International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

Method Of Lines For Nonlinear Redaction-Diffusion Equations

Huda Hammad Mohammed[1], Abdelsalam Abuzreda[2]

[1]Assissant Lecturer, at The Higher Institute of Engineering Techniques, Benghazi, Libya
hudaalshaari300@gmail.com
[2]Associate Professor & Postdoctoral Research fellow, Senior Advisor Department of Health Safety and Environmental (HSE), Arabian Gulf Oil Company (AGOCO) and University of Benghazi, and the higher Institute of Engineering Techniques Benghazi, Libya.

Corresponding Author: Abdelsalam Abuzreda. E-mail: bozrida@yahoo.com

(cc) BY

Abstract: Method of lines is used to solve nonlinear redaction-diffusion equations. The study focus three partial differential equations, Burger's Equation, Fisher's Equation and Burgers-Fisher Equation. This method is transforming the nonlinear partial differential equations to systems of nonlinear ordinary differential equations. Some numerical examples are presented to show the efficiency of considered method and compared this method with a previous study. In addition, the graphical represented the solutions, which had been given by MATLAB program.

Keywords: finite differences approximations, method of lines, partial differential equations, Rung Kutta method.

1. INTRODUCTION

Diffusion phenomena is one the most important topic in heat transfer, especially in Mechanics Engineering, so the nonlinear phenomena of this topic to obtain the exact solution is very difficult and sometime impossible and some methods of previous studies is difficult. In this work, I study the nonlinear revolution equations where each equation contains the dissipative term u_{xx} in addition to other partial derivatives. This new family of nonlinear equations obtained from scientific applications and physical phenomena. In this paper, using method of lines for solving nonlinear reduction–diffusion equations; **The objective** of this research is comparison this method with previous studies that is discuss nonlinear reduction–diffusion equations the methods are explicit method (EM), exponential method (EXPM) and Dufort- Frankel method (DFM) [1]. The new family of nonlinear equations that will be discuss in this paper is the form

$$\frac{\partial u}{\partial t} - v\frac{\partial^2 u}{\partial x^2} = p(u), \qquad (1)$$

where $u(x,t)$ is a function of space x and time t, v is constant viscosity and $p(u)$ is nonlinear term [1]. If:

i. $p(u) = -uu_x$, obtain the Burgers equation.
ii. $p(u) = ku(1-\frac{u}{k})$, obtain the Fisher's equation.
iii. $p(u) = uu_x + ku(1-\frac{u}{k})$, obtain the Burgers-Fisher equation.
iv. $p(u) = 0$, obtain the linear diffusion equation.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

1.1 Burgers Equation

The Burgers equation is a balance between time evolution, nonlinearity and diffusion. This is the simple nonlinear model equation for diffusive wave in fluid dynamics. Burgers (1948) first developed this equation primarily to shed light on the study of turbulence described by the interaction of the two opposite effects of convection and diffusion [1]. The standard form of Burgers equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2} = 0 \quad , v > 0 \qquad (2)$$

where v is a constant that defines the kinematic viscosity, if $v = 0$ the equation is called inviscid Burgers equation [2].

1.2 Fisher's Equation

Fisher (1936) first introduced a nonlinear evolution equation to investigate the wave propagation of an advantageous gene a population. His equation also describes the logistic growth- diffusion process and has the form.

$$\frac{\partial u}{\partial t} - v\frac{\partial^2 u}{\partial x^2} = ku\left(1 - \frac{u}{k}\right) \quad , v > 0 \quad , k > 0 \quad (3)$$

where v is a diffusion constant, k is the linear growth rate and carrying capacity of the environment. The term $ku\left(1 - \frac{u}{k}\right)$ represent a nonlinear growth rate [2].

2. METHOD OF LINES

The main idea of the method of lines is to transform the partial differential equations to systems of ordinary differential equations. We use finite differences approximations, finite elements approximations and finite volume approximations [3]. In this paper, we using finite differences and central operator difference to approximate the derivatives.

2.1 Finite Difference Method

The finite differences approximations for derivatives are one of the simplest and of the oldest methods to solve differential equations. L. Euler knew it, in this paper, we using the central operator difference to approximate the derivatives [3]. The first and second derivatives defined by

$$\frac{\partial u(x_i, t)}{\partial x} = \frac{u(x_{i+1}, t) - u(x_{i-1}, t)}{2(\Delta x)} + O((\Delta x)^2) \ (4)$$

$$\frac{\partial^2 u(x_i, t)}{\partial x^2} = \frac{u(x_{i+1}, t) - 2u(x_i, t) + u(x_{i-1}, t)}{(\Delta x)^2} + \ O((\Delta x)^2) \qquad (5)$$

2.2 Runge Kutta Method

The Runge Kutta Method is numerical method using to solve ordinary differential equations, in this paper, we use Runge Kutta method in order four (RK4), which is denoted by

$$u_{i+1,j} = u_{i,j} + \frac{1}{6}\left(k_{1,j} + 2k_{2,j} + 2k_{3,j} + k_{4,j}\right) \ (7)$$

where

$$k_{1,j} = hf(t_i, u_{1,j}, u_{2j}, \dots, u_{nj})$$

$$k_{2,j} = hf\left(t_i + \frac{h}{2}, u_{1,j} + \frac{k_{1,j}}{2}, \dots, u_{n,j} + \frac{k_{1,j}}{2}\right)$$

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

$$k_{3,j} = hf(t_i + \frac{h}{2}, u_{1,j} + \frac{k_{2,j}}{2}, \dots, u_{n,j} + \frac{k_{2,j}}{2})$$

$$k_{4,j} = hf(t_i + h, u_{1,j} + k_{3,j}, \dots, u_{n,j} + k_{3,j})$$

$$h = \frac{t_{max} - t_{min}}{n}$$

for j = 1,2,…,m & i = 1,2,2…,n

Example 1. Method of lines of Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2} = 0 \quad , v > 0 \quad (8)$$

BCs: $u(0,t) = \alpha(t)$ & $u(L,t) = \beta(t)\,, t > 0$

IC: $u(x,0) = f(x)$, $0 < x < L$

Applied method of lines we obtain

$$\frac{du_i}{dt} = u_i\left(\frac{u_{i-1} - u_{i+1}}{2(\Delta x)}\right) + v\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}\right) \quad (9)$$

for i=1,….,n-1 we obtain nonlinear systems of ordinary differential equations.

