Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 3, No. 3, 2015, pp. 218-230

Non-polynomial Spline Method for Solving Coupled Burgers' Equations

Khalid K. Ali*
Department, Faculty of Science,
Al-Azhar University, Nasr City (11884), Cairo, Egypt.
E-mail: khalidkaram2012@yahoo.com.

K. R. Raslan
Department, Faculty of Science,
Al-Azhar University, Nasr City (11884), Cairo, Egypt.
E-mail: kamal_raslan@yahoo.com

Talaat S. EL-Danaf
Mathematics Department, Faculty of Science,
Menoufia University, Shebein El-Koom, Egypt.
E-mail: talaat11@yahoo.com

Abstract In this paper, non-polynomial spline method for solving Coupled Burgers' Equations are presented. We take a new spline function. The stability analysis using Von-Neumann technique shows the scheme is unconditionally stable. To test accuracy the error norms L_2, L_∞ are computed and give two examples to illustrate the sufficiency of the method for solving such nonlinear partial differential equations. These results show that the technique introduced here is accurate and easy to apply.

Keywords. Non-polynomial spline method; Coupled Burgers' Equations.

2010 Mathematics Subject Classification.

1. Introduction

The purpose of this paper is to apply the non-polynomial spline method to the coupled Burgers' equation. The coupled Burgers' equation in the form

$$u_t - u_{xx} + k_1 u u_x + k_2 (uv)_x = 0, \tag{1.1}$$

$$v_t - v_{xx} + k_1 v v_x + k_3 (uv)_x = 0. \tag{1.2}$$

Where k_1, k_2 and k_3 are real constants and subscripts x and t denote differentiation, x
distance, and t time is considered. Boundary conditions

$$\begin{aligned} &u(a,t) = f_1(a,t), \;\; u(b,t) = f_2(b,t), \\ &v(a,t) = g_1(a,t), \;\; v(b,t) = g_2(b,t), \;\; 0 \le t \le T \end{aligned} \tag{1.3}$$

Received: 18 April 2016 ; Accepted: 1 August 2016.
* Corresponding author.

and initial conditions

$$\begin{aligned} u(x,0) &= f(x), \\ v(x,0) &= g(x), \quad a \le x \le b. \end{aligned} \tag{1.4}$$

Various methods are used solve the nonlinear coupled Burgers' equations numerically; which is suggested by [1] firstly. A solution based on fourth order accurate compact ADI scheme by Radwan [2], Khater et al [3] studied Burgers' equations by using A Chebyshev spectral collocation method, Ali et al proposed the algorithm for the numerical solution of two-dimension coupled Burgers' equations using A meshfree technique [4], Rashid and MD studied Viscous coupled Burgers' equations by using the Fourier pseudo-spectral method [5], Liu and Hou used the generalized differential transform method for solving the space and time fractional coupled Burgers' equations [6], cubic B-spline collocation method used by Mittal and Arora for solving the nonlinear coupled viscous Burgers' equations [7], Mokhtari et al used generalized differential quadrature method for solving coupled Burgers' equations [8], Sadek and Kucuk [9] studied Burgers' equations by using a robust technique for solving optimal control of Burgers' type equations, a differential quadrature method [10], Kutluay and Ucar used Galerkin quadratic B-spline finite element method for solving coupled Burgers' equations [11], Srivastava et al studied one-dimensional coupled nonlinear Burgers' equations by using a fully implicit finite-difference method [12], a composite numerical scheme based on finite difference [13], Kumar and Pandit used an implicit logarithmic finite difference method for solving coupled Burgers' equations [14], Mittal and Tripathi studied coupled Burgers' equations by using modified cubic B-spline collocation method [15]. There are not many articles about non-polynomial spline method for the solving nonlinear differential equation system. In this paper, we take a new spline function as form: $T_3 = \text{span}\{1,\ \text{x},\ \tanh(\omega\text{x}),\ \text{sech}(\omega\text{x})\}$, where ω is the frequency of the trigonometric part of the spline functions which will be used to raise the accuracy of the method [16]. This spline function gives the same results if we used the spline function based on other trigonometric functions sin and cos but in this paper, we take spline function with different form. Also, we take linearization of the nonlinear term and we used finite difference approximation and applying Crank-Nicolson scheme. The paper is organized as follows. In Section 2, some details about non-polynomial spline method are provided. In Section 3, the stability is documented. In section 4, numerical results for two different problems and some related figures are given in order to show the efficiency as well as the accuracy of the proposed method. Finally, conclusions are followed in Section 5.

2. Derivation of the Numerical Method

In this section, we gave theoretically discussed for the numerical method using new spline function.

We take spline function in this form $T_3 = \text{span}\{1,\ \text{x},\ \tanh(\omega\text{x}),\ \text{sech}(\omega\text{x})\}$. To set up the non-polynomial spline method, we select an integer $N > 0$ and a time step size $k > 0$ with $h = \frac{b-a}{N+1}$, the mesh points (x_j, t_n) are $x_j = a + jh$ and $t_n = nk$, for $n = 0, 1, ...,$ and $j = 0, 1, ..., N+1$. Let U_j^n and V_j^n be an approximation to $u(x_j, t_n)$

and $v(x_j,t_n)$ respectively, obtained by the segment $p_j(x,t_n)$ of the mixed spline function passing through the points (x_j,U_j^n),(x_{j+1},U_{j+1}^n),(x_j,V_j^n) and (x_{j+1},V_{j+1}^n) respectively. Each segment has the form

$$p_j(x,t_n)=a_j(t_n)\tanh\omega(x-x_j)+b_j(t_n)\sec h\omega(x-x_j)+c_j(t_n)(x-x_j)+d_j(t_n), \tag{2.1}$$

for each $j=0,1,...,N$. To obtain expressions for the coefficients of Eq. (2.1) in terms of U_j^n,U_{j+1}^n, S_j^n and S_{j+1}^n which are as follows:

$$U_j^n=p_j(x_j,t_n),U_{j+1}^n=p_j(x_{j+1},t_n),\ S_j^n=p_j''(x_j,t_n) \text{ and } S_{j+1}^n=p_j''(x_{j+1},t_n). \tag{2.2}$$

