American International Journal of Sciences and Engineering Research
Vol. 4, No. 1; 2021
ISSN 2641-0303 E- ISSN 2641-0311
Published by American Center of Science and Education, USA

NUMERICAL SOLUTION FOR NONLINEAR BURGERS' EQUATION WITH SOURCE TERM

Falade Kazeem Iyanda
Department of Mathematics
Faculty of Computing and Mathematical Sciences
Kano University of Science and Technology, Wudil, P.M.B 3244, Kano State, Nigeria
E-mail: faladekazeem2016@kustwudil.edu.ng

Bello Kareem Akanbi
Department of Mathematics
Faculty of Physical Sciences, University of Ilorin
P.M.B.1515, Ilorin, Kwara State, Nigeria
E-mail: bello.ak@unilorin.edu.ng

Nuru Muazu
Department of Mathematics
Faculty of Computing and Mathematical Sciences
Kano University of Science and Technology, Wudil, P.M.B 3244, Kano State, Nigeria
E-mail: nurumuazu@kustwudil.edu.ng

ABSTRACT

In this paper, we formulate a four step computational algorithm to solve nonlinear Burger's equation with source terms whose occur in aerodynamics engineering which play a major roles in convection and diffusion whose present in viscous fluid flow engineering problems. Numerical assessment was carried out to study effect of source term $\lambda\varphi$ which represents the heat released in the boundary layer. Increase the source λ term and decrease in v kinematic viscosity which play a major roles in obtaing velocity $\varphi(x,t)$. Eventually, we subject the nonlinear Burger's equation with source terms to initial and boundary conditions available in the literature. The results revealed that the new algorithm is capable and realiable to solve similar nonlinear partial differential equations occur in applied physics and engineering.

Keywords: A four step algorithm, Nonlinear Burger's equation, Source terms, Aerodynamics theory, Viscous fluid flow.

INTRODUCTION

In this paper, we consider nonlinear Burgers' equation with source terms in aerodynamics engineering of the form (Mayur et al., 2018):

$$\frac{\partial \varphi}{\partial t} + \varphi \frac{\partial \varphi}{\partial x} = v \frac{\partial \varphi}{\partial x} x - \lambda \varphi \qquad x, v, \lambda > 0 \tag{1}$$

with initial and boundary conditions given (Chandrasekharan Nair, & Awasthi, 2019).

$$\begin{cases} \varphi(x,t_0) = \dfrac{x}{1+\dfrac{1}{c_0}e^{\frac{x^2}{4v}}} & 0 < x < 1 \\ \varphi(x_0,t) = t_0 & 1 \le t \le T \\ \varphi(1,t) = t_0 & 1 \le t \le T \end{cases} \tag{2}$$

Where $\varphi(x,t)$ is the dependent variable which x is space coordinate and t is time, $\lambda\varphi$ represents source terms which plays a role of releases heat in the boundary layer, v is kinematic viscosity and c_0 is a constant.

Burgers' equation was proposed by Johannes Martinus Burgers (1895–1981), a Dutch physicist, and become one of the leading equations in the field of fluid mechanics which explained details in theory of fluid flow dynamics of traffic and turbulence shock waves. Its contributed a great significance in the diversified physical problems related to engineering and applied sciences. Burgers' equation is one of the famous non-linear partial differential equations which is suitable for the analysis of various important areas in applied mathematics and engineering physics. (Burgers, 1939, Nieuwstadt & Steketee, 2012, Abdou & Soliman, 2005, Burgers, 1948, Panayotounakos & Drikakis, 1995).

In recent years, several reseaachers have developed a significant computational techniques for solving linear and non-linear PDEs found in many physical phenomena in applied mathematics. Burgers' equation is one of the most partial equations describes non-linear propagation and diffusive effects which occurs and represents various physical problems arising in applied sciences which is difficult to solve analytically. The presence of non-linear convective term $\varphi\frac{\partial\varphi}{\partial x}$ and $v\frac{\partial\varphi}{\partial x}x$ feature in Burgers' equation (1) describe a situation when v approaches zero, it become inviscid Burgers' equation of a model of wave propagation. Moreover, as φ tends to zero, equation (1) becomes the heat equation which describe unsteady heat equation without internal heat generation arises in the mathematical modelling of many physical phenomena occurring in nature. Ismail et al. (2004) applied adomian decomposition method to obtain numerical solutions of Burger's equation, homotopy perturbation method was employed to solve coupled Burger's equations by (Desai & Pradhan, 2012), authors (Abdou & Soliman, 2005) presented and applied variational iteration method for solving Burger's and coupled Burger's equations and (Mittal & Jain, 2021) proposed modified cubic B-splines over finite elements as a method for solving Burgers' equation and (Chandrasekharan & Awasthi, 2019) presented and applied quintic trigonometric spline based numerical scheme for nonlinear modified Burgers' equation and just to mention a few.