Algorithm of Burgers Equation

To obtain the numerical solution of Burgers equation.

Input: endpoint L; maximum time T; constant of viscosity v; integer n & m

Output: approximations $u(x_i, t_j)$

Step 1: Δx =L/n

h=Tmax/m

Step 2: when i=0

$u(x_0, t) = \alpha(t)$

when i=n

$u(x_n, t) = \beta(t)$

Step 3: for i=1,…..,n-1

$$\frac{du_i}{dt} = u_i\left(\frac{u_{i-1} - u_{i+1}}{2(\Delta x)}\right) + v\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}\right)$$

end

Step 4: for i=1,…..,n-1

$u(x_i, t_0) = f(x_i)$

end

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

Step 5: for .j=1,2,3,….,m

$t_{j+1} = t_j + h;$

$k_1 = h * f(t_j, u(:, t_j))$

$k_2 = h * f(t_j + \frac{1}{2}h, u(:, t_j) + \frac{1}{2}k_1)$

$k_3 = h * f(t_j + \frac{1}{2}h, u(:, t_j) + \frac{1}{2}k_2)$

$k_4 = h * f(t_j + h, u(:, t_j) + k_3)$

$$u(:, t_{j+1}) = u(:, t_j) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

end

function dudt=f(t,u)

dudt=[System of differential equations form

Step 3];

Step 6: output $u_1, u_2, \ldots . u_{n-1}$

Step 7: the solution $u = [u_0, u_1, \ldots . u_{n-1}, u_n]^T$

Step 8: Stop (the producer is complete)

Example 2. Method of lines of Fisher's equation

$$\frac{\partial u}{\partial t} = v\frac{\partial^2 u}{\partial x^2} + ku\left(1 - \frac{u}{k}\right) \quad , \quad k > 0 \ \& \ v > 0 \quad (10)$$

BCs: $u(0,t) = \alpha(t) \ \& \ u(L,t) = \beta(t) \, , t > 0$

IC: $u(x,0) = f(x) \quad , \quad 0 < x < L$

Applied method of lines we obtain

$$\frac{du_i}{dt} = v\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}\right) + ku_i\left(1 - \frac{u_i}{k}\right) \quad (11)$$

for i=1,….,n-1 we obtain nonlinear systems of ordinary differential equations

Algorithm of Fisher's Equation

To obtain the numerical solution of Fisher's equation.

Input: endpoint L; maximum time T; constant of viscosity v; constant k; integer n & m

Output: approximations $u(x_i, t_j)$

Step 1: Δx =L/n

h=Tmax/m

Step 2: when i=0

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

$u(x_0, t) = \alpha(t)$

when i=n

$u(x_n, t) = \beta(t)$

Step 3: for i=1,…..,n-1

$$\frac{du_i}{dt} = v\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}\right) + ku_i\left(1 - \frac{u_i}{k}\right)$$

end

Step 4: for i=1,…..,n-1

$u(x_i, t_0) = f(x_i)$

end

Step 5: for .j=1,2,3,….,m

$t_{j+1} = t_j + h;$

$k_1 = h * f(t_j, u(:, t_j))$

$k_2 = h * f(t_j + \frac{1}{2}h, u(:, t_j) + \frac{1}{2}k_1)$

$k_3 = h * f(t_j + \frac{1}{2}h, u(:, t_j) + \frac{1}{2}k_2)$

$k_4 = h * f(t_j + h, u(:, t_j) + k_3)$

$$u(:, t_{j+1}) = u(:, t_j) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

end

function dudt=f(t,u)

dudt=[System of differential equations form

Step 3];

Step 6: output $u_1, u_2, \ldots\ldots u_{n-1}$

Step 7: the solution $u = [u_0, u_1, \ldots\ldots u_{n-1}, u_n]^T$

Step 8: Stop (the producer is complete)

Example 3. Method of lines of Burgers-Fisher equation

$$\frac{\partial u}{\partial t} = v\frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x} + ku\left(1 - \frac{u}{k}\right), \qquad k > 0 \ \& \ v > 0 \qquad (12)$$

BCs: $u(0, t) = \alpha(t) \ \& \ u(L, t) = \beta(t), t > 0$

IC: $u(x, 0) = f(x) \ , \ 0 < x < L$

Applied method of lines we obtain

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

$$\frac{du_i}{dt} = v\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}\right) + u_i\left(\frac{u_{i-1} - u_{i+1}}{2(\Delta x)}\right) + ku_i\left(1 - \frac{u_i}{k}\right) \quad (13)$$

for i=1,….,n-1 we obtain nonlinear systems of ordinary differential equations.

Algorithm of Burgers-Fisher Equation

To obtain the numerical solution of Burgers-Fisher equation.

