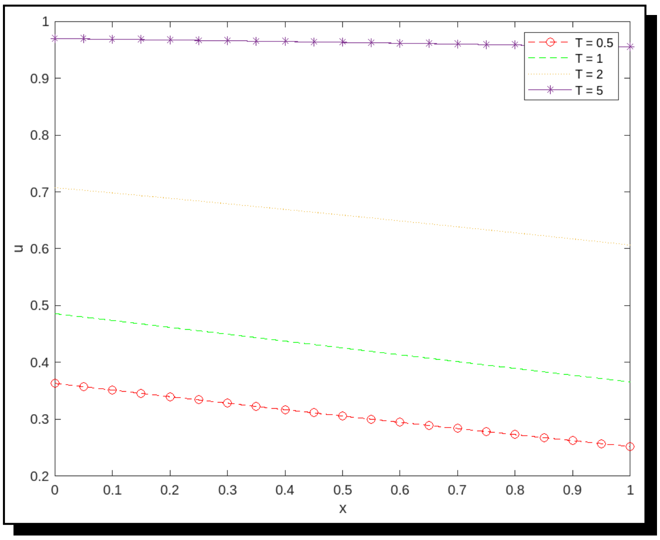

Figure 4. Solution at $\alpha = 1$, $\Delta t = 0.000005$, and $N = 20$ for Example 4.2

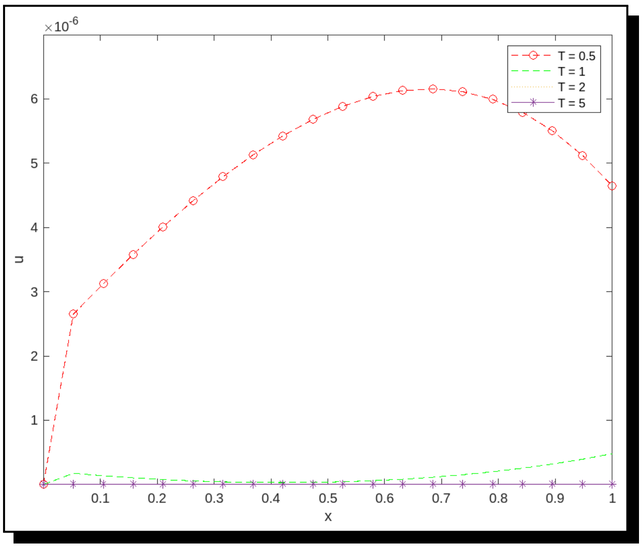

Figure 5. Absolute errors at $\alpha = 6$, $\Delta t = 0.00001$, and $N = 20$ for Example 4.1

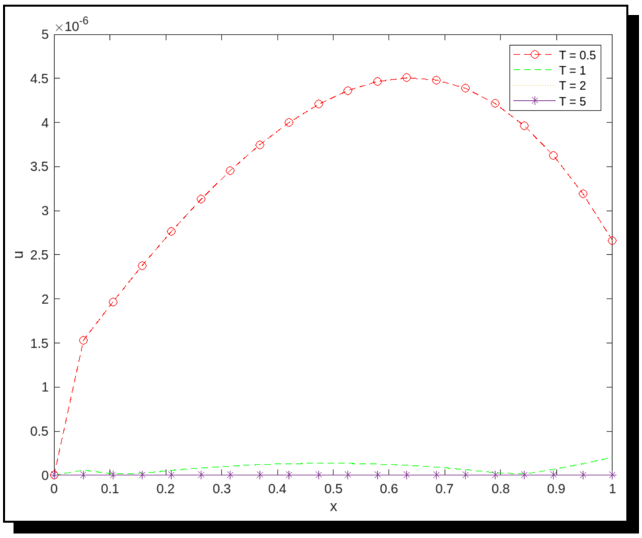

Figure 6. Absolute errors at $\alpha = 6$, $\Delta t = 0.000005$, and $N = 20$ for Example 4.1

