Application of the Non-Polynomial Spline Approach to the Solution of the Burgers' Equation

Mohamed A. Ramadan*[,1], Talaat S. El-Danaf[1] and Faisal E.I. Abd Alaal[2]

[1]*Department of Mathematics, Faculty of Science, Menoufia University, Shebeen El-Koom, Egypt*

[2]*Graduate Student with the Department*

Abstract: In this paper, we propose a non-polynomial spline based method to develop a numerical method for approximation to the Burgers' equation. Applying the Von-Neumann stability analysis, we show that the proposed method is unconditionally stable. A numerical example is given to illustrate the applicability and the accuracy of the presented new method.

Keywords: Non-polynomial spline, nonlinear burgers' equation, von neumann stability, accuracy.

INTRODUCTION

Consider the Burgers' equation of the form:

$$u_t + uu_x - \nu\, u_{xx} = 0, \quad a \le x \le b, \quad t \ge 0 \tag{1}$$

subject to the conditions

$$\begin{aligned} &u(a,t) = \beta_1(t),\ u(b,t) = \beta_2(t),\\ &u_x(a,t) = \lambda_1, u_x(b,t) = \lambda_2, \quad t \ge 0 \end{aligned} \tag{2}$$

The last two additional conditions in (2) are true at the initial time for $0 < \nu < 1$, so we suppose that it is true for any time; the initial condition takes the form:

$$u(x,0) = f(x), \qquad a \le x \le b \tag{3}$$

The study of this Burgers' equation is important due to its application in the approximate theory of flow through a shock wave propagating in a viscous fluid [1] and in the modeling of turbulence [2]. In the past few years a great deal of efforts has been expended to study the equation (1) as well as other forms of this partial differential equation, both theoretically and numerically, see for example, [3-11].

Recently, there is a wide use of non-polynomial splines based methods for approximating the solution of boundary value problems of different orders, see for example, [12-15]. However, the numerical analysis of literature contains little for using these non-polynomial splines dealing with the solution of partial differential equations [16,17].

In this paper, we are concerned with the problem of applying non-polynomial spline functions to develop a numerical method for obtaining approximation for the solution for non-linear Burgers' equation (1). The non-polynomial spline function in this work has a trigonometric part, and a polynomial part of first degree.

*Address correspondence to this author at the Department of Mathematics, Faculty of Science, Menoufia University, Shebeen El-Koom, Egypt; E-mail: mramadan@mailer.eun.eg

Remark

The C^{∞} - differentiability of the trigonometric part of non-polynomial spline compensates for the loss of smoothness inherited by polynomial splines.

The paper is organized as follows: In section 2, a new method which depends on the use of the non-polynomial splines is derived. In section 3, the stability analysis is theoretically discussed. Using Von Neuman method, for given values of specified parameters, the proposed method is shown to be unconditionally stable. Finally, in section 4, a numerical example is included to illustrate the practical implementation of the proposed method. The accuracy performance of the obtained numerical results is compared with the exact solution. Since we are using a new type of (moving) boundary conditions to improve the accuracy, therefore we can not make an accuracy comparison with other exciting methods for this problem. The numerical results show that our proposed method is a promising approach for solving different types of this nonlinear partial differential equation problem.

DERIVATION OF THE NUMERICAL METHOD FOR THE NON-POLYNOMIAL SPLINE APPROACH TO BURGERS' EQUATION

To set up the non-polynomial spline method, select an integer $N > 0$ and time-step size $k > 0$.

With $h = \dfrac{b-a}{N+1}$, the mesh points (x_i, t_j) are $x_i = a + ih$, for each $i = 0,1,\ldots,N+1$,

and $t_j = jk$, for each $j = 0,1,\ldots$.