Input: endpoint L; maximum time T; constant of viscosity v; constant k; integer n & m

Output: approximations $u(x_i, t_j)$

Step 1: Δx =L/n

h=Tmax/m

Step 2: when i=0

$u(x_0, t) = \alpha(t)$

when i=n

$u(x_n, t) = \beta(t)$

Step 3: for i=1,…..,n-1

$$\frac{du_i}{dt} = v\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2}\right) + u_i\left(\frac{u_{i-1} - u_{i+1}}{2(\Delta x)}\right) + ku_i\left(1 - \frac{u_i}{k}\right)$$

end

Step 4: for i=1,…..,n-1

$u(x_i, t_0) = f(x_i)$

end

Step 5: for .j=1,2,3,….,m

$t_{j+1} = t_j + h;$

$k_1 = h * f(t_j, u(:, t_j))$

$k_2 = h * f(t_j + \frac{1}{2}h, u(:, t_j) + \frac{1}{2}k_1)$

$k_3 = h * f(t_j + \frac{1}{2}h, u(:, t_j) + \frac{1}{2}k_2)$

$k_4 = h * f(t_j + h, u(:, t_j) + k_3)$

$$u(:, t_{j+1}) = u(:, t_j) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

end

function dudt=f(t,u)

dudt=[System of differential equations form

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

Step 3];

Step 6: output $u_1, u_2, \ldots\ldots u_{n-1}$

Step 7: the solution $u = [u_0, u_1, \ldots\ldots u_{n-1}, u_n]^T$

Step 8: Stop (the producer is complete)

3. NUMERICAL RESULTS

In this section, we discuss the numerical results for example 1, example 2 and example3.

TABLE 1: NUMERICAL SOLUTION OF SYSTEMS ODEs OBTAINED FROM APPLIED METHOD OF BURGERS EQUATION. IF LENGTH OF X-AXIS EQUAL TEN & WIDTH EQUAL ONE, CONSIDER $n = 6$, $\alpha = 1$, $\beta = 2$ $v = 1$, $f(x) = x^2$.

$u_1(t)$	$u_2(t)$	$u_3(t)$	$u_4(t)$	$u_5(t)$
2.7778	11.1111	25.0000	44.4444	69.4444
2.7552	10.8472	24.1067	42.1731	72.6412
2.7335	10.5992	23.2984	39.8456	75.7809
2.7124	10.3654	22.5697	37.4921	78.8407
2.6920	10.1442	21.9151	35.1409	81.8000
2.6352	9.5516	20.3631	28.4457	89.7840
2.5830	9.0319	19.2752	22.4580	96.4402
2.5350	8.5662	18.5316	17.4162	101.6853
2.4904	8.1411	18.0464	13.3878	105.5656
2.4616	7.8673	17.8269	11.1565	107.5377
2.4341	7.6067	17.6767	9.3113	109.0270
2.4079	7.3579	17.5811	7.8025	110.1026
2.3829	7.1195	17.5271	6.5784	110.8337
2.3591	6.8905	17.5039	5.5881	111.2855
2.3362	6.6704	17.5046	4.7921	111.5067
2.3144	6.4586	17.5235	4.1558	111.5398
2.2935	6.2545	17.5549	3.6476	111.4239
2.2742	6.0646	17.5928	3.2536	111.2017
2.2556	5.8814	17.6360	2.9371	110.8913
2.2377	5.7046	17.6822	2.6837	110.5101
2.2206	5.5339	17.7295	2.4805	110.0737
2.2033	5.3617	17.7786	2.3100	109.5716
2.1866	5.1958	17.8261	2.1749	109.0326
2.1706	5.0362	17.8713	2.0681	108.4649
2.1552	4.8825	17.9132	1.9836	107.8759
2.1388	4.7189	17.9553	1.9098	107.2051
2.1230	4.5622	17.9922	1.8532	106.5204
2.1078	4.4123	18.0238	1.8099	105.8261
2.0932	4.2689	18.0498	1.7769	105.1258
2.0769	4.1103	18.0727	1.7480	104.3099
2.0612	3.9599	18.0879	1.7273	103.4926
2.0462	3.8172	18.0955	1.7128	102.6761
2.0318	3.6819	18.0957	1.7028	101.8623
2.0151	3.5281	18.0863	1.6948	100.8855
1.9992	3.3841	18.0670	1.6903	99.9158
1.9839	3.2493	18.0381	1.6885	98.9542
1.9693	3.1232	18.0004	1.6883	98.0017
1.9517	2.9769	17.9413	1.6893	96.8224

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

1.9350	2.8425	17.8703	1.6915	95.6583
1.9190	2.7191	17.7883	1.6946	94.5092
1.9037	2.6058	17.6965	1.6979	93.3756
1.8848	2.4727	17.5643	1.7021	91.9286
1.8668	2.3536	17.4190	1.7062	90.5067
1.8496	2.2471	17.2626	1.7104	89.1091
1.8332	2.1518	17.0966	1.7142	87.7356
1.8132	2.0449	16.8737	1.7183	86.0205
1.7943	1.9519	16.6404	1.7220	84.3422
1.7761	1.8711	16.3989	1.7253	82.6991
1.7588	1.8008	16.1513	1.7281	81.0906
1.7421	1.7395	15.8991	1.7303	79.5157
1.7260	1.6859	15.6440	1.7321	77.9732
1.7105	1.6391	15.3870	1.7336	76.4622
1.6955	1.5982	15.1293	1.7348	74.9819
1.6810	1.5622	14.8716	1.7357	73.5316
1.6669	1.5306	14.6148	1.7363	72.1103
1.6533	1.5027	14.3594	1.7368	70.7174
1.6400	1.4781	14.1059	1.7370	69.3523
1.6271	1.4564	13.8548	1.7371	68.0142
1.6145	1.4371	13.6064	1.7369	66.7026
1.6022	1.4200	13.3610	1.7367	65.4168
1.5903	1.4048	13.1188	1.7363	64.1563
1.5786	1.3912	12.8801	1.7359	62.9205
1.5672	1.3791	12.6449	1.7353	61.7089
1.5561	1.3682	12.4134	1.7347	60.5210
1.5452	1.3584	12.1856	1.7339	59.3563
1.5393	1.3534	12.0606	1.7335	58.7183
1.5334	1.3486	11.9367	1.7330	58.0871
1.5276	1.3442	11.8141	1.7326	57.4629
1.5219	1.3399	11.6927	1.7321	56.8454