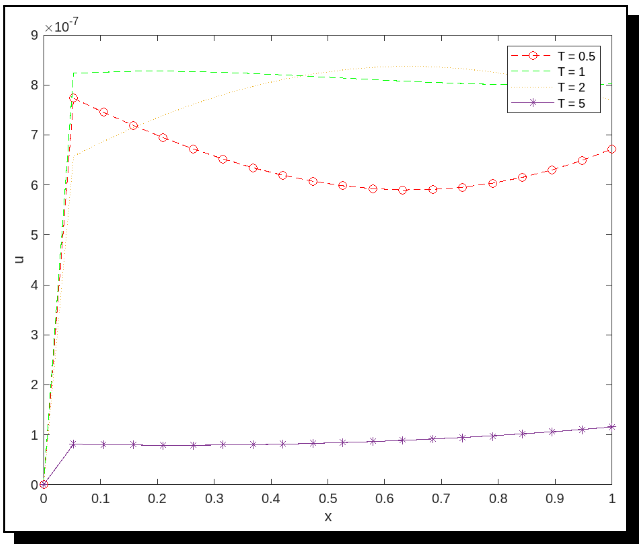

Figure 7. Absolute errors at $\alpha = 1$, $\Delta t = 0.00001$, and $N = 20$ for Example 4.2

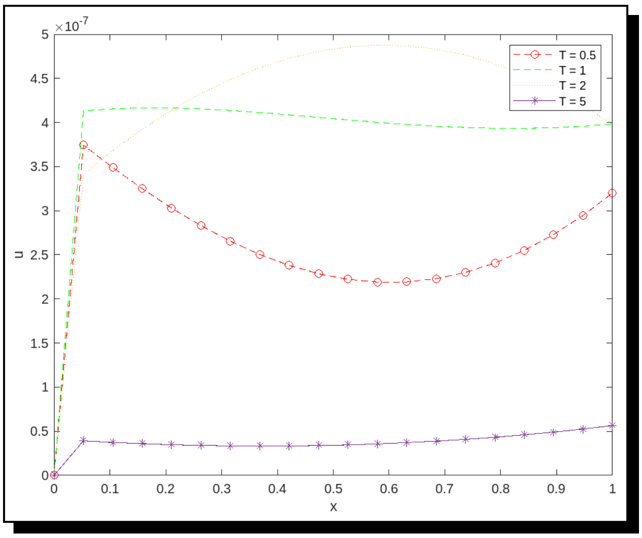

Figure 8. Absolute errors at $\alpha = 1$, $\Delta t = 0.000005$, and $N = 20$ for Example 4.2

Table 4. Present and existing numerical solution comparison with exact solution at $\alpha = 6$, $N = 20$ for Example 4.1

		$\Delta t = 0.0001$			$\Delta t = 0.00005$		
x	T	Mittal [22]	Present solution	Exact solution	Mittal [22]	Present solution	Exact solution
0.25	0.5	0.81847	0.818420	0.818393	0.81843	0.818407	0.818393
	1.0	0.98293	0.982921	0.982919	0.98292	0.982920	0.982919
	2.0	0.99988	0.999883	0.999883	0.99988	0.999883	0.999883
	5.0	1.00000	1.000000	1.000000	1.00000	1.000000	1.000000
0.5	0.5	0.77590	0.775837	0.775803	0.77580	0.775822	0.775803
	1.0	0.97816	0.978150	0.978147	0.97815	0.978148	0.978147
	2.0	0.99985	0.999850	0.999850	0.99985	0.999850	0.999850
	5.0	1.00000	1.000000	1.000000	1.00000	1.000000	1.000000
0.75	0.5	0.72594	0.725862	0.725824	0.72588	0.725844	0.725824
	1.0	0.97209	0.972075	0.972071	0.92208	0.972073	0.972071
	2.0	0.99981	0.999808	0.999808	0.99981	0.999808	0.999808
	5.0	1.00000	1.000000	1.000000	1.00000	1.000000	1.000000

Table 5. Present and existing numerical solution comparision with exact solution at $\alpha = 1$, $N = 20$ for Example 4.2