Let $U_i^j \equiv U(x_i,t_j)$ be an approximation to $u_i^j \equiv u(x_i,t_j)$, obtained by the segment $P_i(x,t_j)$ of the mixed spline function passing through the points (x_i, U_i^j) and (x_{i+1}, U_{i+1}^j). Each segment has the form:

1874-1142/07 2007 Bentham Science Publishers Ltd.

$$P_i(x,t_j) = a_i(t_j)\cos\omega(x-x_i) + b_i(t_j)$$
$$\sin\omega(x-x_i) + c_i(t_j)(x-x_i) + d_i(t_j) \tag{4}$$

for each $i = 0,1,\ldots,N$. To obtain expressions for the coefficients of (4) in terms of U_i^j, U_{i+1}^j, S_i^j, and S_i^{j+1}, we first define

$$P_i(x_i,t_j) = U_i^j,\ \ P_i(x_{i+1},t_j) = U_{i+1}^j,$$
$$P_i^{(2)}(x_i,t_j) = S_i^j, and\ \ P_i^{(2)}(x_{i+1},t_j) = S_{i+1}^j \tag{5}$$

Using the Eqns. (4) and (5), we get:

$$a_i + d_i = U_i^j,$$
$$a_i\cos\theta + b_i\sin\theta + c_i h + d_i = U_{i+1}^j \tag{6}$$
$$-a_i\omega^2 = S_i^j$$
$$-a_i\omega^2\cos\theta - b_i\omega^2\sin\theta = S_{i+1}^j$$

where,
$a_i \equiv a_i(t_j),\ b_i \equiv b_i(t_j),\ c_i \equiv c_i(t_j),\ d_i \equiv d_i(t_j)$,

and $\theta = \omega h$. By solving the last four equations, we obtain the following expressions:

$$a_i = -\frac{h^2}{\theta^2}S_i^j, b_i = \frac{h^2\left(\cos\theta\, S_i^j - S_{i+1}^j\right)}{\theta^2\sin\theta}$$
$$c_i = \frac{\left(U_{i+1}^j - U_i^j\right)}{h} + \frac{h\left(S_{i+1}^j - S_i^j\right)}{\theta^2}, d_i = \frac{h^2}{\theta^2}S_i^j + U_i^j, \tag{7}$$

Using the continuity condition of the first derivative at $x = x_i$, that is, $P_i^{(1)}(x_i,t_j) = P_{i-1}^{(1)}(x_i,t_j)$, we get the following relation:

$$b_i\omega + c_i = -a_{i-1}\omega\sin\theta + b_{i-1}\omega\cos\theta + c_{i-1} \tag{8}$$

Using (7), equation (8) gives us the following tridiagonal system:

$$U_{i+1}^j - 2U_i^j + U_{i-1}^j = \alpha S_{i+1}^j + \beta S_i^j + \alpha S_{i-1}^j,.$$

For $i = 1,2,\ldots,N$ (9)

Where,

$$\alpha = \frac{h^2}{\theta\sin\theta} - \frac{h^2}{\theta^2}, \beta = -\frac{2h^2\cos\theta}{\theta\sin\theta} + \frac{2h^2}{\theta^2}$$

and

$$S_i^j = \frac{\partial^2 U_i^j}{\partial x^2} = \frac{1}{\nu}\left(\frac{\partial U_i^j}{\partial t} + (U_i^j)\frac{\partial U_i^j}{\partial x}\right).$$

Replacing j by $j+1/2$, system (9) becomes:

$$U_{i+1}^{j+1/2} - 2U_i^{j+1/2} + U_{i-1}^{j+1/2} =$$
$$\alpha S_{i+1}^{j+1/2} + \beta S_i^{j+1/2} + \alpha S_{i-1}^{j+1/2},\quad i = 1,2,K,N. \tag{10}$$

where,

$$U_i^{j+1/2} \equiv U(x_i,t_{j+1/2}),\ t_{j+1/2} = \frac{t_{j+1}+t_j}{2}$$

and,

$$S_i^{j+1/2} = \frac{1}{\nu}\left(\frac{\partial U_i^{j+1/2}}{\partial t} + (U_i^{j+1/2})\frac{\partial U_i^{j+1/2}}{\partial x}\right).$$

Using the finite difference method, we obtain

$$U_i^{j+1/2} \approx \frac{U_i^{j+1}+U_i^j}{2},\quad \frac{\partial}{\partial t}U_i^{j+1/2} \approx \frac{U_i^{j+1}-U_i^j}{k},$$
$$and\ \frac{\partial}{\partial x}U_i^j \approx \frac{U_{i+1}^j - U_i^j}{h}. \tag{11}$$