The objective of this paper is to modified the algorithm presented by (Falade & Tiamiyu, 2020) and to investigate behavior of parameters ($\lambda\varphi$ represents source terms and v kinematic viscosity) on Burgers' equation (1) which explains the role of diffusion and convection when a non-autonomous reaction term produces heat constantly in a viscous flow (Mohamed, 2019) , furthermore, we present 3D plots, 2D plots and densityplots for the heat distrubution profiles of Burgers' equation.

NEW FOUR STEPS ALGORITHM (NFSA)

In this section, we present a four steps algorithm using MAPLE18 codes software package to solve equation (1) with initial conditions (2) as follows:

restart:

Step 1:

$Digits := 23;$

$N := 4;$

$\mathrm{C}[0] := \frac{1}{2};$

$v := \begin{cases} [0.01, 0.001, 0.0001, 0.00001] \\ [0.10, 0.20, 0.30, 0.40\ ] \end{cases};$

$\lambda := \begin{cases} [0.10, 0.20, 0.30, 0.40\ ] \\ [0.01, 0.001, 0.0001, 0.00001] \end{cases};$

$$IBC := \begin{cases} \varphi(x,0) = \dfrac{x}{1 + \dfrac{1}{c_0} e^{\frac{x^2}{4v}}} \\ \varphi(x_0, t) = 0 \\ \varphi(1,t) = 0 \end{cases};$$

$\varphi[0] := IBC;$

Step 2:

$BPDE := value(-\varphi[0] * (diff(\varphi[0], x) + v * (diff(\varphi[0], x) * x - \lambda * \varphi[0]);$

$\varphi[1] := value\big(int(BPDE, t = 0 \ldots t)\big);$

$for\ m\ from\ 1\ to\ N\ do$

$\varphi[m+1] := value\Big(\Big(int\big((-(sum(\varphi[n], n = 0 \ldots m, x) * Diff(sum(\varphi[n], n = 0 \ldots m, x) + v * Diff\big(sum(\varphi[n], n = 0 \ldots m, x) * x - \lambda * (sum(\varphi[n], n = 0 \ldots m, x)\big), t = 0 \ldots t)\Big) - (int(-(sum(\varphi[n], n = 0 \ldots m-1, x) * Diff(sum(\varphi[n], n = 0 \ldots m-1, x) + v * Diff(sum(\varphi[n], n = 0 \ldots m-1, x) * x - \lambda * (sum(\varphi[n], n = 0 \ldots m-1, x), t = 0 \ldots t))\Big):$

(3)

$end\ do$

$BPDE^* := sum(\varphi[k], k = 0 \ldots . N+1);$

$\varphi[sol] := evalf(BPDE^*);$

$for\ i\ from\ 0\ by\ 0.1\ to\ 1\ do$

$\varphi[i] := evalf(eval(\varphi[sol, [x = i, t = i]]))$

$end\ do$

Step 3:

$sol1 := BPDE^*;$

$M[0] := eval(sol1, [t = 0]);$

$M[2] := eval(sol1, [t = 2]);$

$M[4] := eval(sol1, [t = 4]);$

$M[6] := eval(sol1, [t = 6]);$

$M[8] := eval(sol1, [t = 8]);$

$M[10] \coloneqq eval(sol1, [t = 10]);$
$end\ do$
Step 4:
withplot:
$\varphi[3Dplot] \coloneqq plot3d([sol1], x = 0 \ldots 1, t = 0 \ldots 1, grid = [100,100], red, blue, green, orange);$
$\varphi[2Dplot] \coloneqq plot2d([M[0], M[2], M[4], M[6], M[8], M[10],], t = 0 \ldots 3, color: red, blue, yellow, purple, black, green);$
$\varphi[densityplot] \coloneqq densityplot([sol1], x = -\pi \ldots \pi, t = -\pi \ldots \pi, axes = boxed, colourstye = HUE);$
Output: See Table 1, Table 2 and Figures 1,2,3,..6
where N is the computational length and positive integer.

COMPUTATIONAL EXPERIMENT AND NUMERICAL SOLUTIONS

In this section, we performed numerical experiment on equation (1) using formulated algorithm when source term λ is greater than kinematic viscosity v and vice versa.

