Ultra Scientist Vol. 25(1)A, 156-168 (2013).

Linear Method For Two Dimensional Burgers Equation

SACHIN S.WANI[1] and SARITA THAKAR[2]

(Acceptance Date 22nd February, 2013)

Abstract

A Linear method is constructed for numerical solution of two dimensional non-linear Burgers equation. The scheme is derived from Crank-Nicolson finite difference scheme for linear terms and averaging for nonlinear terms. The method is shown to be consistent and second order accurate in time and space. The numerical solutions are obtained for two test problems at different time t and Reynolds number Re. The numerical solutions are compared with exact solution and other existing methods. Though the method is linear numerical solutions are compatible with ADM and Crank-Nicolson method.

Key words : Two dimensional Burgers equation, Crank Nicolson, Reynolds Number, Consistent.

AMS Subject classification:-65M12

1. Introduction

Burgers equation is an important partial differential equation in fluid dynamics. It has many applications in a variety of physical and engineering areas such as modeling of dynamics, heat conduction, acoustic waves ,investigating the shallow water waves[1,2], in examining the chemical reaction ,diffusion model of Brusselator[3], model of turbulence[4,5] and approximate theory of flow through a shock wave travelling in viscous fluid[6]. In 1995 S.E. Esipov derived coupled Burgers equation as a simple model of polydispersive sedimentation or evaluation of scaled volume concentrations of two kinds of particle in fluid suspensions[7]. The two dimensional Burges equation is same as the incompressible Navier Stokes equations without the pressure gradient terms. This constitutes an appropriate model for developing computational algorithms for solving the incompressible Navier Stokes equations.

In 1983 Fletcher J.D developed a procedure for generating exact solutions for two dimensional Burgers equation[8]. In 1983 Fletcher C.J obtained numerical solutions of one and two dimensional Burgers equation by finite difference method and numerical solutions are compared with exact solutions. Finite difference method is proved to be suitable for testing computational algorithms for solving two dimensional Burgers equation[9]. In 1992 F.W. Wubs, E.D.de Goede developed an explicit –implicit method for a class of time dependent partial differential equations[10]. J.H. He *et al*.[11], M.A. Abdau, A.A. Soliman[12] used variational iteration method to solve two dimensional Burgers equation. In 2000 Solimann[13] used the similarity reductions for the partial differential equations to develop a scheme for two dimensional Burgers equaton. In 2001 D.Kaya gave explicit solution for coupled viscous Burgers equation by decomposition method[14]. In 2004 S.F. Radwan used high order accurate schemes for solving the unsteady two dimensional Burgers equation. They also discussed the fourth order accurate two point compact scheme and the fourth order accurate Dulfort-Frankel scheme[15]. J. Ouyang, L-zhang, X-H-Zhang gave nonstandard element free Galerkin method for solving unsteady convection dominated problems[16]. They proposed element free characteristics Galerkin Methods[17] and a variational multiscale element free Galerkin Methods[18] for two dimensional Burgers equation. In 2008 Xiaonan Wu Jiewei-Zhang gave numerical solution of two dimensional Burgers equation in unbounded domains by artificial boundary method. This method is based on the Hopf Cole transformation and fourier series expansion.They obtain the exact boundary condition and series of approximating boundary conditions on the artificial boundary. The original problem is reduced to an equivalent problem on the bounded domain[19]. In 2008 D.L. Young C.M. Fan Etal Proposed the Eulerian Lagrange's method of fundamental solutions for two dimensional unsteady Burgers equation[20]. The method is meshless and is applicable for irregular domains. In 1978 P.C. Jain, D,N. Holla gave technique for two dimensional Burgers equation[21]. A. Refik Bahadir[22] used fully implicit discretization where as V.K. Shrivastava *et al*.[23] proposed Crank Nicolson scheme to solve two dimensional Burgers equation.Both these method leads to difference equations.Newtons Method is used to solve these nonlinear systems.In 2010 Hangging zhu,Huazhang shu Meiyu Ding[24] proposed Adomian Decomposition method to solve two dimensional Burgers equation. Above methods[21,22,23,24] are either implicit or nonlinear and are solved by Newton's method. Since methods are non linear evaluation of solution is time consuming.

In this paper we construct a linear finite difference method for two dimensional Burgers equation by approximating second order terms by using Crank Nicolson Scheme and nonlinear terms by central differences. The method is shown to be consistent and second order in time and space. Two test problems are solved for different initial and boundary conditions at different time t and different Reynolds number Re. In example one it is observed that the numerical solutions are compatible with exact solutions. In the second example numerical solutions obtained by linear method are compared with numerical solutions of existing methods[21, 22, 23, 24].

The paper is arranged as follows. In section 2 we developed the finite difference scheme and the scheme is shown to be consistent and second order accurate in time and space. In section 3 numerical solutions of two test examples are obtained by linear methods and results are compared with analytic solution and numerical solutions of other existing methods[21,22,23,24].