		$\Delta t = 0.0001$			$\Delta t = 0.00005$		
x	T	Mittal [22]	Present solution	Exact solution	Mittal [22]	Present solution	Exact solution
0.25	0.5	0.33412	0.334102	0.334094	0.33409	0.334098	0.334094
	1.0	0.45576	0.455747	0.455739	0.45574	0.455743	0.455739
	2.0	0.68397	0.683957	0.683951	0.68394	0.683954	0.683951
	5.0	0.96653	0.966526	0.966525	0.96653	0.966525	0.966525
0.5	0.5	0.30576	0.305746	0.305739	0.30574	0.305742	0.305739
	1.0	0.42553	0.425517	0.425509	0.42551	0.425513	0.425509
	2.0	0.65924	0.659223	0.659216	0.65921	0.659220	0.659216
	5.0	0.96303	0.963029	0.963028	0.96303	0.963029	0.963028
0.75	0.5	0.27838	0.278361	0.278353	0.27835	0.278357	0.278353
	1.0	0.39544	0.395420	0.395411	0.39542	0.395415	0.395411
	2.0	0.63338	0.633365	0.633358	0.63336	0.633361	0.633358
	5.0	0.95918	0.959179	0.959178	0.95918	0.959179	0.959178

Table 6. Comparison of numerical solution for Example 4.1 at $\Delta t = 0.0004$, $T = 0.4$ and $\alpha = 6$.

x	SIS [13]	Present solution	Exact solution
0	0.77580349	0.77580349	0.77580349
0.1	0.75685967	0.75684613	0.75671127
0.2	0.73668727	0.73656282	0.73641959
0.3	0.71528604	0.71508119	0.71492899
0.4	0.69266980	0.69241517	0.69225459
0.5	0.66886832	0.66859637	0.66842802
0.6	0.64392921	0.64367440	0.64349899
0.7	0.61791942	0.61771822	0.61753662
0.8	0.59092643	0.59081698	0.59063034
0.9	0.56305891	0.56305630	0.56289023
1	0.53444665	0.53444665	0.53444665

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

x	BDF1 [32]	BDF2 [32]	Present solution	Exact solution
0.1	0.35841806	0.35842071	0.35842328	0.35842691
0.2	0.32997086	0.32997260	0.32997524	0.32998421
0.3	0.30230060	0.30230157	0.30230424	0.30231742
0.4	0.27558402	0.27558442	0.27558708	0.27560315
0.5	0.24997987	0.24997993	0.24998256	0.25000000
0.6	0.22562504	0.22562500	0.22562757	0.22564477
0.7	0.20263156	0.20263170	0.20263415	0.20264943
0.8	0.18108475	0.18108534	0.18108759	0.18109917
0.9	0.16104234	0.16104363	0.16104557	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 [30]	BDF2 [30]	Present solution	Exact solution
0.1	0.26073733	0.26073824	0.26073855	0.26073843
0.2	0.25042002	0.25042078	0.25042103	0.25042110
0.3	0.24031064	0.24031127	0.24031149	0.24031169
0.4	0.23041738	0.23041791	0.23041810	0.23041838
0.5	0.22074766	0.22074815	0.22074832	0.22074865
0.6	0.21130823	0.21130872	0.21130889	0.21130920
0.7	0.20210505	0.20210558	0.20210576	0.20210601
0.8	0.19314332	0.19314394	0.19314415	0.19314428
0.9	0.18442751	0.18442824	0.18442848	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 [30]	BDF2 [30]	Present solution	Exact solution
0.1	0.35840890	0.35842119	0.35842328	0.35842691
0.2	0.32996342	0.32997349	0.32997524	0.32998421
0.3	0.30229465	0.30230278	0.30230424	0.30231742
0.4	0.27557923	0.27558586	0.27558708	0.27560315
0.5	0.24997582	0.24998148	0.24998256	0.25000000
0.6	0.22562126	0.22562655	0.22562757	0.22564477
0.7	0.20262760	0.20263311	0.20263415	0.20264943
0.8	0.18108020	0.18108646	0.18108759	0.18109917
0.9	0.16103685	0.16104428	0.16104557	0.16105159