Table 1. Numerical solutions $\varphi(x,t)$ when source term λ greater than kinematic viscosity v

(x,t)	v kinematic viscosity	λ source term	$\varphi(x,t)$ Analytical solutions $\lambda > v$	$\varphi(x,t)$ Algorithm solutions $\lambda > v$
(0,0)	0.0100000	0.1000000	0.000000000000000000000000	0.000000000000000000000000
	0.0010000	0.2000000	0.000000000000000000000000	0.000000000000000000000000
	0.0001000	0.3000000	0.000000000000000000000000	0.000000000000000000000000
	0.0000100	0.4000000	0.000000000000000000000000	0.000000000000000000000000
(0.1, 0.1)	0.0100000	0.1000000	0.001920579197785102005257 2	0.001920579197785102005257 7
	0.0010000	0.2000000	0.000196614320676673391590 0	0.000196614320676673391590 5
	0.0001000	0.3000000	0.000019731007459946011866 3	0.000019731007459946011866 8
	0.0000100	0.4000000	0.000001975114599482473748 9	0.000001975114599482473749 4
(0.2, 0.2)	0.0100000	0.1000000	0.003690145737918452088006 4	0.003690145737918452088007 2
	0.0010000	0.2000000	0.000379698559955874827468 3	0.000379698559955874827468 7
	0.0001000	0.3000000	0.000038168242256233003902 9	0.000038168242256233003903 3
	0.0000100	0.4000000	0.000003823909997534372847 2	0.000003823909997534372847 6
(0.3, 0.3)	0.0100000	0.1000000	0.005216102880027940112387 3	0.005216102880027940112387 7
	0.0010000	0.2000000	0.000539101530516035702614 5	0.000539101530516035702614 8
	0.0001000	0.3000000	0.000054279230532175481090 1	0.000054279230532175481090 5
	0.0000100	0.4000000	0.000005442503575721410432 9	0.000005442503575721410433 3
(0.4, 0.4)	0.0100000	0.1000000	0.006428753085705186966545 8	0.006428753085705186966546 1
	0.0010000	0.2000000	0.000666954988598454322373 9	0.000666954988598454322374 1
	0.0001000	0.3000000	0.000067255852752981545597 1	0.000067255852752981545597 4
	0.0000100	0.4000000	0.000006749190734094126076 8	0.000006749190734094126077 1
(0.5, 0.5)	0.0100000	0.1000000	0.007285362894096138966283 8	0.007285362894096138966284 1
	0.0010000	0.2000000	0.000758289068949001223368 0	0.000758289068949001223368 3
	0.0001000	0.3000000	0.000076579863032005960217 6	0.000076579863032005960217 9
	0.0000100	0.4000000	0.000007691133311507934426 8	0.000007691133311507934428 1
	0.0100000	0.1000000	0.007771489730263951930615 7	0.007771489730263951930615 0

(0.6, 0.6)	0.0010000	0.2000000	0.00081128512814558134558 54	0.00081128512814558134558 57
	0.0001000	0.3000000	0.00008205139413185243927 71	0.00008205139413185243927 71
	0.0000100	0.4000000	0.00000824735093227031320 49	0.00000824735093227031320 51
(0.7, 0.7)	0.0100000	0.1000000	0.00789982060625534447834 36	0.00789982060625534447834 38
	0.0010000	0.2000000	0.00082716152113828594498 21	0.00082716152113828594498 23
	0.0001000	0.3000000	0.00008377932630234771012 03	0.00008377932630234771012 05
	0.0000100	0.4000000	0.00000842787822604776866 22	0.00000842787822604776866 24
(0.8, 0.8)	0.0100000	0.1000000	0.00770680928534916816310 66	0.00770680928534916816310 68
	0.0010000	0.2000000	0.00080973915738776286119 18	0.00080973915738776286119 20
	0.0001000	0.3000000	0.00008213815144135411092 39	0.00008213815144135411092 41
	0.0000100	0.4000000	0.00000826953495943635215 44	0.00000826953495943635215 46
(0.9, 0.9)	0.0100000	0.1000000	0.00724744619451910297126 82	0.00724744619451910297126 84
	0.0010000	0.2000000	0.00076477351432498208920 61	0.00076477351432498208920 63
	0.0001000	0.3000000	0.00007770044531167807455 52	0.00007770044531167807455 54
	0.0000100	0.4000000	0.00000782921453314313777 45	0.00000782921453314313777 47
(1.0, 1.0)	0.0100000	0.1000000	0.00658863171819542622392 20	0.00658863171819542622392 21
	0.0010000	0.2000000	0.00069916310805349410454 39	0.00069916310805349410454 40
	0.0001000	0.3000000	0.00007115658116892684987 57	0.00007115658116892684987 58
	0.0000100	0.4000000	0.00000717585689446226225 71	0.00000717585689446226225 72