2. Finite Difference Scheme for Two Dimensional Nonlinear Burgers Equation:

Consider two dimensional Burgers eqution

$$u_t + uu_x + vu_y = \frac{1}{Re}\left(u_{xx} + u_{yy}\right)$$

$$v_t + uv_x + vv_y = \frac{1}{Re}\left(v_{xx} + v_{yy}\right) \tag{2.1}$$

With intial conditions

$$u(x,y,0) = f(x,y)\,,(x,y) \in D$$

$$v(x,y,0) = g(x,y)\,,(x,y) \in D \quad (2.2)$$

And boundary conditions

$$u(x,y,t) = f_1(x,y,t) \quad ,x,y \in \partial D, t > 0$$

$$v(x,y,t) = g_1(x,y,t), x,y \in \partial D, t > 0 \quad (2.3)$$

Where $D = \{(x,y), a \le x \le b, a \le y \le b\}$ and ∂D is its boundary $u(x,y,t)$ and $v(x,y,t)$ are the velocity components to be determined $f, g, f_1 \& g_1$ are known functions and Re is the Reynolds number. The computational Domain D is discretized with uniform mesh $x_i = a + i\Delta x, y_j = a + j\Delta y$ where $\Delta x = \frac{b-a}{n_x}$, $\Delta y = \frac{b-a}{n_y}$. Denote the approximations of $u(x,y,t)$ and $v(x,y,t)$ at mesh points $(a + i\Delta x, a + j\Delta y, n\Delta t)$ by $u_{i,j}^n$ and $v_{i,j}^n$ resp $(i = 0,1,2 - -n_x, j = 0,1,2 - - - n_y, n = 0,1,2 - - -)$ and Δt represents the increment in time.

We approximate u_t and v_t by forward difference, partial derivatives in non-linear terms are replaced by central difference operator at $t = t_n$ and $t = t_{n+1}$. where as second order derivatives are approximated by usual Crank Nicolson expression.

Then discretization of equation (2.1) gives

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + u_{i,j}^{n+1}\frac{u_{i+1,j}^n - u_{i-1,j}^n}{4\Delta x}$$
$$+ u_{i,j}^n\frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{4\Delta x}$$
$$+ v_{i,j}^{n+1}\frac{u_{i,j+1}^n - u_{i,j-1}^n}{4\Delta y}$$
$$+ v_{i,j}^n\frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{4\Delta y}$$
$$= \frac{1}{2Re(\Delta x)^2}\left[u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1} + u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n\right]$$
$$+ \frac{1}{2Re(\Delta y)^2}\left[u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1} + u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n\right]$$

And

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n+1}\frac{v_{i+1,j}^{n}-v_{i-1,j}^{n}}{4\Delta x}+u_{i,j}^{n}\frac{v_{i+1,j}^{n+1}-v_{i-1,j}^{n+1}}{4\Delta x}+v_{i,j}^{n+1}\frac{v_{i,j+1}^{n}-v_{i,j-1}^{n}}{4\Delta y}+v_{i,j}^{n}\frac{v_{i,j+1}^{n+1}-v_{i,j-1}^{n+1}}{4\Delta y}$$
$$=\frac{1}{2Re(\Delta x)^2}\left[v_{i+1,j}^{n+1}-2v_{i,j}^{n+1}+v_{i-1,j}^{n+1}+v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}\right]+\frac{1}{2Re(\Delta y)^2}\left[v_{i,j+1}^{n+1}-2v_{i,j}^{n+1}+v_{i,j-1}^{n+1}+v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}\right] \quad (2.4)$$

Define $\frac{\Delta t}{4\Delta x}=k_1, \frac{\Delta t}{4\Delta y}=k_2, \frac{\Delta t}{2Re(\Delta x)^2}=r_1, \frac{\Delta t}{2Re(\Delta y)^2}=r_2$

Rearrangement of equation (2.4) gives

$$u_{i+1,j}^{n+1}(k_1u_{i,j}^{n}-r_1)+u_{i,j}^{n+1}\left[1+k_1(u_{i+1,j}^{n}-u_{i-1,j}^{n})+2r_1+2r_2\right]+u_{i-1,j}^{n+1}(-k_1u_{i,j}^{n}-r_1)+u_{i,j+1}^{n+1}(k_2v_{i,j}^{n}-r_2)+u_{i,j-1}^{n+1}(-k_2v_{i,j}^{n}-r_2)+v_{i,j}^{n+1}(u_{i,j+1}^{n}-u_{i,j-1}^{n})k_2$$
$$=(1-2r_1-2r_2)u_{i,j}^{n}+u_{i+1,j}^{n}r_1+u_{i-1,j}^{n}r_1+u_{i,j+1}^{n}r_2+u_{i,j-1}^{n}r_2$$