Using these formulas allows us to express $S_i^{j+1/2}$ as,

$$S_i^{j+1/2} \approx \frac{1}{\nu k}\left(U_i^{j+1} - U_i^j\right) +$$
$$\frac{(U_i^{j+1/2})}{2\nu}\left(\frac{U_{i+1}^{j+1} - U_i^{j+1}}{h} + \frac{U_{i+1}^j - U_i^j}{h}\right) \tag{12}$$

The use of (11) and (12) in equation (10) gives us the following system:

$$A_iU_{i-1}^{j+1} + B_iU_i^{j+1} + C_iU_{i+1}^{j+1} + D_iU_{i+2}^{j+1} =$$
$$A_i^*U_{i-1}^j + B_i^*U_i^j + C_i^*U_{i+1}^j + D_i^*U_{i+2}^j,$$

For each $i = 1,2,3,\ldots,N-1$ $j = 0,1,2,\ldots$ (13)

where,

$$D_i = \frac{\alpha\delta_{i+1}}{2h\nu},\qquad D_i^* = \frac{-\alpha\delta_{i+1}}{2h\nu},$$
$$C_i = -\frac{1}{2} + \frac{\alpha}{\nu k} - \frac{\alpha\delta_{i+1}}{2\nu h} + \frac{\beta\delta_i}{2\nu h},\qquad C_i^* = \frac{1}{2} + \frac{\alpha}{\nu k} + \frac{\alpha\delta_{i+1}}{2\nu h} - \frac{\beta\delta_i}{2\nu h},$$
$$B_i = 1 + \frac{\beta}{\nu k} - \frac{\beta\delta_i}{2\nu h} + \frac{\alpha\delta_{i-1}}{2\nu h},\qquad B_i^* = -1 + \frac{\beta}{\nu k} + \frac{\beta\delta_i}{2\nu h} - \frac{\alpha\delta_{i-1}}{2\nu h},$$
$$A_i = -\frac{1}{2} + \frac{\alpha}{\nu k} - \frac{\alpha\delta_{i-1}}{2\nu h},\qquad A_i^* = \frac{1}{2} + \frac{\alpha}{\nu k} + \frac{\alpha\delta_{i-1}}{2\nu h},$$

and,

$$\delta_i = (U_i^{j+1/2})$$

System (13) consists of **N-1** equations in the unknowns U_i, $i = 1,\ldots,N+1$. To get a solution to this system we need 3-additional equations. These equations are obtained from the conditions in (2). The first two parts in (2) are replaced by:

$$U_0^j = \beta_1\left(t_j\right),\quad U_{N+1}^j = \beta_2\left(t_j\right),\quad j = 0,1,\ldots. \tag{14}$$

but the last part in (2) is discretized by the following equation:

$$U_{N-2}^j + U_{N-1}^j - 13U_N^j + 11U_{N+1}^j \approx$$
$$8h\frac{\partial}{\partial x}U_{N+1}^j = 8h\lambda_2\left(t_j\right),\ \ j \ge 0. \tag{15}$$

The last equation is true for any time. Writing Eqns. (13)-(15) in matrix form gives:

$$Q\,U^{j+1} = Q^*\,U^j + r^{j+1} \tag{16}$$

where,

$$U^j = \left(U_0^j,\ \ U_1^j,\ \ U_2^j,\ \ \ldots\ \ \ldots\ \ \ldots\ \ U_{N-1}^j,\ \ U_N^j,\ \ U_{N+1}^j\right)^T,$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots\ldots & 0 \\ A_1 & B_1 & C_1 & D_1 & 0 & 0 & 0 & \ldots\ldots & 0 \\ 0 & A_2 & B_2 & C_2 & D_2 & 0 & 0 & \ldots\ldots & 0 \\ 0 & O & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & 0 & A_{N-2} & B_{N-2} & C_{N-2} & D_{N-2} & 0 \\ 0 & \ldots\ldots & 0 & 0 & 0 & A_{N-1} & B_{N-1} & C_{N-1} & D_{N-1} \\ 0 & \ldots\ldots & 0 & 0 & 0 & 1 & 1 & -13 & 11 \\ 0 & \ldots\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$Q^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots\ldots & 0 \\ A_1^* & B_1^* & C_1^* & D_1^* & 0 & 0 & 0 & \ldots\ldots & 0 \\ 0 & A_2^* & B_2^* & C_2^* & D_2^* & 0 & 0 & \ldots\ldots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & 0 & A_{N-2}^* & B_{N-2}^* & C_{N-2}^* & D_{N-2}^* & 0 \\ 0 & \ldots\ldots & 0 & 0 & 0 & A_{N-1}^* & B_{N-1}^* & C_{N-1}^* & D_{N-1}^* \\ 0 & \ldots\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ldots\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and,