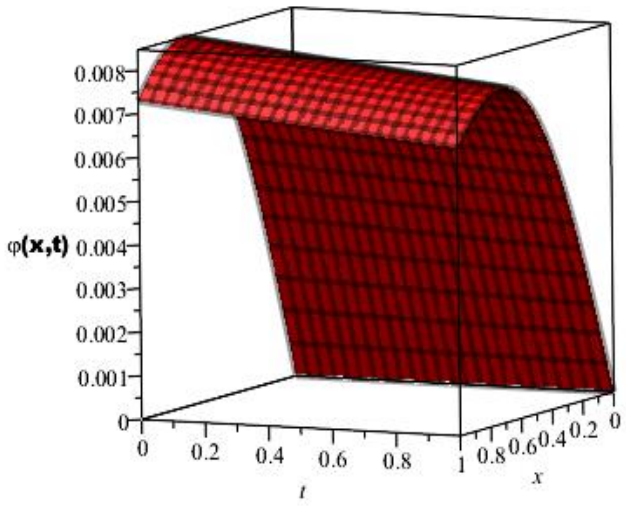

Figure.1 Numerical solutions φ(x,t) on 3D plots when source term λ=0.100 is greater than kinematic viscosity v=0.01

Figure 1. depict 3Dplot for velocity $\varphi(x,t)$ profile when $\lambda > v$

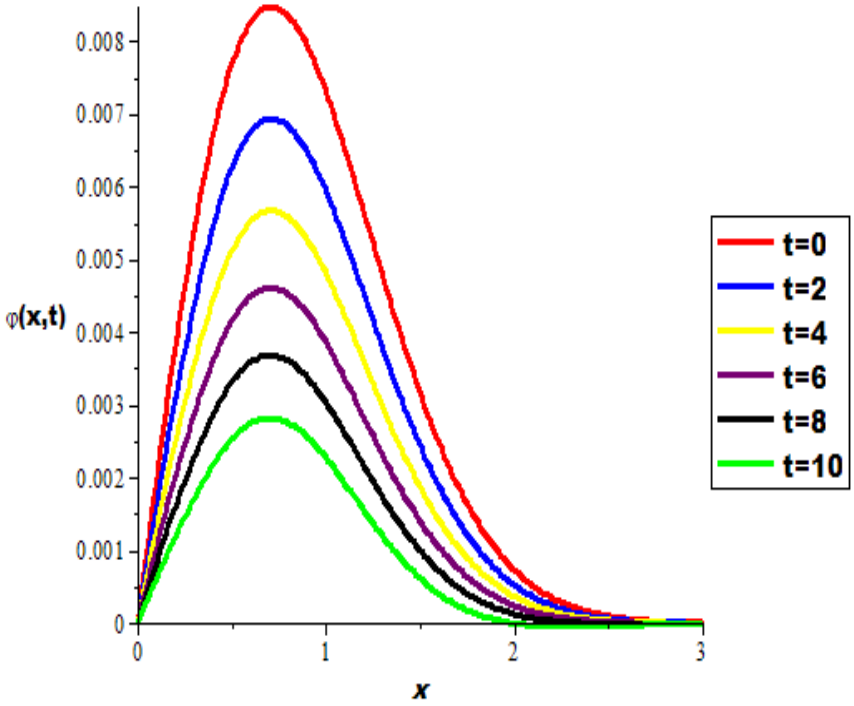

Figure 2. Numerical solutions φ(x,t) on 2D plots time interval 0≤t≤10 When source term λ=0.100 is greater than kinematic viscosity v=0.01

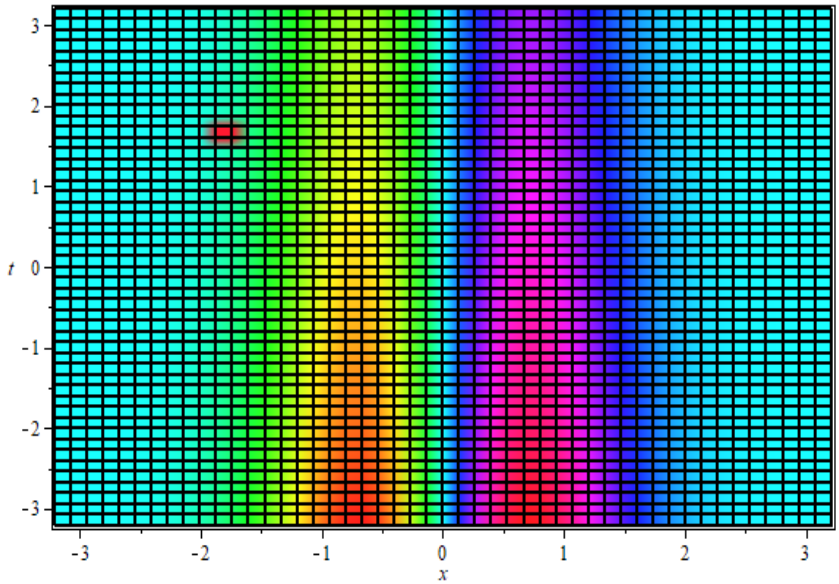

Figure 3. Density plots of x space coordinate on ta given time when source term λ=0.100 is greater than kinematic viscosity v=0.01

Table 2. Numerical solution $\varphi(x,t)$ when kinematic viscosity v greater than source term λ