And

$$v_{i+1,j}^{n+1}(k_1u_{i,j}^{n}-r_1)+v_{i,j}^{n+1}\left[1+k_2(v_{i,j+1}^{n}-v_{i,j-1}^{n})+2r_1+2r_2\right]+v_{i-1,j}^{n+1}(-k_1u_{i,j}^{n}-r_1)+v_{i,j+1}^{n+1}(k_2v_{i,j}^{n}-r_2)+v_{i,j-1}^{n+1}(-k_2v_{i,j}^{n}-r_2)+u_{i,j}^{n+1}(v_{i+1,j}^{n}-v_{i-1,j}^{n})k_1$$
$$=(1-2r_1-2r_2)v_{i,j}^{n}+v_{i+1,j}^{n}r_1+v_{i-1,j}^{n}r_1+v_{i,j+1}^{n}r_2+v_{i,j-1}^{n}r_2 \quad (2.5)$$

We prove that the scheme given by (2.5) is consistent and is of order two in space time. The truncation error at $t=t_n$, $x=x_i$ & $y=y_j$ for u and v is given by

$$T_{i,j}^{n+1}(x)=u_t+uu_x+vu_y-\frac{1}{Re}(u_{xx}+u_{yy})-\frac{u(x_i,y_j,t_{n+1})-u(x_i,y_j,t_n)}{\Delta t}-\frac{1}{4\Delta x}\{u(x_i,y_j,t_n)[u(x_{i+1},y_j,t_{n+1})-u(x_{i-1},y_j,t_{n+1})]$$

$$
\begin{aligned}
&+ u(x_i, y_j, t_{n+1})[u(x_{i+1}, y_j, t_n) \\
&- u(x_{i-1}, y_j, t_n)]\} \\
&- \frac{1}{4\Delta y}\{v(x_i, y_j, t_n)[u(x_i, y_{j+1}, t_{n+1}) \\
&- u(x_i, y_{j-1}, t_{n+1})] \\
&+ v(x_i, y_j, t_{n+1})[u(x_i, y_{j+1}, t_n) \\
&- u(x_i, y_{j-1}, t_n)]\} \\
&+ \frac{1}{2Re(\Delta x)^2}[u(x_{i+1}, y_j, t_{n+1}) \\
&- 2u(x_i, y_j, t_{n+1}) + u(x_{i-1}, y_j, t_{n+1}) \\
&+ u(x_{i+1}, y_j, t_n) - 2u(x_i, y_j, t_n) \\
&+ u(x_{i-1}, y_j, t_n)] \\
&+ \frac{1}{2Re(\Delta y)^2}[u(x_i, y_{j+1}, t_{n+1}) \\
&- 2u(x_i, y_j, t_{n+1}) + u(x_i, y_{j-1}, t_{n+1}) \\
&+ u(x_i, y_{j+1}, t_n) - 2u(x_i, y_j, t_n) \\
&+ u(x_i, y_{j-1}, t_n)]
\end{aligned}
$$

And

$$
\begin{aligned}
&T_{i,j}^{n+1}(y) \\
&= v_t + vv_y + uv_x - \frac{1}{Re}(v_{xx} + v_{yy}) \\
&- \frac{v(x_i, y_j, t_{n+1}) - v(x_i, y_j, t_n)}{\Delta t} \\
&- \frac{1}{4\Delta x}\{u(x_i, y_j, t_{n+1})[v(x_{i+1}, y_j, t_n) \\
&- v(x_{i-1}, y_j, t_n)] \\
&- u(x_i, y_j, t_n)[v(x_{i+1}, y_j, t_{n+1}) \\
&- v(x_{i-1}, y_j, t_{n+1})]\} \\
&- \frac{1}{4\Delta y}\{v(x_i, y_j, t_{n+1})[v(x_i, y_{j+1}, t_n) \\
&- v(x_i, y_{j-1}, t_n)] \\
&+ v(x_i, y_j, t_n)[v(x_i, y_{j+1}, t_{n+1}) \\
&- v(x_i, y_{j-1}, t_{n+1})]\} \\
&+ \frac{1}{2Re(\Delta x)^2}[v(x_{i+1}, y_j, t_{n+1}) \\
&- 2v(x_i, y_j, t_{n+1}) + v(x_{i-1}, y_j, t_{n+1}) \\
&+ v(x_{i+1}, y_j, t_n) - 2v(x_i, y_j, t_n) \\
&+ v(x_{i-1}, y_j, t_n)] \\
&+ \frac{1}{2Re(\Delta y)^2}[v(x_i, y_{j+1}, t_{n+1}) \\
&- 2v(x_i, y_j, t_{n+1}) + v(x_i, y_{j-1}, t_{n+1}) \\
&+ v(x_i, y_{j+1}, t_n) - 2v(x_i, y_j, t_n) \\
&+ v(x_i, y_{j-1}, t_n)]
\end{aligned}
$$

On expanding above expression in Taylors Series about (x_i, y_j, t_n) we get

$$T_{i,j}^{n+1}(x) = \left[u_t + uu_x + vu_y - \frac{1}{Re}(u_{xx} + u_{yy})\right]_t \frac{\Delta t}{2} + o((\Delta x)^2, (\Delta y)^2, (\Delta t)$$

and

$$
\begin{aligned}
T_{i,j}^{n+1}(y) = \Big[v_t + uv_x + vv_y \\
- \frac{1}{Re}(v_{xx} + v_{yy})\Big]_t \frac{\Delta t}{2} \\
+ o((\Delta x)^2, (\Delta y)^2, (\Delta t)^2)
\end{aligned}
$$

Thus along (2.1) the truncation error

$T_{i,j}^{n+1}(x)$ and $T_{i,j}^{n+1}(y)$ is of $o((\Delta x)^2, (\Delta y)^2, (\Delta t)^2)$. Thus difference scheme for two dimensional Burgers equation is consistent and is of order two.