$r^j = \left(\beta_1\left(t_j\right), 0, 0, \ldots\ldots\ldots, 0, 8h\lambda_2\left(t_j\right), \beta_2\left(t_j\right)\right)^T$ where r^j is an (N+2) dimensional column vector, while *Q and Q** are (N+2)x (N+2) matrices. The initial condition $u(x,0) = f(x)$, for each $a \le x \le b$, implies that $U_i^0 = f(x_i)$, for each $i = 0,1,\ldots,N+1$. These values can be used in Eq. (13) to find the value of U_i^1, for each $i = 0,1,\ldots,N+1$. If the procedure is reapplied once all the approximations U_i^1 are known, the values of $U_i^2, U_i^3, \ldots$ can be obtained in a similar manner.

Remark 1

To cope with the nonlinear terms in System (13), the following steps are followed :

1- At $j = 0$, we approximate δ_i by δ_i^0 calculated from U_i^0 only , that is,

$\delta_i \approx \delta_i^0 = \left(U_i^0\right)$ *for each* $i = 0,1,\ldots,N$, which are needed to compute the elements

of *Q and Q**. We then obtain U^1 from (16).

2- At $j = 1$, we approximate δ_i by δ_i^1 calculated from $0.5\left(U_i^0 + U_i^1\right)$, that is,

$\delta_i \approx \delta_i^1 = \left(0.5\left(U_i^0 + U_i^1\right)\right)$ *for each* $i = 0,1,\ldots,N$,

which are needed to compute the elements of *Q and Q**. We then obtain U^2 from **(16).**

3- At $j = n$, we approximate δ_i by δ_i^n

calculated from $0.5\left(U_i^{n-1} + U_i^n\right)$, That is, $\delta_i \approx \delta_i^n = \left(0.5\left(U_i^{n-1} + U_i^n\right)\right)$ for each i=1,2,..N, which are needed to compute the elements of *Q and Q**. We then obtain U^{n+1} from (16).

THE STABILITY ANALYSIS

For stability analysis, we use the Von Neumann method. To do this, we must linearize the nonlinear term (uu_x) of the Burgers' equation (1) by making $\delta_{i+1} = \delta_i = \delta_{i-1} = d$ in the numerical scheme (16). According to the Von Neuman method we have:

$$U_i^j = \zeta^j \exp\left(q\varphi i h\right), \tag{17}$$

where φ is the mode number, $q = \sqrt{-1}$, h is the element size, and ζ is the amplification factor of the scheme. Substitute Eq. (17) into Eq. (13) we get:

$$\begin{aligned} &\zeta^{j+1}\begin{Bmatrix} A_i \exp\left((i-1)q\varphi h\right) + B_i \exp\left(iq\varphi h\right) + \\ C_i \exp\left((i+1)q\varphi h\right) + D_i \exp\left((i+2)q\varphi h\right) \end{Bmatrix} = \\ &\qquad \zeta^j \begin{Bmatrix} A_i^* \exp\left((i-1)q\varphi h\right) + B_i^* \exp\left(iq\varphi h\right) + \\ C_i^* \exp\left((i+1)q\varphi h\right) + D_i^* \exp\left((i+2)q\varphi h\right) \end{Bmatrix} \end{aligned} \tag{18}$$