(x,t)	λ source term	v kinematic viscosity	$\varphi(x,t)$ Analytical solutions $v > \lambda$	$\varphi(x,t)$ Algorithm solutions $v > \lambda$
(0,0)	0.0100000	0.1000000	0.000000000000000000000000	0.000000000000000000000000
	0.0010000	0.2000000	0.000000000000000000000000	0.000000000000000000000000
	0.0001000	0.3000000	0.000000000000000000000000	0.000000000000000000000000
	0.0000100	0.4000000	0.000000000000000000000000	0.000000000000000000000000
(0.1, 0.1)	0.0100000	0.1000000	0.016407022894644080299719 5	0.016407022894644080299720 0
	0.0010000	0.2000000	0.008972473552912830631516 5	0.008972473552912830631517 0
	0.0001000	0.3000000	0.006174309493603526048875 0	0.006174309493603526048875 5
	0.0000100	0.4000000	0.004706530422119724387589 3	0.004706530422119724387589 8
(0.2, 0.2)	0.0100000	0.1000000	0.031813039959280339399926 8	0.031813039959280339399927 3
	0.0010000	0.2000000	0.017406799447183025863356 1	0.017406799447183025863356 6
	0.0001000	0.3000000	0.011979739393427971038823 3	0.011979739393427971038823 8
	0.0000100	0.4000000	0.009132377572460050265249 4	0.009132377572460050265249 9
(0.3, 0.3)	0.0100000	0.1000000	0.045596991817186256678875 7	0.045596991817186256678876 1
	0.0010000	0.2000000	0.024904474035463347737492 1	0.024904474035463347737492 5
	0.0001000	0.3000000	0.017125708771418741810272 5	0.017125708771418741810272 1
	0.0000100	0.4000000	0.013049298130298764383015 6	0.013049298130298764383015 9
(0.4, 0.4)	0.0100000	0.1000000	0.057258387125447124923482 8	0.057258387125447124923483 2
	0.0010000	0.2000000	0.031145698060141965482494 1	0.031145698060141965482494 5
	0.0001000	0.3000000	0.021379426796171213212814 9	0.021379426796171213212815 3
	0.0000100	0.4000000	0.016274622587876773717876 9	0.016274622587876773717877 3
(0.5, 0.5)	0.0100000	0.1000000	0.066391372043335683776252 6	0.066391372043335683776253 0
	0.0010000	0.2000000	0.035892132321611553571553 8	0.035892132321611553571554 1
	0.0001000	0.3000000	0.024573283661787394657274 1	0.024573283661787394657274 5
	0.0000100	0.4000000	0.018679288556545162509228 8	0.018679288556545162509229 2
(0.6, 0.6)	0.0100000	0.1000000	0.072663191602533830112012 4	0.072663191602533830112012 8
	0.0010000	0.2000000	0.038987643636071314350786 0	0.038987643636071314350786 4
	0.0001000	0.3000000	0.026608636721932159374244 0	0.026608636721932159374244 0
	0.0000100	0.4000000	0.020192011131754067441784 6	0.020192011131754067441785 0
(0.7, 0.7)	0.0100000	0.1000000	0.075809526377497499187511 4	0.075809526377497499187511 7
	0.0010000	0.2000000	0.040361942153800919169221 6	0.040361942153800919169221 9
	0.0001000	0.3000000	0.027458316558523598625617 1	0.027458316558523598625617 5
	0.0000100	0.4000000	0.020800872689238910703232 1	0.020800872689238910703232 5
(0.8, 0.8)	0.0100000	0.1000000	0.075658429708917308614778 7	0.075658429708917308614779 0
	0.0010000	0.2000000	0.040038946553818200957746 7	0.040038946553818200957747 0
	0.0001000	0.3000000	0.027168349465533574203416 7	0.027168349465533574203417 0
	0.0000100	0.4000000	0.020552569690185607653126 8	0.020552569690185607651266 0
(0.9, 0.9)	0.0100000	0.1000000	0.072193745600898301868014 0	0.072193745600898301868014 3
	0.0010000	0.2000000	0.038147792213893003176341 9	0.038147792213893003176342 2
	0.0001000	0.3000000	0.025857347624649695660152 5	0.025857347624649695660152 7
	0.0000100	0.4000000	0.019548473055350362935924 3	0.019548473055350362935924 5

(1.0, 1.0)	0.0100000	0.1000000	0.065658447950442229784374 6	0.065658447950442229784374 6
	0.0010000	0.2000000	0.034929733165043497581687 8	0.034929733165043497581687 9
	0.0001000	0.3000000	0.023710423260481942962768 4	0.023710423260481942962768 5
	0.0000100	0.4000000	0.017936076856817250759779 0	0.017936076856817250759779 1