3. Numerical Results

Numerical solutions of two dimensional non-linear Burgers equation (2.1) are obtained by linear method given by equation (2.5) for two different initial and boundary conditions. The numerical solutions obtained by (2.5) are compared with numerical solutions of existing methods[21,22,23,24] and exact solution. The comparison of Numerical solutions obtained by (2.5) for example 1 with ADM and exact solution is given in Table 1 and 2. In table 3,4,5,6 the error in numerical solution obtained by (2.5) are compared with the error in numerical solution given by [21,22]. It is observed from tables 3,4,5 and 6 that the method (2.5) gives more accurate solutions than fully implicit method[21] and are compatible with ADM method[24]. Table 7,8,9 and 10 shows the comparison of solution of (2.1) by Linear method (2.5) and Crank Nicolson scheme for different values of t and Reynolds number Re. The comparison of numerical solution of Burgers Equation by (2.5) for example 2 with existing methods[21,22,23,24] is shown in tables (11, 12, 13, 14).

Example 1: Exact solution of two dimensional Burgers equation (2.1) using Hopf Cole transformation is given in[8] is as follows

$$u(x,y,t) = \frac{3}{4} - \frac{1}{4(1+e^{(Re(-t-4x+4y)/32)})}$$

$$v(x,y,t) = \frac{3}{4} + \frac{1}{4(1+e^{(Re(-t-4x+4y)/32)})}$$

The initial and boundary conditions are taken form exact solution. The computational domain for this problem is $D = \{(x,y), 0 \le x \le 1, 0 \le y \le 1\}$. The numerical computations are performed using uniform mesh $\Delta x = \Delta y = 0.05, \Delta t = 0.001$. Tables (1-8) gave comparison of numerical solution with exact and other methods at different time t and different Reynolds number Re.

Example 2: in this example computational domain is taken as

$$D = \{(x,y), 0 \le x \le 0.5, 0 \le y \le 0.5\}$$

and Burgers equation (2.1) considered with initial conditions

$$\left.\begin{array}{r} u(x,y,0) = sin(\pi x) + cos(\pi y) \\ v(x,y,0) = x + y \end{array}\right\} 0 \le x \le 0.5, 0 \le y \le 0.5, t \ge 0$$

And boundary conditions

$$\left.\begin{array}{r} u(0,y,t) = cos(\pi y), u(0.5,y,t) = sin(\pi x) \\ v(x,0,t) = x, v(x,0.5,t) = x + 0.5 \end{array}\right\} 0 \le x \le 0.5, 0 \le y \le 0.5, t \ge 0$$

This example is solved with 20×20 mesh and Δt=0.0001 Re=50 *at* t=0.625 and results are compared with other existing methods[21,22,23,24]. The solution are listed in[11-14] for comparison.

Table1. Comparison of Linear method (2.5) with Adomian Decomposition method and exact solution at t=0.05,t=0.2,t=0.5 for u(x,y,t) for Re=80,N=20

	t=0.05			t=0.2			t=0.5		
MeshPoint	**ADM**	**EXACT**	**LINEAR**	**ADM**	**EXACT**	**LINEAR**	**ADM**	**EXACT**	**LINEAR**
(0.1,0.1)	0.61733	0.61720	0.617329	0.59465	0.59438	0.594561	0.55601	0.55567	0.555590
(0.9,0.2)	0.50020	0.50020	0.500199	0.50013	0.50014	0.500134	0.50004	0.50006	0.500063
(0.8,0.3)	0.50147	0.50148	0.501467	0.50098	0.50102	0.500987	0.50029	0.50048	0.500456
(0.7,0.4)	0.51046	0.51052	0.510459	0.50714	0.50733	0.507149	0.50277	0.50352	0.503315
(0.9,0.5)	0.50395	0.50398	0.503948	0.50266	0.50275	0.502669	0.50086	0.50130	0.501215
(0.1,0.6)	0.74810	0.74811	0.748101	0.74723	0.74725	0.747236	0.74435	0.74426	0.744226
(0.8,0.6)	0.52658	0.52667	0.526577	0.51867	0.51896	0.518652	0.50922	0.50933	0.508877
(0.3,0.7)	0.74491	0.74492	0.744907	0.74264	0.74267	0.742630	0.73527	0.73498	0.734960
(0.4,0.7)	0.73663	0.73665	0.736635	0.73102	0.73103	0.731014	0.71370	0.71299	0.713278
(0.2,0.8)	0.74930	0.74930	0.749298	0.74897	0.74898	0.748973	0.74789	0.74786	0.747827
(0.6,0.8)	0.71677	0.71676	0.716772	0.70457	0.70439	0.704574	0.67009	0.66979	0.670679
(0.1,0.9)	0.74990	0.74990	0.749905	0.74986	0.74986	0.749861	0.74971	0.74971	0.749705
(0.9,0.9)	0.61733	0.61720	0.617334	0.59465	0.59438	0.594662	0.55601	0.55567	0.555420