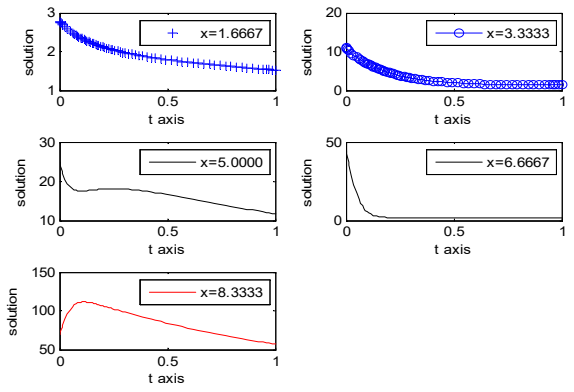

Fig. 1: Graphical Representation of Nonlinear Systems of ODEs Obtain Form Burgers Equation.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

TABLE 2: NUMERICAL SOLUTION OF BURGERS EQUATION. IF LENGTH OF X-AXIS EQUAL TEN & WIDTH EQUAL ONE, CONSIDER $n = 6,\ \alpha = 1\ , \beta = 2\ \ v = 1,\ f(x) = x^2.$

Approximation Solution				
t=0.925	t=0.950	t=0.975	t=1.00	Error
1.0000	1.0000	1.0000	1.0000	0.0000
1.9517	1.9350	1.9190	1.9037	0.0153
2.9769	2.8425	2.7191	2.6058	0.1133
17.9413	17.8703	17.7883	17.6965	0.0918
1.6893	1.6915	1.6946	1.6979	0.0033
96.8224	95.6583	94.5092	93.3756	1.1336
2.0000	2.0000	2.0000	2.0000	0.0000

Error=$\left|\underline{u}^{j+1} - \underline{u}^{j}\right|$

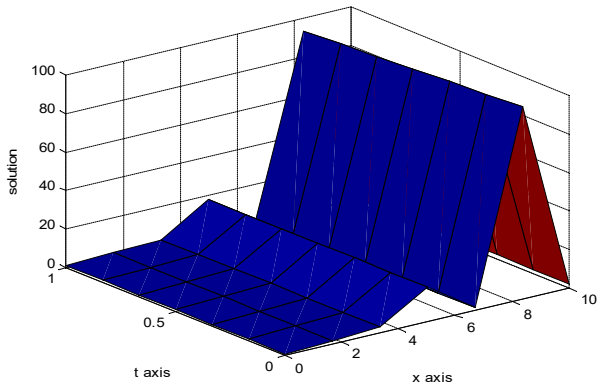

Fig. 2: Graphical Representation of Burgers Equation.

Table 1 explain the solution of Burger's differential equation at difference time and table 2 explain the convergence solution of Burger's differential equation and estimation error.

TABLE 3: NUMERICAL SOLUTION OF SYSTEMS ODEs OBTAINED FROM APPLIEDMETHOD OF FISHER'S EQUATION. IF LENGTH OF X-AXIS EQUAL TEN & WIDTH EQUAL ONE, CONSIDER $n = 6,\ \alpha = 1\ , \beta = 2\ \ v = 1,$ $k = 2,\ f(x) = x.$

$u_1(t)$	$u_2(t)$	$u_3(t)$	$u_4(t)$	$u_5(t)$
1.6667	3.3333	5.0000	6.6667	8.3333
1.6735	3.3004	4.8904	6.4421	7.9373
1.6801	3.2686	4.7870	6.2352	7.5819
1.6865	3.2377	4.6892	6.0437	7.2614
1.6927	3.2077	4.5966	5.8662	6.9707
1.7070	3.1389	4.3918	5.4855	6.3671
1.7204	3.0750	4.2106	5.1624	5.8768
1.7329	3.0155	4.0493	4.8850	5.4722
1.7445	2.9601	3.9049	4.6440	5.1308
1.7583	2.8942	3.7402	4.3770	4.7614
1.7708	2.8338	3.5953	4.1493	4.4557
1.7822	2.7783	3.4671	3.9531	4.1992
1.7926	2.7272	3.3530	3.7823	3.9805

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

1.8027	2.6770	3.2442	3.6228	3.7798
1.8117	2.6309	3.1471	3.4831	3.6072
1.8199	2.5884	3.0600	3.3598	3.4574
1.8272	2.5493	2.9815	3.2503	3.3263
1.8338	2.5131	2.9104	3.1525	3.2107
1.8397	2.4796	2.8459	3.0648	3.1083
1.8450	2.4486	2.7871	2.9858	3.0172
1.8497	2.4199	2.7335	2.9143	2.9356
1.8539	2.3932	2.6843	2.8494	2.8622
1.8576	2.3683	2.6392	2.7903	2.7961
1.8609	2.3452	2.5977	2.7363	2.7362
1.8638	2.3236	2.5594	2.6869	2.6819
1.8663	2.3035	2.5241	2.6415	2.6324
1.8685	2.2847	2.4913	2.5998	2.5873
1.8705	2.2672	2.4610	2.5613	2.5460
1.8721	2.2508	2.4329	2.5257	2.5081
1.8735	2.2354	2.4067	2.4928	2.4733
1.8748	2.2210	2.3823	2.4623	2.4413
1.8758	2.2075	2.3596	2.4340	2.4118
1.8766	2.1948	2.3385	2.4077	2.3845
1.8773	2.1830	2.3187	2.3832	2.3594
1.8779	2.1718	2.3002	2.3604	2.3361
1.8783	2.1613	2.2829	2.3391	2.3145
1.8786	2.1514	2.2667	2.3192	2.2944
1.8788	2.1422	2.2515	2.3007	2.2758
1.8789	2.1334	2.2373	2.2833	2.2585
1.8789	2.1252	2.2239	2.2670	2.2424
1.8789	2.1175	2.2114	2.2518	2.2273
1.8788	2.1099	2.1992	2.2370	2.2129
1.8786	2.1028	2.1877	2.2232	2.1994
1.8784	2.0961	2.1770	2.2102	2.1869
1.8781	2.0898	2.1669	2.1981	2.1752