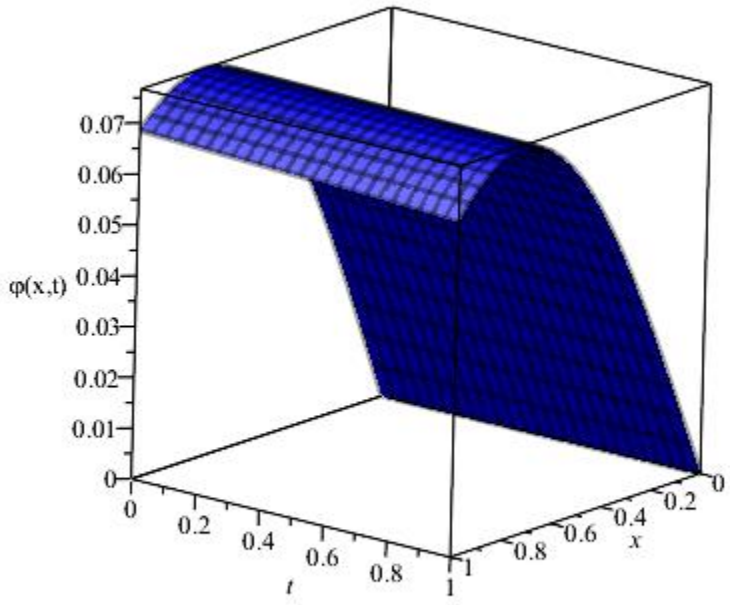

Figure. 4 Numerical solutions φ(x,t) on 3D plots when kinematic viscosity v=0.100 is greater than source term λ =0.01

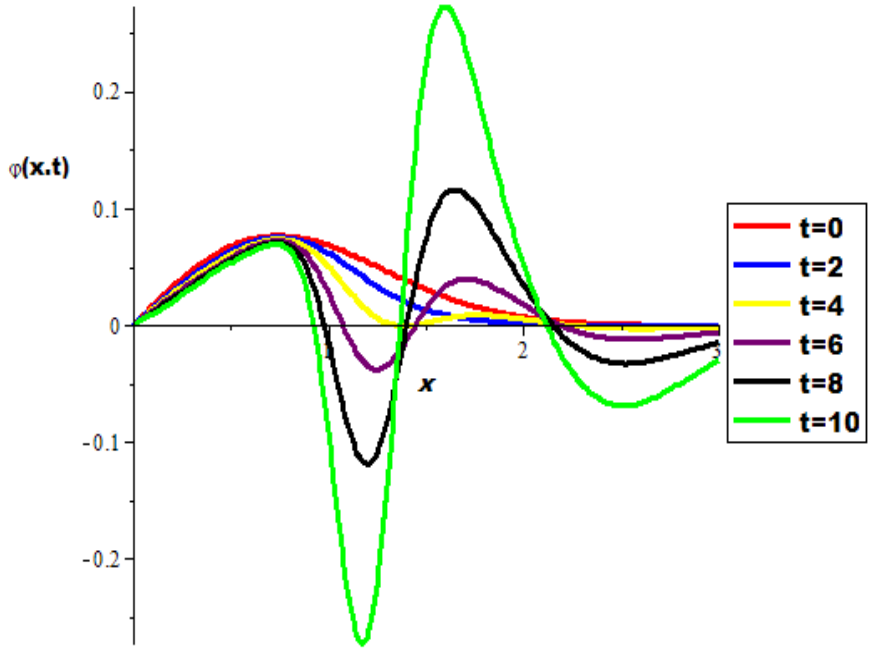

Figure 5. Numerical solutions φ(x,t) on 2D plots time interval 0≤t≤10 when kinematic viscosity v=0.100 is greater than source term λ=0.01

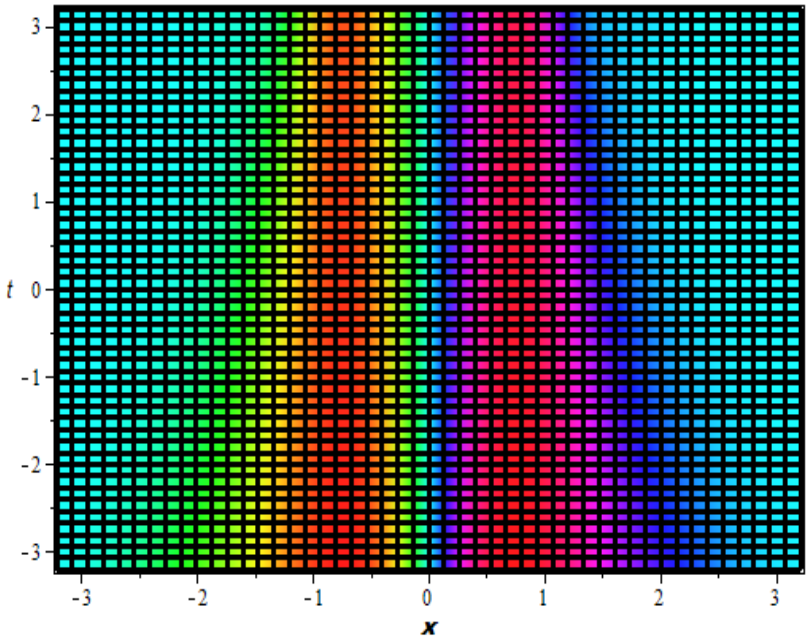

Figure 6 Density plots of x space coordinate on a given time when kinematic viscosity v=0.100 is greater thanSource term λ=0.01