Table2.Comparison of Linear method (2.5) with Adomian Decomposition method and exact solution at t=0.05,t=0.2,t=0.5 for v(x,y,t) for Re=80,N=20

	t=0.05			t=0.2			t=0.5		
MeshPoint	**ADM**	**EXACT**	**LINEAR**	**ADM**	**EXACT**	**LINEAR**	**ADM**	**EXACT**	**LINEAR**
(0.1,0.1)	0.88267	0.88280	0.882671	0.90534	0.90561	0.905439	0.94399	0.94432	0.94441
(0.9,0.2)	0.99980	0.9980	0.999801	0.99987	0.99986	0.999866	0.99996	0.99993	0.999937
(0.8,0.3)	0.99853	0.99852	0.998533	0.99902	0.998998	0.999013	0.99971	0.99952	0.999544
(0.7,0.4)	0.98954	0.98948	0.989541	0.99286	0.992670	0.992851	0.99722	0.99648	0.996685
(0.9,0.5)	0.99605	0.99602	0.996052	0.99734	0.99725	0.997331	0.99913	0.99869	0.998785
(0.1,0.6)	0.75190	0.75189	0.751899	0.75277	0.75275	0.752764	0.75565	0.75574	0.755774
(0.8,0.6)	0.97342	0.97333	0.973423	0.98133	0.98103	0.981348	0.99078	0.99067	0.991123
(0.3,0.7)	0.75509	0.75508	0.755093	0.75736	0.75733	0.75737	0.76473	0.76502	0.76504
(0.4,0.7)	0.76336	0.76335	0.763365	0.76898	0.76896	0.768986	0.78630	0.78701	0.786722
(0.2,0.8)	0.75070	0.75070	0.750702	0.75103	0.75102	0.751027	0.75211	0.75214	0.752173
(0.6,0.8)	0.78323	0.78324	0.783228	0.79543	0.79561	0.795426	0.82991	0.83020	0.829321
(0.1,0.9)	0.75009	0.75009	0.750095	0.75014	0.75014	0.750139	0.75029	0.75029	0.750295
(0.9,0.9)	0.88267	0.88280	0.882666	0.90534	0.90561	0.905338	0.94399	0.94432	0.944578

Table 3.Comparison of absolute errors for u(x,y,t) at Re=100, at t=0.01

MeshPoint	**ADM**	**BAHADIR**	**LINEAR**	**EXACT**
(0.1,0.1)	5.91368E-5	7.24132E-5	5.91472E-5	0.62305
(0.5,0.1)	4.84030E-6	2.42869E-5	4.84480E-6	0.50162
(0.9,0.1)	3.41000E-8	8.39751E-6	3.42106E-8	0.50001
(0.3,0.3)	5.91368E-5	8.25331E-5	5.91380E-5	0.62305
(0.7,0.3)	4.84030E-6	3.43163E-5	5.0678E-6	0.50162
(0.1,0.5)	1.64290E-6	5.62014E-5	1.63327E-6	0.74827
(0.5,0.5)	5.91368E-5	7.32522E-5	5.91538E-5	0.62305

Table 4. Comparison of absolute errors for u(x,y,t) at Re=100, at t=0.5.

MeshPoint	**ADM**	**BAHADIR**	**LINEAR**	**EXACT**
(0.1,0.1)	2.77664E-4	5.13431E-4	3.21647E-4	0.54332
(0.5,0.1)	4.52081E-4	8.85712E-4	1.12088E-5	0.50035
(0.9,0.1)	3.37430E-6	6.53372E-5	8.3748E-8	0.50000
(0.3,0.3)	2.77664E-4	7.31601E-4	6.3383E-4	0.54332
(0.7,0.3)	4.52081E-4	6.27245E-4	3.55900E-5	0.50035
(0.1,0.5)	2.86553E-4	4.01942E-4	6.38835E-5	0.74221
(0.5,0.5)	2.77664E-4	3.46823E-4	8.17693E-4	0.54332

Table 5. Comparison of absolute errors for v(x,y,t) at Re=100,$\Delta t = 0.001$ at t=0.01.

MeshPoint	**ADM**	**BAHADIR**	**LINEAR**	**EXACT**
(0.1,0.1)	5.91368E-5	8.35601E-5	5.91472E-5	0.87695
(0.5,0.1)	4.84030E-6	5.13642E-5	4.8448E-6	0.99838
(0.9,0.1)	3.41000E-8	7.03298E-6	3.42106E-8	0.99999
(0.3,0.3)	5.91368E-5	6.15201E-5	5.91538E-5	0.87695
(0.7,0.3)	4.84030E-6	5.41000E-5	5.0670E-6	0.99838
(0.1,0.5)	1.64290E-6	7.35192E-5	1.63327E-6	0.75173
(0.5,0.5)	5.91368E-5	8.51040E-5	5.91538E-5	0.87695

Table 6. Comparison of absolute errors for v(x,y,t) at Re=100,$\Delta t = 0.001$ at t=0.5.