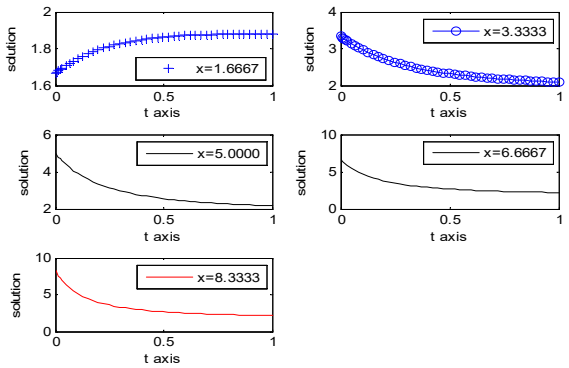

Fig. 3: Graphical Representation of Nonlinear Systems of ODEs Obtain Form Fisher's Equation.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

TABLE 4: NUMERICAL SOLUTION OF FISHER'S EQUATION. IF LENGTH OF X-AXIS EQUAL TEN & WIDTH EQUAL ONE, CONSIDER $n = 6,\ \alpha = 1\ , \beta = 2\ \ v = 1,\ k = 2,\ f(x) = x.$

Approximation Solution				
t=0.925	t=0.95	t=0.975	t=1.00	Error
1.0000	1.0000	1.0000	1.0000	0.0000
1.8788	1.8789	1.8789	1.8789	0.0000
2.1422	2.1334	2.1252	2.1175	0.0077
2.2515	2.2373	2.2239	2.2114	0.0126
2.3007	2.2833	2.2670	2.2518	0.0152
2.2758	2.2585	2.2424	2.2273	0.0150
2.0000	2.0000	2.0000	2.0000	0.0000

Error=$\left|\underline{u}^{j+1} - \underline{u}^{j}\right|$

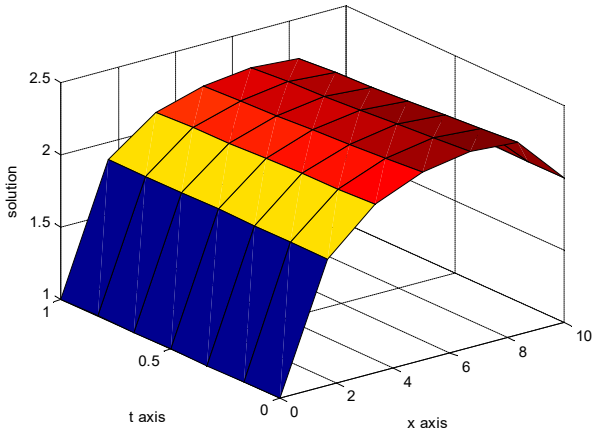

Fig. 4: Graphical Representation of Fisher's Equation.

Table 3 explain the solution of Fisher's differential equation at difference time and table 4 explain the convergence solution of Fisher's differential equation and estimation error.

TABLE 5: NUMERICAL SOLUTION OF SYSTEMS ODEs OBTAINED FROM APPLIEDMETHOD OF BURGERS-FISHER EQUATION. IF LENGTH OF X-AXIS EQUAL TEN & WIDTH EQUAL ONE, CONSIDER $n = 6,\ \alpha = 1\ , \beta = 2\ \ v = 1,\ k = 2,\ f(x) = 200$

$u_1(t)$	$u_2(t)$	$u_3(t)$	$u_4(t)$	$u_5(t)$
200.0000	200.0000	200.0000	200.0000	200.0000
194.7253	192.5577	192.5698	192.5575	190.4226
189.7003	185.6304	185.6747	185.6302	181.7040
184.9093	179.1678	179.2589	179.1679	173.7352
180.3377	173.1256	173.2737	173.1265	166.4235
168.4251	157.9124	158.2497	157.9192	148.5568
157.9118	145.1057	145.6508	145.1219	134.1500
148.5769	134.1913	134.9437	134.2190	122.3348
140.2404	124.7674	125.7063	124.8083	112.4017
130.8910	114.5308	115.6668	114.5890	101.8341