where,

$$\begin{aligned} &D_i = \frac{\alpha d}{2hv}, && D_i^* = \frac{-\alpha d}{2hv}, \\ &C_i = -\frac{1}{2} + \frac{\alpha}{vk} - \frac{\alpha d}{2vh} + \frac{\beta d}{2vh}, && C_i^* = \frac{1}{2} + \frac{\alpha}{vk} + \frac{\alpha d}{2vh} - \frac{\beta d}{2vh}, \\ &B_i = 1 + \frac{\beta}{vk} - \frac{\beta d}{2vh} + \frac{\alpha d}{2vh}, && B_i^* = -1 + \frac{\beta}{vk} + \frac{\beta d}{2vh} - \frac{\alpha d}{2vh}, \\ &A_i = -\frac{1}{2} + \frac{\alpha}{vk} - \frac{\alpha d}{2vh}, && A_i^* = \frac{1}{2} + \frac{\alpha}{vk} + \frac{\alpha d}{2vh}, \end{aligned} \tag{19}$$

Dividing both sides of Eq. (18) by *exp* $\left(iq\varphi h\right)$ we obtain:

$$\begin{aligned} &\zeta^{j+1}\left\{A_i \exp\left(-q\varphi h\right) + B_i + C_i \exp\left(q\varphi h\right) + D_i \exp\left(2q\varphi h\right)\right\} = \\ &\zeta^j \left\{A_i^* \exp\left(-q\varphi h\right) + B_i^* + C_i^* \exp\left(q\varphi h\right) + D_i^* \exp\left(2q\varphi h\right)\right\} \end{aligned} \tag{20}$$

Using the following Euler's formula:

$$\exp[q\phi] = \cos\phi + q\sin\phi, \quad \phi = \varphi h \tag{21}$$

Eq. (20) can be represented in the form:

$$\zeta^{j+1}\left\{\begin{array}{l} A_i\left(\cos\phi - q\sin\phi\right) + B_i + C_i\left(\cos\phi + q\sin\phi\right) + \\ D_i\left(\cos 2\phi + q\sin 2\phi\right) \end{array}\right\} =$$

$$\zeta^{j}\left\{\begin{array}{l} A_i^*\left(\cos\phi - q\sin\phi\right) + B_i^* + C_i^*\left(\cos\phi + q\sin\phi\right) + \\ D_i^*\left(\cos 2\phi + q\sin 2\phi\right) \end{array}\right\}$$

After simple calculations, we obtain:

$$\zeta = \frac{X^* + qY^*}{X + qY} \tag{22}$$

where,

$$\begin{aligned} X^* &= \left(A_i^* + C_i^*\right)\cos\phi + D_i^*\cos 2\phi + B_i^*, \\ X &= \left(A_i + C_i\right)\cos\phi + D_i\cos 2\phi + B_i, \\ Y^* &= \left(C_i^* - A_i^*\right)\sin\phi + D_i^*\sin 2\phi \\ Y &= \left(C_i - A_i\right)\sin\phi + D_i\sin 2\phi \end{aligned} \tag{23}$$

Eqns. (19) and (23) together give:

$$\begin{aligned} X^* &= \frac{1}{\nu k}\left(\beta + 2\alpha\right) - \frac{4\alpha}{\nu k}\sin^2\frac{\phi}{2} - 2\sin^2\frac{\phi}{2} + \\ &\qquad \frac{d}{\nu h}\left(\beta - 2\alpha\right)\sin^2\frac{\phi}{2} + \frac{\alpha d}{\nu h}\sin^2\phi, \\ X &= \frac{1}{\nu k}\left(\beta + 2\alpha\right) - \frac{4\alpha}{\nu k}\sin^2\frac{\phi}{2} + 2\sin^2\frac{\phi}{2} - \\ &\qquad \frac{d}{\nu h}\left(\beta - 2\alpha\right)\sin^2\frac{\phi}{2} - \frac{\alpha d}{\nu h}\sin^2\phi, \\ Y^* &= -\frac{\beta d}{2\nu h}\sin\phi - \frac{\alpha d}{2\nu h}\sin 2\phi \\ Y &= \frac{\beta d}{2\nu h}\sin\phi + \frac{\alpha d}{2\nu h}\sin 2\phi \end{aligned} \tag{24}$$

The necessary and sufficient condition for (13) to be stable is:

$$|\zeta| = \sqrt{\frac{X^{*2} + Y^{*2}}{X^2 + Y^2}} \le 1$$

Simplifying the above inequality, we obtain:

$$X^2 \ge X^{*2} \tag{25}$$

where $Y^2 = Y^{*2}$.