MeshPoint	**ADM**	**BAHADIR**	**LINEAR**	**EXACT**
(0.1,0.1)	2.77664E-4	6.17325E-4	3.21647E-4	0.95668
(0.5,0.1)	4.52081E-4	4.67046E-4	1.12088E-5	0.99965
(0.9,0.1)	3.37400E-6	1.70434E-5	8.3748E-8	1.00000
(0.3,0.3)	2.77664E-4	6.25402E-4	6.3383E-4	0.95668
(0.7,0.3)	4.52081E-4	4.66046E-4	3.55900E-5	0.99965
(0.1,0.5)	2.86553E-4	8.72422E-4	6.38835E-5	0.75779
(0.5,0.5)	2.77664E-4	6.23291E-4	8.17693E-4	0.95668

Table 7. Comparison table with Re=500,$\Delta t = 0.001$ at t=0.5 &t=2 for u(x,y,t)

	t=0.5			t=2		
MeshPoint	CNS	LINEAR	EXACT	CNS	LINEAR	EXACT
(0.1,0.1)	0.48372	0.487322	0.50010	0.49725	0.497208	0.50000
(0.5,0.1)	0.50002	0.500015	0.50000	0.50025	0.50027	0.50000
(0.9,0.1)	0.50000	0.500000	0.50000	0.49932	0.500152	0.50000
(0.3,0.3)	0.49531	0.495305	0.50010	0.50687	0.507043	0.50000
(0.7,0.3)	0.50001	0.500012	0.50000	0.49929	0.499878	0.50000
(0.1,0.5)	0.74990	0.749900	0.75000	0.43945	0.439928	0.50048
(0.5,0.5)	0.49438	0.494385	0.50010	0.49959	0.499825	0.50000
(0.9,0.5)	0.49977	0.499980	0.50000	0.51376	0.501287	0.50000
(0.3,0.7)	0.75001	0.750008	0.75000	0.41654	0.415846	0.50048
(0.7,0.7)	0.49324	0.494292	0.50010	0.51050	0.503249	0.50000
(0.1,0.9)	0.75000	0.750000	0.75000	0.75003	0.749429	0.75000
(0.5,0.9)	0.75001	0.750002	0.75000	0.42895	0.423373	0.50048
(0.9,0.9)	0.47380	0.493362	0.50010	0.56293	0.507622	0.50000

Table 8 Comparison table with Re=500,$\Delta t = 0.001$ at t=0.5 &t=2 for v(x,y,t)

	t=0.5			t=2		
MeshPoint	CNS	LINEAR	EXACT	CNS	LINEAR	EXACT
(0.1,0.1)	1.01286	1.01268	0.99990	1.00276	1.002790	1.00000
(0.5,0.1)	1.0000	0.999985	1.00000	0.99976	0.999730	1.00000
(0.9,0.1)	1.0000	1.00000	1.00000	1.00068	0.999848	1.00000
(0.3,0.3)	1.00481	1.00469	0.99990	0.99313	0.992957	1.00000
(0.7,0.3)	0.99999	0.999988	1.00000	1.00072	1.000120	1.00000
(0.1,0.5)	0.75010	0.750103	0.75000	1.06055	1.060070	0.99952
(0.5,0.5)	1.00571	1.00561	0.99990	1.00041	1.000170	1.00000
(0.9,0.5)	1.00022	1.00002	1.00000	0.98624	0.998713	1.00000
(0.3,0.7)	0.74999	0.749992	0.75000	1.08346	1.084150	0.99952
(0.7,0.7)	1.00676	1.00571	0.99990	0.98950	0.996751	1.00000
(0.1,0.9)	0.75000	0.75000	0.75000	0.74997	0.750571	0.75000
(0.5,0.9)	0.74999	0.749998	0.75000	1.07105	1.076630	0.99952
(0.9,0.9)	1.02620	1.00664	0.99990	0.93707	0.992378	1.00000

Table 9. Comparison table with Re=500, $\Delta t = 0.01$ at t=0.5 &t=2 for u(x,y,t)