122.6760	105.8290	107.1304	105.9030	93.0844
115.4065	98.3479	99.7868	98.4358	85.7337
108.9300	91.8409	93.3906	91.9405	79.4502
100.8306	83.8981	85.5662	84.0102	71.9014
93.8367	77.2187	78.9706	77.3397	65.6733
87.7409	71.5300	73.3395	71.6569	60.4573
82.3787	66.6179	68.4642	66.7481	56.0080
75.8914	60.7806	62.6518	60.9118	50.7763
70.3490	55.8924	57.7671	56.0217	46.4522
65.5625	51.7441	53.6077	51.8697	42.8249
61.3832	48.1714	50.0139	48.2919	39.7266
56.3369	43.9131	45.7154	44.0251	36.0601
52.0629	40.3599	42.1144	40.4626	33.0281
48.3996	37.3539	39.0568	37.4470	30.4834
45.2197	34.7705	36.4205	34.8541	28.3089
41.3996	31.6955	33.2719	31.7661	25.7328
38.1846	29.1357	30.6406	29.1940	23.6016
35.4441	26.9745	28.4113	27.0214	21.8121
33.0753	25.1198	26.4924	25.1562	20.2823
30.2365	22.9115	24.2008	22.9346	18.4670
27.8588	21.0764	22.2898	21.0877	16.9648
25.8404	19.5293	20.6738	19.5302	15.7034
24.1011	18.2030	19.2847	18.1946	14.6249
22.0202	16.6234	17.6262	16.6037	13.3435
20.2834	15.3124	16.2456	15.2830	12.2833
18.8138	14.2085	15.0801	14.1708	11.3932
17.5503	13.2628	14.0795	13.2180	10.6323
16.0401	12.1362	12.8849	12.0828	9.7274
14.7832	11.2022	11.8923	11.1418	8.9792
13.7222	10.4166	11.0555	10.3502	8.3514
12.8117	9.7441	10.3380	9.6728	7.8150
11.7246	8.9429	9.4819	8.8658	7.1772
10.8218	8.2795	8.7716	8.1978	6.6504
10.0613	7.7221	8.1737	7.6366	6.2089
9.4097	7.2454	7.6618	7.1568	5.8321
8.6331	6.6781	7.0519	6.5861	5.3848
7.9894	6.2089	6.5467	6.1143	5.0161
7.4483	5.8152	6.1222	5.7187	4.7078
6.9854	5.4789	5.7593	5.3809	4.4454
6.4358	5.0801	5.3286	4.9807	4.1353
5.9812	4.7508	4.9725	4.6506	3.8806
5.5998	4.4749	4.6740	4.3743	3.6685
5.2742	4.2397	4.4193	4.1391	3.4888
4.9506	4.0063	4.1663	3.9059	3.3114
4.6750	3.8077	3.9511	3.7079	3.1618
4.4380	3.6372	3.7661	3.5381	3.0343
4.2317	3.4890	3.6053	3.3908	2.9244
4.0508	3.3590	3.4643	3.2621	2.8290
3.8911	3.2445	3.3400	3.1488	2.7457
3.7492	3.1430	3.2298	3.0486	2.6726
3.6225	3.0524	3.1315	2.9594	2.6081
3.5088	2.9711	3.0433	2.8797	2.5508
3.4062	2.8979	2.9639	2.8081	2.4998

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

3.3133	2.8318	2.8921	2.7436	2.4543
3.2290	2.7718	2.8270	2.6853	2.4135
3.1521	2.7171	2.7677	2.6324	2.3769
3.0818	2.6672	2.7137	2.5843	2.3439
3.0173	2.6215	2.6642	2.5404	2.3140
2.9581	2.5796	2.6188	2.5003	2.2871
2.9035	2.5411	2.5771	2.4636	2.2626
2.8531	2.5055	2.5386	2.4299	2.2404
2.8065	2.4727	2.5032	2.3989	2.2203
2.7633	2.4423	2.4704	2.3704	2.2019
2.7232	2.4141	2.4400	2.3442	2.1851
2.6859	2.3880	2.4119	2.3199	2.1699
2.6512	2.3637	2.3857	2.2975	2.1559
2.6188	2.3411	2.3615	2.2768	2.1431
2.5866	2.3186	2.3373	2.2563	2.1307
2.5566	2.2978	2.3149	2.2374	2.1193
2.5286	2.2784	2.2942	2.2200	2.1090
2.5026	2.2603	2.2749	2.2039	2.0996

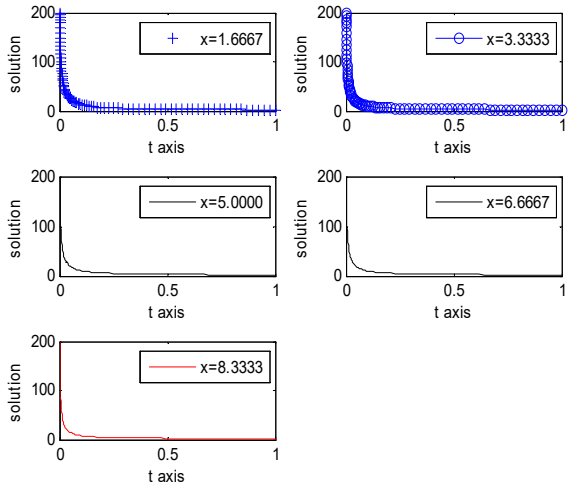

Fig. 5: Graphical Representation of Nonlinear Systems of ODEs Obtain Form Burgers-Fisher Equation.

TABLE 6: NUMERICAL SOLUTION OF BURGERS-FISHER EQUATION. IF LENGTH OF X-AXIS EQUAL TEN & WIDTH EQUAL ONE, CONSIDER $n = 6,\ \alpha = 1\ , \beta = 2\ \ v = 1,\ k = 2,\ f(x) = 200.$

Approximation Solution				
t=0.925	t=0.950	t=0.975	t=1.00	Error
1.0000	1.0000	1.0000	1.0000	0.0000
16.0401	14.7832	13.7222	12.8117	0.9105
12.1362	11.2022	10.4166	9.7441	0.6725
12.8849	11.8923	11.0555	10.3380	0.7175
12.0828	11.1418	10.3502	9.6728	0.6774
9.7274	8.9792	8.3514	7.8150	0.5364
2.0000	2.0000	2.0000	2.0000	0.0000

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

$$\text{Error}=\left|\underline{u}^{j+1}-\underline{u}^{j}\right|$$

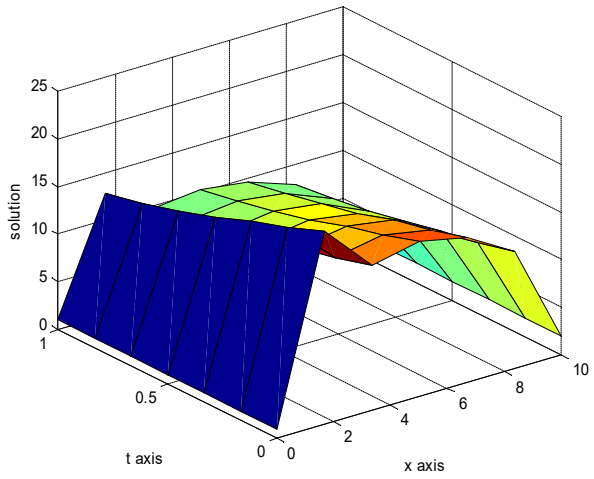

Fig. 6: Graphical Representation of Burgers-Fisher Equation.