The last inequality gives us:

$$\left[\frac{1}{\nu k}\left(\beta + 2\alpha\right) - \frac{4\alpha}{\nu k}\sin^2\frac{\phi}{2}\right]$$

$$\left[2\sin^2\frac{\phi}{2} - \frac{d}{\nu h}\left(\beta - 2\alpha\right)\sin^2\frac{\phi}{2} - \frac{\alpha d}{\nu h}\sin^2\phi\right] \ge 0 \tag{26}$$

For $\beta > 2\alpha$, $\beta > 0$, and $\alpha > 0$ we obtain:

$$\left[\frac{1}{\nu k}\left(\beta + 2\alpha\right) - \frac{4\alpha}{\nu k}\sin^2\frac{\phi}{2}\right] \ge 0$$

which enables us to write (26) in the form:

$$\left[2\sin^2\frac{\phi}{2} - \frac{d}{\nu h}\left(\beta - 2\alpha\right)\sin^2\frac{\phi}{2} - \frac{\alpha d}{\nu h}\sin^2\phi\right] \ge 0 \tag{27}$$

Our system is conditionally stable. The condition of stability is $\beta \ge 2\alpha$, $\alpha \to 0$ and $\beta \to 0$.

NUMERICAL RESULTS

We now obtain the approximate numerical solution of Burgers' equation for one standard problem. The accuracy of our proposed numerical method is measured by computing the difference between the analytic and numerical solutions at each mesh point, and use these differences to compute the L_2 and L_∞ error norms.

The analytic solution of the Burgers' equation (1) [8] is given by:

$$u(x,t) = \frac{(x/t)}{1 + (t/\sigma)^{1/2}\exp(x^2/4\nu t)},$$

Where, $0 \le x \le 1$, $t \ge 1$ (28)

Where $\sigma = \exp(1/8\nu)$ make initial condition to be the equation (28) evaluated at t = **1**. The boundary conditions are:

$$u(0,t) = 0, \quad u(1,t) = \frac{(1/t_j)}{1 + (t_j/\sigma)^{1/2}\exp(1/4\nu t_j)}, \tag{29}$$

Note that the second condition in Eq(28) is a type of moving boundary condition.

Remark 2

The additional conditions:

$u_x(a,t) = \frac{\partial}{\partial x}u\left(0,t_0\right)$ and $u_x(b,t) = \frac{\partial}{\partial x}u\left(1,t_0\right)$ are true at the initial time for $0 < \nu < 1$, so we suppose that the analytical solution (28) satisfies these conditions

The obtained numerical results are summarized in the following tables for $\Delta x = 0.025$ (Tables **1-4**). Table **1** gives the numerical and exact solutions at time **t =3**.

Table 1. $\nu = 0.05, \Delta t = 0.01, \alpha = 0.2\times 10^{-6}$ **and** $\beta = 6\times 10^{-4}$

x_i	Exact Solution	Numerical Solution
0.0	0.0000000	0.0000000
0.1	0.0221546	0.0220390
0.2	0.0435601	0.0433457
0.3	0.0634297	0.0631439
0.4	0.0809113	0.0805862
0.5	0.0950889	0.0947549
0.6	0.1050300	0.1047120
0.7	0.1099090	0.1096260
0.8	0.1092040	0.1089760
0.9	0.1029440	0.1028100
1.0	0.0918946	0.0918946

Table 2. $\nu = 0.05, \Delta t = 0.01, \alpha = 0.2 \times 10^{-6}$ and $\beta = 6 \times 10^{-4}$

Time	L_2 – error x 10^3	L_∞- error x 10^3
2	0.129883	0.203959
2.5	0.146748	0.235351
3	0.246727	0.334341
3.5	0.311609	0.428702
5	0.342211	0.482112