DISCUSSION AND CONCLUSION

In this paper, we proposed and applied a computational algorithm for the numerical solutions of Burgers'equation occur with source. We axamined and obtained velocity $\varphi(x,t)$ when kinematics viscosity v is less than λ source term and vice-versa. From the numerical solutions obtained, we observed the following:

i. Decrease in kinematics viscosity v and increase in source term λ yielded less velocity $\varphi(x,t)$ air flows on wave travelling space within the given domian Table 1.
ii. Increase in kinematics viscosity v and decrease in source term λ yielded less velocity $\varphi(x,t)$ air flows on wave travelling space within the given domian Table 2.
iii. More decrease are recorded in (i) compare to (ii).
iv. Figure 1 depicts 3Dplot for velocity $\varphi(x,t)$ profile when $\lambda > v$.
v. Figure 2 depicts 3Dplot for velocity $\varphi(x,t)$ profile when $v < \lambda$.
vi. Figure 3 depicts 2Dplot for velocity $\varphi(x,t)$ air flows on wave travelling on time interval $0 \leq t \leq 10$ for $\lambda > v$.
vii. Figure 4 depicts 2Dplot for velocity $\varphi(x,t)$ of air flows on wave travelling on time interval $0 \leq t \leq 10$ for $v < \lambda$.
viii. Figure 5 depicts desityplot for velocity $\varphi(x,t)$ of air flows on wave travelling at x space coordinate on a given time t for $\lambda > v$.
ix. Figure 5 depicts desityplot for velocity $\varphi(x,t)$ of air flows on wave travelling at x space coordinate on a given time t for$v < \lambda$.

Application of a new four steps algorithm for the numerical solution of Burger's equation with source term in aerodynamics theory was considered. We observed from the analytical and numerical solutions are in close proximity and good agreement. The computational solutions represents a prototype of real experimental situation in a well set laboratory enviroment and given

computational ideas of what is obtainable in computational engineering. Thus, the proposed algorithm is hereby subjected to further study of Burgers'equation in gas dynamics, traffic flow and quantum field theory.

REFERENCES

Abdou, M. A., & Soliman, A. A. (2005). Variational iteration method for solving Burger's and coupled Burger's equations. *Journal of computational and Applied Mathematics*, *181*(2), 245-251.

Burgers, J. M. (1948). A mathematical model illustrating the theory of turbulence. In *Advances in applied mechanics* (Vol. 1, pp. 171-199). Elsevier.

Burgers, J.M. (1939). Trans. R. Netherlands *Acad. Sci. Amster-dam, 17,* 1.

Chandrasekharan Nair, L., & Awasthi, A. (2019). Quintic trigonometric spline based numerical scheme for nonlinear modified Burgers' equation. *Numerical Methods for Partial Differential Equations*, *35*(3), 1269-1289.

Desai, K. R., & Pradhan, V. H. (2012). Solution of Burger's equation and coupled Burger's equations by Homotopy perturbation method. *International Journal of Engineering Research and Applications*, *2*(3), 2033-2040.

Falade, K. I., & Tiamiyu, A.T. (2020). Numerical solution of partial differential equations with fractional variable coefficients using new iterative method (NIM), *I.J. Mathematical Sciences and Computing, 3,* 12-21.

Ismail, H. N., Raslan, K., & Abd Rabboh, A. A. (2004). Adomian decomposition method for Burger's–Huxley and Burger's–Fisher equations. *Applied mathematics and computation*, *159*(1), 291-301.

Mayur, P. A., Lakshmı, C., Vıjıtha, M. & Aswın, V.S. (2018). A systematic literature review of Burgers' equation with recent advances, *Pramana – J. Phys, 69.*

Mittal, R., & Jain, R. (2021). Solving Burgers' equation based on collocation of the modified cubic B-splines over finite elements, *Appl. Math. Comput, 218,* 7839.

Mohamed, N.A. (2019). Solving one- and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes, *Arab journal of basıc and applıed scıences, 26*(1), 254–268.

Nieuwstadt, F. T., & Steketee, J. A. (Eds.). (2012). *Selected papers of JM Burgers*. Springer Science & Business Media.

Panayotounakos, D. E., & Drikakis, D. (1995). On the closed-form solutions of the wave, diffusion and Burgers equations in fluid mechanics. *ZAMM-Journal of Applied*

Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, *75*(6), 437-447.