Table 5 explain the solution of Burger- Fisher differential equation at difference time and table 6 explain the convergence solution of Burger- Fisher differential equation and estimation error.

The previous methods are study the Burger-Fisher equation, in the following form [1, 6]

$$\frac{\partial u}{\partial t}=\mu\frac{\partial^2 u}{\partial x^2}+u^{\delta}\frac{\partial u}{\partial x}+u\left(1-u^{\delta}\right)\ ,\quad \mu>0$$

with boundary conditions:

$$u(0,t)=\left(\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{\alpha\delta}{2(\delta+1)}\left(\frac{\alpha}{(\delta+1)}+\frac{\beta(\delta+1)}{\alpha}\right)\right)t\right)^{\frac{1}{\delta}}$$

$$u(L,t)=\left(\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}\left(1-\left(\frac{\alpha}{(\delta+1)}+\frac{\beta(\delta+1)}{\alpha}\right)t\right)\right)\right)^{\frac{1}{\delta}}$$

and initial condition:

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

The exact solution is

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

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

TABLE 7: NUMERICAL SOLUTION OF BURGERS-FISHER EQUATION. IF LENGTH OF X-AXIS EQUAL ONE, $\mu = 1$, $\beta = 0.01$, $\alpha = 1$, $\delta = 2$, $t = 0.0006$ & $n = 10$

Comparison between numerical methods								
Exact	EM	EXPM	DFM	MOL	Error EM	Error EXPM	Error DFM	Error MOL
0.7071	0.7071	0.7071	0.7071	0.7071	0.0000	0.0000	0.0000	0.0000
0.6953	0.6953	0.6953	0.6952	0.6952	0.0000	0.0000	0.0000	0.0000
0.6832	0.6832	0.6832	0.6832	0.6832	0.0000	0.0000	0.0000	0.0000
0.6710	0.6710	0.6710	0.6709	0.6709	0.0000	0.0000	0.0000	0.0000
0.6586	0.6586	0.6586	0.6586	0.6586	0.0000	0.0000	0.0000	0.0000
0.6461	0.6461	0.6461	0.6461	0.6461	0.0000	0.0000	0.0000	0.0000
0.6335	0.6335	0.6335	0.6335	0.6335	0.0000	0.0000	0.0000	0.0000
0.6208	0.6208	0.6208	0.6208	0.6208	0.0000	0.0000	0.0000	0.0000
0.6081	0.6081	0.6081	0.6081	0.6081	0.0000	0.0000	0.0000	0.0000
0.5953	0.5953	0.5953	0.5953	0.5953	0.0000	0.0000	0.0000	0.0000
0.5825	0.5825	0.5825	0.5825	0.5825	0.0000	0.0000	0.0000	0.0000

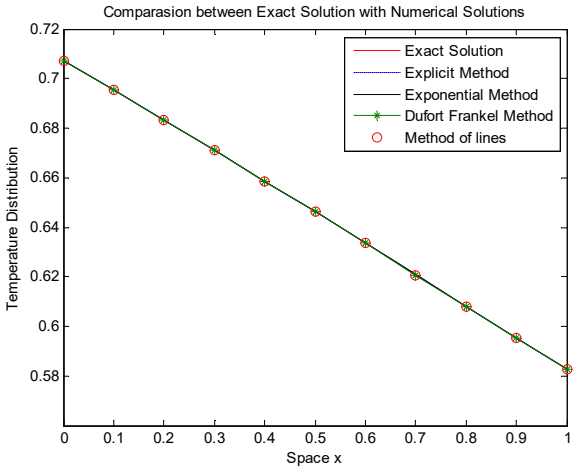

Fig. 7: Comparison between numerical methods of Burger-Fisher Equation when the interval [0, 1].

Table 7 and Figure 7, explain the convergence solution of Burger- Fisher differential equation and estimation error when the interval is [0, 1].

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

TABLE 8: NUMERICAL SOLUTION OF BURGERS-FISHER EQUATION. IF LENGTH OF X-AXIS EQUAL TEN, $\mu = 1$, $\beta = 0.001$, $\alpha = 5$, $\delta = 1$, $t = 1$ & $n = 10$

Comparison between numerical methods								
Exact	EM	EXPM	DFM	MOL	Error EM	Error EXPM	Error DFM	Error MOL
0.9981	0.9981	0.9981	0.9981	0.9981	0.0000	0.0000	0.0000	0.0000
0.9770	1.0212	1.0913	1.0108	1.1044	0.0441	0.1143	0.0338	0.1273
0.7775	0.4288	0.6101	0.4043	0.6988	0.3487	0.1674	0.3732	0.0787
0.2229	0.0874	0.1557	0.0793	0.1916	0.1355	0.0672	0.1436	0.0313
0.0230	0.0136	0.0290	0.0119	0.0353	0.0094	0.0060	0.0111	0.0123
0.0019	0.0018	0.0049	0.0016	0.0056	0.0001	0.0030	0.0004	0.0037
0.0002	0.0002	0.0008	0.0002	0.0008	0.0001	0.0006	0.0000	0.0007
0.0000	0.0000	0.0001	0.0000	0.0001	0.0000	0.0001	0.0000	0.0001
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