Table 3. $\nu = 0.5, \Delta t = 0.1, \alpha = 0.2 \times 10^{-6}$ *and* $\beta = 6 \times 10^{-4}$

Time	L_2 – error x 10^3	L_∞- error x 10^3
2	0.6625430	0.9333200
2.5	0.3534180	0.4977590
3	0.2141600	0.3015790
3.5	0.1422560	0.2002460
5	0.0574751	0.0808847

Table 4. $\nu = 0.005, \Delta t = 0.05, \alpha = 0.2 \times 10^{-9}$ *and* $\beta = 4 \times 10^{-4}$

Time	L_2 – error x 10^3	L_∞- error x 10^2
2	8.015390	2.187940
2.5	9.622790	2.618230
3	9.835130	2.848630
3.5	5.998950	1.912170
5	1.520370	0.228921

Remark

Using these new types of boundary conditions, (29) allows us to compute the approximate solution for large value of the time t with an acceptable accuracy. In previous existing methods for Burgers' equation, numerical solutions are computed for the time t with $t \ll 5$.

Next, we draw some of the obtained approximate solutions $U(x,t)$ for this test problem versus the distance x. Figs. (**1**) and (**2**) illustrate the behavior of the numerical solution at $\nu = 0.05$, $\Delta t = 0.01$, $\Delta x = 0.025$ but Figs. (**3**) and (**4**) illustrate the behavior of the numerical solution at $\nu = 0.005$, $\Delta t = 0.05$, $\Delta x = 0.025$ for some different times t =2 and 3 respectively.

From Figs. (**1**), (**3**) at t =2 and Figs (**2**), (**4**) at t =3, we can conclude that as í in the dispersion term νu_{xx}, increases from 0.005 to 0.05, the effect of the nonlinearity term, the second term uu_x of Burgers' equation (1), decreases.

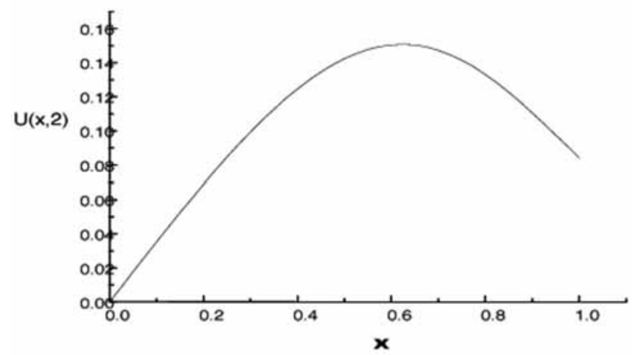

Fig. (1).

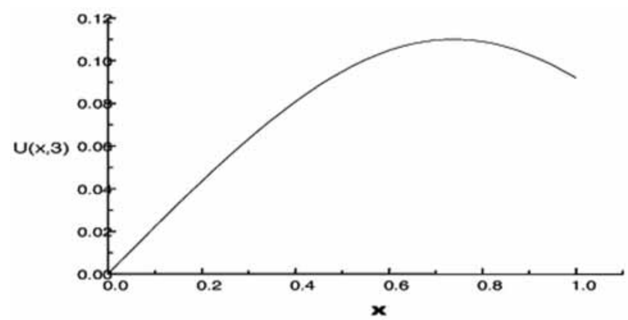

Fig. (2).

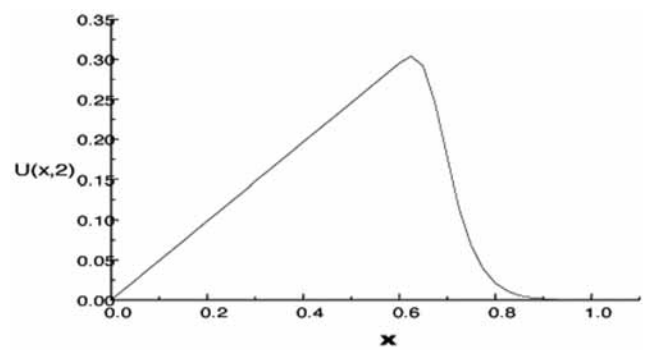

Fig. (3).