APPENDICES

Appendix A:

For $\lambda > v$

Step 1:

```
restart;
Digits := 23;
N := 4 :
c[0] := 1/2;
v := 0.01;
λ := 0.1;
IBC := x / (1 + (1/c[0]) · e^(x^2) / (4·v));
φ[0] := IBC;
```

Step 2:

```
BPDE := value(-(φ[0])·Diff((φ[0]), x) + v·Diff((φ[0]), x)·x − λ·(φ[0]));
φ[1] := value(Int(BPDE, t = 0 ..t));
for m from 1 to N do
 φ[m + 1] := value(Int(−sum(φ[n], n = 0 ..m)·Diff(sum(φ[n], n = 0 ..m), x) + v
     ·Diff(sum(φ[n], n = 0 ..m), x)·x − λ·sum(φ[n], n = 0 ..m), t = 0 ..t) − Int(
     −sum(φ[n], n = 0 ..m − 1)·Diff(sum(φ[n], n = 0 ..m − 1), x) + v·Diff(sum(φ[n], n
     = 0 ..m − 1), x)·x − λ·sum(φ[n], n = 0 ..m − 1), t = 0 ..t))
 end do:
 BDPE* := evalf(sum(φ[k], k = 0 ..N + 1)) :
 for i from 0 by 0.1 to 1 do φ[i] := evalf(eval(BDPE*, [x = i, t = i])); end do;
φ[1] := value(Int(BPDE, t = 0 ..t));
```

Step 3:

```
 sol1 := BDPE* :
 M[0] := eval(sol1, t = 0) :
 M[1] := eval(sol1, t = 2) :
 M[2] := eval(sol1, t = 4) :
 M[3] := eval(sol1, t = 6) :
 M[4] := eval(sol1, t = 8) :
 M[5] := eval(sol1, t = 10) :
```

Step 4:

```
with(plots);
plot3d(sol1, x = 0 ..1, t = 0 ..1, grid = [100, 100], color = "red")
```

```
plot([M[0],M[1],M[2],M[3],M[4],M[5]],x=0..3,color=[red,blue,yellow,purple,
    black,green],axes=BOXED,title="BURGER WITH SOURCE TERMS v<λ")

densityplot(sol1,x=-Pi..Pi,t=-Pi..Pi,axes=boxed,colorstyle=HUE);
```

$$Digits := 23$$

$$c_0 := \frac{1}{2}$$

$$v := 0.01$$

$$\lambda := 0.1$$

$$IBC := \frac{x}{1 + 50.000000000000000000000\, e^{x^2}}$$

$$\varphi_0 := \frac{x}{1 + 50.000000000000000000000\, e^{x^2}}$$

$$BPDE := -\frac{1}{1. + 50.000000000000000000000\, e^{x^2}}\left(x\left(\frac{1}{1. + 50.000000000000000000000\, e^{x^2}} - \frac{100.000000000000000000000\, x^2 e^{x^2}}{\left(1. + 50.000000000000000000000\, e^{x^2}\right)^2}\right)\right) + 0.01\left(\frac{1}{1. + 50.000000000000000000000\, e^{x^2}} - \frac{100.000000000000000000000\, x^2 e^{x^2}}{\left(1. + 50.000000000000000000000\, e^{x^2}\right)^2}\right)x - \frac{0.1\, x}{1. + 50.000000000000000000000\, e^{x^2}}$$

$$\varphi_1 := -\frac{1.\, x\left(\frac{1}{1. + 50.\, e^{x^2}} - \frac{100.\, x^2 e^{x^2}}{\left(1. + 50.\, e^{x^2}\right)^2}\right)t}{1. + 50.\, e^{x^2}} + 0.010000000000000000000000\left(\frac{1}{1. + 50.\, e^{x^2}} - \frac{100.\, x^2 e^{x^2}}{\left(1. + 50.\, e^{x^2}\right)^2}\right)x\, t - \frac{0.100000000000000000000000\, x\, t}{1. + 50.\, e^{x^2}}$$

$$\varphi_0 := 0.$$

$$\varphi_{0.1} := 0.00192057919778510200525 77$$
$$\varphi_{0.2} := 0.00369014573791845208800 74$$
$$\varphi_{0.3} := 0.00521610288002794011238 78$$
$$\varphi_{0.4} := 0.00642875308570518696654 64$$
$$\varphi_{0.5} := 0.00728536289409613896628 42$$
$$\varphi_{0.6} := 0.00777148973026395193061 48$$
$$\varphi_{0.7} := 0.00789982060625534447834 38$$
$$\varphi_{0.8} := 0.00770680928534916816310 68$$
$$\varphi_{0.9} := 0.00724744619451910297126 81$$
$$\varphi_{1.0} := 0.00658863171819542622392 21$$