Table 10. Error of Example 4.1 at $\alpha = 1$

	$\Delta t = 0.0001$		$\Delta t = 0.00005$	
T	L_2	L_∞	L_2	L_∞
0.5	2.87954E-05	3.80053E-05	1.54322E-05	2.02224E-05
1.0	2.99375E-06	4.2997E-06	1.38381E-06	2.0213E-06
2.0	1.99398E-08	2.91E-08	8.69856E-09	1.31E-08
5.0	0	0	0	0

Table 11. Error of Example 4.2 at $\alpha = 1$

	$\Delta t = 0.0001$		$\Delta t = 0.00005$	
T	L_2	L_∞	L_2	L_∞
0.5	6.40395E-06	7.6751E-06	3.14702E-06	3.7845E-06
1.0	7.09577E-06	8.2303E-06	3.54533E-06	4.1173E-06
2.0	6.06549E-06	7.2913E-06	3.0837E-06	3.6994E-06
5.0	8.58653E-07	1.0872E-06	4.23784E-07	5.377E-07

Table 12. Error of Example 4.1 at $\alpha = 6$

	$\Delta t = 0.00001$		$\Delta t = 0.000005$	
T	L_2	L_∞	L_2	L_∞
0.5	4.74683E-06	5.9958E-06	3.41395E-06	4.3637E-06
1.0	1.05544E-07	1.986E-07	8.09956E-08	1.355E-07
2.0	5.91608E-10	1E-09	1.5411E-09	2.3E-09
5.0	0	0	0	0

Table 13. Error of Example 4.2 at $\alpha = 1$

	$\Delta t = 0.00001$		$\Delta t = 0.000005$	
T	L_2	L_∞	L_2	L_∞
0.5	5.41759E-07	6.721E-07	2.16534E-07	2.831E-07
1.0	7.04974E-07	8.268E-07	3.49989E-07	4.155E-07
2.0	6.98402E-07	8.301E-07	4.00322E-07	4.857E-07
5.0	7.58889E-08	9.81E-08	3.24257E-08	4.32E-08

5. Conclusion

We have developed a high-order numerical method for solving the Fisher's equation. We implement a technique known as semi-discretization to solve PDEs in the spatial variable using the *Method of Lines* (MOL). Using this approach, we derive a system of ODEs that are solved using the SSP-RK43 scheme. We consider two examples of the Fisher equation, comparing numerical solutions at various time steps for different values of 'α' to determine the efficiency and accuracy of the method. The L_2 and L_∞ norms are evaluated for numerical errors. The introduced method produces better results, approaching the exact solution numerically. The results are also compared with a few existing methods in Tables 4, 5, 6, 7, 8 and 9, they are found to be more precise and accurate. This technique can also be implemented to solve higher-dimensional nonlinear *Partial Differential Equations* (PDEs).

Acknowledgment

The authors would like to thank the editor and the anonymous referees for the helpful suggestions to improve the quality of the manuscript.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] W. M. Abd-Elhameed, A. Ali and Y. Youssri, Newfangled linearization formula of certain nonsymmetric Jacobi polynomials: Numerical treatment of nonlinear Fisher's equation, *Journal of Function Spaces* **2023**(1) (2023), 6833404, DOI: 10.1155/2023/6833404.

[2] M. J. Ablowitz and A. Zeppetella, Explicit solutions of Fisher's equation for a special wave speed, *Bulletin of Mathematical Biology* **41** (1979), 835 – 840, DOI: 10.1007/BF02462380.

[3] K. M. Agbavon, A. R. Appadu, B. Inan and H. M. Tenkam, Convergence analysis and approximate optimal temporal step sizes for some finite difference methods discretising Fisher's equation, *Frontiers in Applied Mathematics and Statistics* **8** (2022), 921170, DOI: 10.3389/fams.2022.921170.

[4] R. E. Baker, *Mathematical Biology and Ecology Lecture Notes*, University of Oxford, Oxford, UK, 92 pages (2011), https://www.math.mun.ca/~zhao/ARRMSschool/MathBioNotes2011.pdf.