	t=0.5			t=2		
MeshPoint	CNS	LINEAR	EXACT	CNS	LINEAR	EXACT
(0.1,0.1)	0.48714	0.487239	0.50010	0.49729	0.497244	0.50000
(0.5,0.1)	0.50002	0.500150	0.50000	0.50024	0.500267	0.50000
(0.9,0.1)	0.50000	0.500000	0.50000	0.49934	0.500152	0.50000
(0.3,0.3)	0.49519	0.495246	0.50010	0.50690	0.507083	0.50000
(0.7,0.3)	0.50001	0.500012	0.50000	0.49928	0.499877	0.50000
(0.1,0.5)	0.74990	0.749900	0.75000	0.43939	0.439812	0.50048
(0.5,0.5)	0.49429	0.494354	0.50010	0.49951	0.499728	0.50000
(0.9,0.5)	0.49978	0.493344	0.50000	0.51355	0.501296	0.50000
(0.3,0.7)	0.75001	0.750008	0.75000	0.41647	0.415613	0.50048
(0.7,0.7)	0.49325	0.494264	0.50010	0.51008	0.503158	0.50000
(0.1,0.9)	0.75000	0.750000	0.75000	0.75004	0.749420	0.75000
(0.5,0.9)	0.75001	0.750002	0.75000	0.42909	0.422900	0.50048
(0.9,0.9)	0.47275	0.493344	0.50010	0.56275	0.507666	0.50000

Table 10. Comparison table with Re=500, $\Delta t = 0.01$ at t=0.5 &t=2 for v(x,y,t)

	t=0.5			t=2		
MeshPoint	CNS	LINEAR	EXACT	CNS	LINEAR	EXACT
(0.1,0.1)	1.01286	1.012760	0.99990	1.00271	1.002760	1.00000
(0.5,0.1)	0.99999	0.999985	1.00000	0.99976	0.999733	1.00000
(0.9,0.1)	1.00000	1.000000	1.00000	1.00066	0.999848	1.00000
(0.3,0.3)	1.00481	1.004750	0.99990	0.99310	0.992917	1.00000
(0.7,0.3)	0.99999	0.999988	1.00000	1.00072	1.000120	1.00000
(0.1,0.5)	0.75010	0.750100	0.75000	1.06061	1.060190	0.99952
(0.5,0.5)	1.00571	1.005650	0.99990	1.00049	1.000270	1.00000
(0.9,0.5)	1.00022	1.000020	1.00000	0.98646	0.998704	1.00000
(0.3,0.7)	0.74999	0.749992	0.75000	1.08353	1.084390	0.99952
(0.7,0.7)	1.00676	1.005740	0.99990	0.98992	0.996842	1.00000
(0.1,0.9)	0.75000	0.750000	0.75000	0.74996	0.750580	0.75000
(0.5,0.9)	0.74999	0.749998	0.75000	1.07091	1.077100	0.99952
(0.9,0.9)	1.02725	1.006660	0.99990	0.93725	0.992334	1.00000

Table 11. Comparison of computed values of u for Re=50 at t=0.625,N=20

MeshPoints	CNS	A.R.BAHADIR	JAIN &HOLLA	LINEAR
(0.1,0.1)	0.97146	0.96688	0.97258	0.971461
(0.3,0.1)	1.15280	1.14827	1.16214	1.152820
(0.2,0.2)	0.86307	0.85911	0.86281	0.863072
(0.4,0.2)	0.97981	0.97637	0.96483	0.979813
(0.1,0.3)	0.66316	0.66019	0.66318	0.663157
(0.3,0.3)	0.77230	0.76932	0.77030	0.772297
(0.2,0.4)	0.58180	0.57966	0.58070	0.581799
(0.4,0.4)	0.75856	0.75678	0.74435	0.758558

Table 12. Comparison of computed values of v for Re=50 at t=0.625,N=20

MeshPoints	CNS	A.R.BAHADIR	JAIN &HOLLA	LINEAR
(0.1,0.1)	0.09869	0.09824	0.09773	0.0986881
(0.3,0.1)	0.14158	0.14112	0.14039	0.141582
(0.2,0.2)	0.16754	0.16681	0.16660	0.167542
(0.4,0.2)	0.17110	0.17065	0.17397	0.171095
(0.1,0.3)	0.26378	0.26261	0.26294	0.263781
(0.3,0.3)	0.22654	0.22576	0.22463	0.226539
(0.2,0.4)	0.32851	0.32745	0.32402	0.328508
(0.4,0.4)	0.32500	0.32441	0.31822	0.324997

Table 13. Comparison of computed values of u for Re=500 at t=0.625,N=20

MeshPoints	CNS	A.R.BAHADIR	JAIN &HOLLA	LINEAR
(0.15,0.1)	0.96870	0.96650	0.95691	0.968969
(0.3,0.1)	1.03200	1.02970	0.95616	1.03202
(0.1,0.2)	0.86178	0.84449	0.84257	0.846187
(0.2,0.2)	0.87814	0.87631	0.86399	0.878141
(0.1,0.3)	0.67920	0.67809	0.67667	0.679202
(0.3,0.3)	0.79947	0.79792	0.76876	0.799471
(0.15,0.4)	0.66036	0.54601	0.54408	0.546743
(0.2,0.4)	0.58959	0.58874	0.58778	0.589589