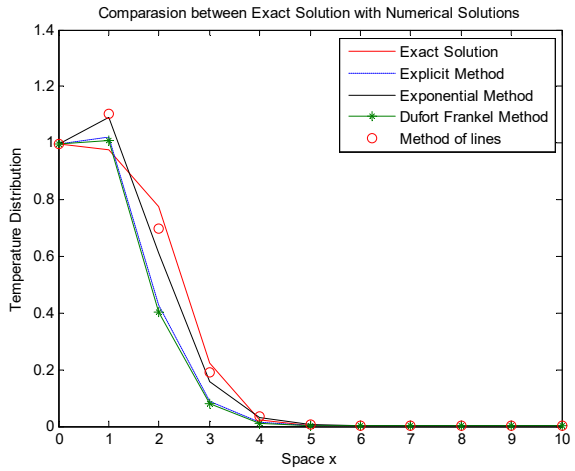

Fig. 8: Comparison between numerical methods of Burger-Fisher Equation when the interval [0, 10].

Table 8 and Figure 8, explain the convergence solution of Burger- Fisher differential equation and estimation error when the interval is [0, 10].

TABLE 9: NUMERICAL SOLUTION OF BURGERS-FISHER EQUATION. IF LENGTH OF X-AXIS EQUAL TEENTEY, $\mu = 1$, $\beta = 0.001$, $\alpha = 5$, $\delta = 1$, $t = 2$ & $n = 10$

Comparison between numerical methods								
Exact	EM	EXPM	DFM	MOL	Error EM	Error EXPM	Error DFM	Error MOL
1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.9994	0.8623	NaN	0.8458	1.0376	0.1371	NaN	0.1537	0.0381
0.9243	0.1550	NaN	0.1479	0.2673	0.7693	NaN	0.7763	0.6570
0.0760	0.0140	NaN	0.0131	0.0299	0.0620	NaN	0.0629	0.0461
0.0006	0.0010	NaN	0.0009	0.0027	0.0004	NaN	0.0003	0.0022
0.0000	0.0001	NaN	0.0000	0.0002	0.0000	NaN	0.0000	0.0002
0.0000	0.0000	NaN	0.0000	0.0000	0.0000	NaN	0.0000	0.0000

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 51 No. 2 July 2025, pp. 183-200

0.0000	0.0000	NaN	0.0000	0.0000	0.0000	NaN	0.0000	0.0000
0.0000	0.0000	NaN	0.0000	0.0000	0.0000	NaN	0.0000	0.0000
0.0000	0.0000	NaN	0.0000	0.0000	0.0000	NaN	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

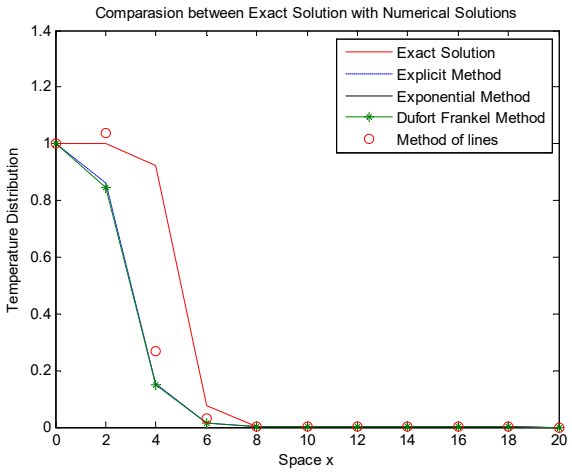

Fig. 9: Comparison between numerical methods of Burger-Fisher Equation when the interval [0, 20].

Table 9 and Figure 9, explain the convergence solution of Burger- Fisher differential equation and estimation error when the interval is [0, 20].

4. CONCLUSION

In this paper, we discussed the numerical solution of nonlinear redaction-diffusion equations. We get the method of lines (MOL) is the good convergence comparison with previous methods explicit method (EM), exponential method (EXPM) and Dufort -Frankel method (DFM). In figure 7, all these methods are given a good approximation and higher accuracy, but when we change the interval in figure [8] and figure [9] we get the method of lines is best and give best accuracy from others methods. I hop in further the researchers use the method of line with finite difference or with spectral method.

REFERENCES

[1] Abdulghafor M. Al-Rozbayani, Karam A. Al-Hayalie. Numerical Solution of Burger's_Fisher Equation in One - Dimensional Using Finite Differences Methods. Fluid Mechanics. Vol. 4, No. 1, pp. 20-26, 2018.

[2] Lokenath Debnath, Nonlinear Partial Differential Equations for Scientists and Engineers, fourth edition, 2007.

[3] Schiesser, W. E. The Numerical Method of Lines: Integration of Partial Differential Equations. 2nd Edition, Clarendon Presses, Oxford, 1978.

[4] Carver, M.B. and Hinds, H.W. The Method of lines and the Advective Equation. Simulation, 1978, **31**, 59-69.

[5] Randall J. Leveque. Finite Difference Methods for Ordinary and Partial Differential Equations, 2007.

[6] Viney Chandraker Ashish Awasthis Simon Jayarj, Numerical Tratment of Burger-Fisher Equation. Vol. 25, pp. 1217 – 1225, 2016.

[7] Elbarjo, R. M., Abuzreda, A., AL-Sammarraie, O. A., Alhrir, E. H., & khaleelalfarrah, H. (2025). The role of linear algebra in artificial intelligence: Foundations, applications, and innovations. International Journal of Engineering Inventions, 14(3), 21-23. https://www.ijeijournal.com

[8] Elbarjo, R. M., Al-Zahawi, A. A., Saeid, H. S., Alzayani, A., & Abuzreda, A. (2025). Alpha Tensor: A transformer-based AI system for optimizing matrix multiplication across modern computing architectures with implications for safety, environment, and civil engineering applications. International Journal of Engineering Inventions, 14(5), 137-140. https://www.ijeijournal.com