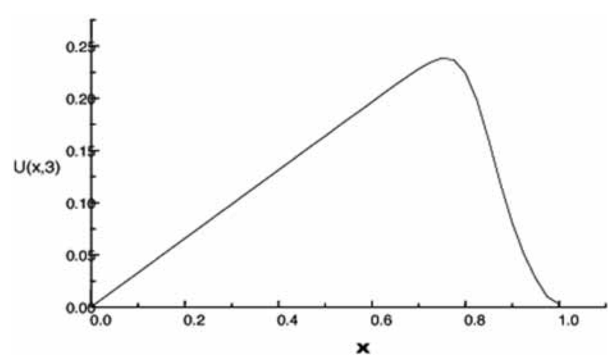

Fig. (4).

CONCLUSION

In this paper, a numerical treatment for the Burgers' equation using non-polynomial spline is proposed. The stability analysis of the method is shown to be unconditionally stable for given values of specified parameters. Namely for $\beta \geq 2\alpha$, $\alpha \to 0$ and $\beta \to 0$, our system is unconditionally stable. The obtained approximate numerical solutions are showed to maintain good accuracy.

REFERENCES

[1] Cole JD. On a quasi-linear parabolic equation occurring in aerodynamics. Quart Appl Math 1951; 9: 225-36.

[2] Burger JM. A mathematical model illustrating the theory of turbulence. Adv in App Mech I, Academic Press, New York 1948; pp. 171-99.

[3] El-Danaf TS. Numerical solution of the Korteweg – Vries Burgers' equation by quintic spline method. Studia Unv. Babes - Bolyai Mathematica 2002; Vol. XLVII (2), pp. 41-55.

[4] Gandarias ML. Nonclassical potential symmetries of the Burgers' equation. Symmetry Nonlinear Mathl Phys 1997; 1: 130-7.

[5] Harris SL. Sonic shocks governed by the Burgers' equation. EJAM 1996; 6: 75-107.

[6] Krstic M. On global stabilization of Burgers' equation by boundary control. Sys Control Lett 1999; 37: 123-41.

[7] Ly HV, Mease KD, Tite ES. Distributed and boundary control of the viscous Burgers' equation. Numer Funct Anal Optim 1997; 18: 143-88.

[8] Ramadan MA, El-Danaf TS, Abd Alaal FEI. A numerical solution of the Burgers' equation using B-splines. Chaos Solitons Fractals 2005; 26: 1249-58.

[9] Ramadan MA, El-Danaf TS. Numerical treatment for the Burgers' equation. Math Comput Simul 2005; 70: 90-8.

[10] Weijiu L. A asymptotic behavior of solutions of time – delayed Burgers' equation. Discrete and Continuous Dynamical Systems Ser B 2002; 2(1): 47-56.

[11] El-Danaf T, Ramadan AM. On the analytical and numerical solutions of the one – dimensional nonlinear Burgers' equation. Open Appl Math J 2007; I: 1-8.

[12] Daele MV, Berghe V, Meyer HD. A smooth approximation for the solution of a fourth –order boundary value problem based on non-polynomial splines. J Comput Appl Math 1994; 51: 383-94.

[13] Islam SU, Khan MA, Tirmizi IA, Twizell EH. Non- polynomial spline approach to the solution of a system of third- order boundary-value problems. Appl Appl Math 2005; 168: 152-63.

[14] Ramadan MA, Lashien IF, Zahra WK. Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems. Appl Math Comput 2007; 184: 476-84.

[15] Ramadan MA, Lashien IF, Zahra WK. A class of methods based on septic non- polynomial spline function for the solution of sixth order two point boundary value problems. Int J Comput Math, Accepted.

[16] El-Danaf TS, Abd Alaal FEI. The use of non-polynomial splines for solving a fourth order parabolic partial differential equation. Proceeding of the Mathematical and Physical Society of Egypt, accepted.

[17] Rahidinia J, Jalilian R, Kazemi V. Spline method for the solution of hyperbolic equations. Appl Math Comput., doi:10.1016lj.amc. 2007.082.

Received: June 12, 2007 Revised: August 30, 2007 Accepted: September 04, 2007