Table 14. Comparison of computed values of v for Re=500 at t=0.625,N=20

MeshPoints	CNS	A.R.BAHADIR	JAIN &HOLLA	LINEAR
(0.15,0.1)	0.09043	0.09020	0.10177	0.0923034
(0.3,0.1)	0.10728	0.10690	0.13287	0.107275
(0.1,0.2)	0.17295	0.17972	0.18503	0.180103
(0.2,0.2)	0.16816	0.16777	0.18169	0.168157
(0.1,0.3)	0.26268	0.26222	0.26560	0.262677
(0.3,0.3)	0.23550	0.23497	0.25142	0.235501
(0.15,0.4)	0.29019	0.31753	0.32084	0.317991
(0.2,0.4)	0.30419	0.30371	0.30927	0.304187

Conclusion

A Linear method is presented to construct numerical solution of two dimensional nonlinear Burgers equation. The scheme is proved to be consistent and second order accurate in time and space. This method is better than existing methods[21,22] since the scheme is a linear the computation time is less than the time requires for other nonlinear schemes. The two test problems are solved for comparison with exact & numerical solution of other existing methods. The solution obtain by linear methods is compatible with Adomian Decomposition method and Crank Nicolson methods[23,24].

References

1. J.D. Logan, "An Introduction to nonlinear partial differential equations," Wily-Interscience, New York (1994).
2. L. Debtnath, "Nonlinear Partial equations for scientist and engineers" Birkhauser, Boston, (1997).
3. G. Adomian, "The diffusion - Brusselator equation", *Computational Math. Appl. 29,* 1-3 (1995).
4. Burger J.M., "A Mathematical model illustrating the theory of Turbulence", *Advances in Applied Mathematics, 3,* 201-230 (1950).
5. J.M. Burger, "A Mathematical model illustrating the theory of Turbulence", *Adv Appl. Mech I,* 171-199 (1948).
6. J.D. Cole, "On a Quasilinear parabolic equations occurring in aerodynamics", *Quart. Appl. Math 9,* 225-236 (1951).
7. S.E. Esipov, "Coupled Burgers equations a model of polydispersive sedimentation", *Phys. RevE-52,* 3711-3718 (1995).
8. J.D. Flether, "Generating exact solutions of the two dimensional Burgers Equation", *Int. J. Numer. Math. Fluid 3,* 213-216 (1983).
9. C.J. Fletcher, A comparison of finite element and finite difference solutions of one and two dimensional Burgers equation", *J. Comput. Phys. 51,* 159-188 (1983).
10. F.W. Wubs, E.D.de Goede "An explicit-implicit method for a class of time-dependent partial differential equations *Appl. Numer.*

Math, 9, 157-181 (1992).

11. J.H. He, “A new approach to nonlinear partial differential equations”, *Comm. Nonlinear science Numer.* Simul. *2*, 230-235 (1997).
12. M.A. Abdou, A.A. Soliman, “Variational iteration method for solving Burgers and coupled Burgers equations”, *J. Comput. Appl. Math, 181(2),* 245-251 (2005).
13. A.A. Solimann, “New numerical technique for Burgers equation based on similarity reductions”, International conference on computational fluid dynamics Beijing China, October Vol. 1720, 559-566 (2000).
14. D. Kaya, “An explicit solution of coupled viscous Burgers equation by decomposition method”, *Int. J. Math. Math. Sci, 27,* 675-680 (2001).
15. S.F. Radwan, “Comparision of higher order accurate schemes for solving the *two dimensional unsteady Burgers equation”, J. Comput. Appl. Math,174,* 383-397 (2004).
16. J. Ouyang, L. Zhang, X-H Zhang, “Nonstandard element free-Galerkin method for solving unsteady convection dominated problems”, *Acta Aerodyn, sin 25,* 287-293 (2007).
17. X-H Zhang, J-Ouyang I zhang “Element free characteristics Galerkin method for Burgers equation”, *Eng. Anal .Boundary Elem 33*, 356-362 (2009).
18. Lin Zhang. J.E. Ouyang Xiaoxia Wang, Xianhua Zhang “Variational multiscale element free Galerkin method for two dimensional Burgers equation”, *Journal of computational physics, 229,* 7147-7161 (2010).
19. Xiaonan Wu, Jiwei Zhang, “Artificial boundary method for two dimensional Burgers equation” *Computers and mathematics with applications, 56,* 242-256 (2008).
20. D.L. Young, C.M. Fan S.P. Hu S N Atluri, “The Eulerian lagrangian method of fundamental solutions for two dimensional unsteady Burgers equation”, *Eng Anal Boundary element 32,* 395-412 (2008).
21. P.C. Jain, D.N. Holla, “Numerical solution of coupled Burgers equation”, *Int. J. Numer. Math. Eng, 12,* 213-222 (1978).
22. A. Refik Bahadir, “A fully implicit finite difference scheme for two dimensional Burgers equation”, *Applied Mathematics and computation, 137,* 131-137 (2003).
23. Vineet Kumar *etal*., “Crank Nicolson scheme for numerical solution of two dimensional coupled Burgers equation”, *International journal of scientific and Engineering Resarch* Volume *2,* Issue *5,* May (2011).
24. Hongqing zhu, Huazhang shu, Meiyu Ding, “Numerical solution of two dimensional Burgers equation by discrete Adomian decomposition method”, *Computers and mathematics with applications, 60,* 840-848 (2010).