[5] M. Bastani and D. K. Salkuyeh, A highly accurate method to solve Fisher's equation, *Pramana* **78** (2012), 335 – 346, DOI: 10.1007/s12043-011-0243-8.

[6] M. Bonkile, A. Awasthi, and S. Jayaraj, A numerical implementation of higher order time integration algorithm on unsteady burgers' equation, in: *International Conference on Mathematical Modeling and Computer Simulation - 2014*, pp. 1 – 10 (2014).

[7] J. R. Branco, J. A. Ferreira and P. de Oliveira, Numerical methods for the generalized Fisher-Kolmogorov-Petrovskii-Piskunov equation, *Applied Numerical Mathematics* **57**(1) (2007), 89 – 102, DOI: 10.1016/j.apnum.2006.01.002.

[8] V. Chandraker, A. Awasthi and S. Jayaraj, A numerical treatment of Fisher equation, *Procedia Engineering* **127** (2015), 1256 – 1262, DOI: 10.1016/j.proeng.2015.11.481.

[9] G. Dattoli, E. Di Palma, E. Sabia and S. Licciardi, Quasi exact solution of the Fisher equation, *Applied Mathematics* **4** (2013), 7 – 12, DOI: 10.4236/am.2013.48A002.

[10] R. A. Fisher, The wave of advance of advantageous genes, *Annals of Eugenics* **7**(4) (1937), 355 – 369, DOI: 10.1111/j.1469-1809.1937.tb02153.x.

[11] M. Gunzburger, L. Hou and W. Zhu, Modeling and analysis of the forced Fisher equation, *Nonlinear Analysis: Theory, Methods & Applications* **62**(1) (2005), 19 – 40, DOI: 10.1016/j.na.2005.01.094.

[12] A. C. Hindmarsh, Toward a systematized collection of ODE solvers, in: *International Mathematics and Computers Simulation Congress on Systems Simulation and Scientific Computation*, Montreal, Canada, 9 August 1982, Technical Report, OSTI ID: 5180574, Lawrence Livermore National Labratory, CA (USA), 1982, https://www.osti.gov/biblio/5180574.

[13] I. Hussen and B. Mebrate, Semi implicit scheme of Fisher equation based on Crank-Nicolson method and method of lagging, *Palestine Journal of Mathematics* **11**(3) (2022), 536 – 548, https://pjm.ppu.edu/sites/default/files/papers/PJM_May_%283%292022_536_to_548.pdf.

[14] R. Jiwrai and R. Mittal, A higher order numerical scheme for singularly perturbed Burger-Huxley equation, *Journal of Applied Mathematics and Informatics* **29**(3-4) (2011), 813 – 829, https://koreascience.kr/article/JAKO201121559392442.pdf.

[15] T. Kawahara and M. Tanaka, Interactions of traveling fronts: An exact solution of a nonlinear diffusion equation, *Physics Letters A* **97**(8) (1983), 311 – 314, DOI: 10.1016/0375-9601(83)90648-5.

[16] A. K. Kolmogorov, N. Petrovsky and S. Piscounov, Etude de I equations de la diffusion avec croissance de la quantitate de matiere et son application a un probolome biologique, *Bull. Univ. Mosku* **1** (1937), 1 – 25.

[17] M. O. Korpusov, A. V. Ovchinnikov and A. V. Sveshnikov, On blow up of generalized Kolmogorov-Petrovskii-Piskunov equation, *Nonlinear Analysis: Theory, Methods & Applications* **71**(11) (2009), 5724 – 5732, DOI: 10.1016/j.na.2009.05.002.

[18] N. A. Kudryashov, Exact solitary waves of the Fisher equation, *Physics Letters A* **342**(1-2) (2005), 99 – 106, DOI: 10.1016/j.physleta.2005.05.025.

[19] D. A. Larson, Transient bounds and time-asymptotic behavior of solutions to nonlinear equations of Fisher type, *SIAM Journal on Applied Mathematics* **34**(1) (1978), 93 – 104, DOI: 10.1137/0134008.

[20] W. X. Ma and B. Fuchssteiner, Explicit and exact solutions to a Kolmogorov-Petrovskii-Piskunov equation, *International Journal of Non-Linear Mechanics* **31**(3) (1996), 329 – 338, DOI: 10.1016/0020-7462(95)00064-X.

[21] J. E. Macías-Díaz, I. Medina-Ramírez and A. Puri, Numerical treatment of the spherically symmetric solutions of a generalized Fisher–Kolmogorov–Petrovsky–Piscounov equation, *Journal of Computational and Applied Mathematics* **231**(2) (2009), 851 – 868, DOI: 10.1016/j.cam.2009.05.008.

[22] R. C.Mittal and R. Jiwari, Numerical study of Fisher's equation by using differential quadrature method, *International Journal Information and Systems Science* **5**(1) (2009), 143 – 160.

[23] O. Oymak and N. Selçuk, Method-of-lines solution of time-dependent two-dimensional Navier-Stokes equations, *International Journal for Numerical Methods in Fluids* **23**(5) (1996), 455 – 466, DOI: 10.1002/(sici)1097-0363(19960915)23:5<455::aid-fld435>3.0.co;2-j.

[24] R. A. V. Parambu, A. Awasthi, V. Vimal and N. Jha, A numerical implementation of higher-order time integration method for the transient heat conduction equation with a moving boundary based on boundary immobilization technique, *AIP Conference Proceedings* **2336**(1) (2021), 030012, DOI: 10.1063/5.0045874.

[25] A. D. Polyanin and A. I. Zhurov, A new method for constructing exact solutions to nonlinear delay partial differential equations, *arXiv preprint*, arXiv:1304.5473 (2013), DOI: 10.48550/arXiv.1304.5473.

[26] E. Rothe, Zweidimensionale parabolische Randwertaufgaben als Grenzfall eindimensionaler Randwertaufgaben, *Mathematische Annalen* **102**(1930), 650 – 670, DOI: 10.1007/bf01782368.

[27] C.-W. Shu, Total-variation-diminishing time discretizations, *SIAM Journal on Scientific and Statistical Computing* **9**(6) (1988), 1073 – 1084, DOI: 10.1137/0909073.

[28] J. J. Tyson and P. K. Brazhnik, On traveling wave solutions of Fisher's equation in two spatial dimensions, *SIAM Journal on Applied Mathematics* **60**(2) (2000), 371 – 391, DOI: 10.1137/s0036139997325497.

[29] A. Verma, R. Jiwari and M. E. Koksal, Analytic and numerical solutions of nonlinear diffusion equations via symmetry reductions, *Advances in Difference Equations* **2014** (2014), Article number: 229, DOI: 10.1186/1687-1847-2014-229.

[30] V. Vimal, R. K. Sinha and P. Liju, An unconditionally stable numerical scheme for solving nonlinear Fisher equation, *Nonlinear Engineering* **13**(1) (2024), 20240006, DOI: 10.1515/nleng-2024-0006.

[31] V. Vimal, R. K. Sinha and P. Liju, Backward differentiation formula based numerical method to Solve Fisher equation, *South East Asian Journal of Mathematics and Mathematical Sciences* **20**(2) (2024), 477 – 494, DOI: 10.56827/SEAJMMS.2024.2002.34.

[32] V. Vimal, R. K. Sinha and L. Pannikkal, Numerical methods for solving nonlinear fisher equation using backward differentiation formula, *Applications and Applied Mathematics: An International Journal* **19**(4) (2024), Article 4, URL: https://digitalcommons.pvamu.edu/aam/vol19/iss4/4.

[33] N. K. Vitanov, I. P. Jordanov and Z. I. Dimitrova, On nonlinear population waves, *Applied Mathematics and Computation* **215**(8) (2009), 2950 – 2964, DOI: 10.1016/j.amc.2009.09.041.

[34] X. Y. Wang, Exact and explicit solitary wave solutions for the generalised fisher equation, *Physics letters A* **131**(4-5) (1988), 277 – 279, DOI: 10.1016/0375-9601(88)90027-8.