Using Eqs. (2.1) and (2.2), we get

$$\begin{aligned}
&a_j+d_j=U_j^n,\\
&a_j\tanh\theta+b_j\sec h\theta+c_jh+d_j=U_{j+1}^n,\\
&-b_j\omega^2=S_j^n,\\
&-2a_j\omega^2\sec h^2\theta\tanh\theta-b_j\omega^2\sec h^3\theta+b_j\omega^2\sec h\theta\tanh^2\theta=S_{j+1}^n,
\end{aligned} \tag{2.3}$$

where $a_j\equiv a_j(t_n),b_j\equiv b_j(t_n),c_j\equiv c_j(t_n),d_j\equiv d_j(t_n)$ and $\theta=\omega h$. By solving last four equations in (2.3) we obtain expressions for the coefficients as:

$$\begin{aligned}
&b_j=\tfrac{-h^2}{\theta^2}S_j^n,\\
&rrra_i=\frac{h^2(S_j^n\sec h^3\theta-S_j^n\sec h\theta\tanh^2\theta-S_{j+1}^n)}{2\theta^2\sec h^2\theta\tanh\theta},\\
&d_j=\tfrac{h^2}{\theta^2}S_j^n+U_j^n,\\
&c_j=\frac{U_{j+1}^n-U_j^n}{h}+\frac{h(S_j^n\sec h\theta-S_j^n)}{\theta^2}+\frac{h(S_j^n\sec h^3\theta+S_j^n\sec h\theta\tanh^2\theta+S_{j+1}^n)}{2\theta^2\sec h^2\theta},
\end{aligned} \tag{2.4}$$

using the continuity condition of the first derivative at $x=x_j$, that is $p_j'(x_j,t_n)=p_{j-1}'(x_j,t_n)$, we get the flowing equations:

$$a_j\omega+c_j=a_{j-1}\omega\sec h^2\theta-b_{j-1}\omega\sec h\theta\tanh\theta+c_{j-1}, j=1,...,N. \tag{2.5}$$

Using Eq. (2.4), after slight rearrangements, then Eq. (2.5) becomes

$$U_{j+1}^n-2U_j^n+U_{j-1}^n=\alpha S_{j+1}^n+\beta S_j^n+\gamma S_{j-1}^n,\quad j=1,...,N. \tag{2.6}$$

Now to obtain expressions for the coefficients of Eq. (2.1) in terms of V_j^n,V_{j+1}^n, δ_j^n and δ_{j+1}^n which are as follows:

$$V_j^n=p_j(x_j,t_n),V_{j+1}^n=p_j(x_{j+1},t_n),\delta_j^n=p_j''(x_j,t_n) \text{ and } \delta_{j+1}^n=p_j''(x_{j+1},t_n). \tag{2.7}$$

Using Eqs. (2.1) and (2.7) and applying the same way we can get

$$V_{j+1}^n-2V_j^n+V_{j-1}^n=\alpha\delta_{j+1}^n+\beta\delta_j^n+\gamma\delta_{j-1}^n, j=1,...,N, \tag{2.8}$$

where

$$\begin{aligned}&\alpha=\frac{h^3\omega}{2\theta^2\sec h^2\theta\tanh\theta}-\frac{h^2}{2\theta^2\sec h^2\theta},\beta=\frac{-h^3\omega\sec h^3\theta}{2\theta^2\sec h^2\theta\tanh\theta}\\&+\frac{h^3\omega\sec h\theta\tanh^2\theta}{2\theta^2\sec h^2\theta\tanh\theta}+\frac{h^2}{\theta^2}-\frac{h^2\sec h\theta}{2\theta^2}+\frac{h^2\sec h^3\theta}{2\theta^2}-\\&\frac{h^2\sec h\theta\tanh^2\theta}{2\theta^2\sec h^2\theta}-\frac{h^2\omega}{2\theta^2\tanh\theta}+\frac{h^2}{2\theta^2\sec h^2\theta},\gamma=\frac{h^3\omega(\sec h^3\theta-\sec h\theta\tanh^2\theta)}{2\theta^2\tanh\theta}\\&+\frac{h^3\omega\sec h\theta\tanh\theta}{\theta^2}+\frac{h^2(\sec h\theta-1)}{\theta^2}+\frac{h^2(-\sec h^3\theta+\sec h\theta\tanh^2\theta)}{2\theta^2\sec h^2\theta},\ \text{and } \theta=\omega h.\end{aligned}$$

Remark As $\omega\to 0$, that is $\theta\to 0$, then $(\alpha,\beta,\gamma)\to(\frac{h^2}{6},\frac{4h^2}{6},\frac{h^2}{6})$.

Then we consider Eqs. (2.6) and (2.8) at two-time level n and $n+1$, addition them to obtain the following relations:

$$\begin{aligned}&(U_{j+1}^{n+1}+U_{j+1}^{n})-2(U_j^{n+1}+U_i^n)+(U_{j-1}^{n+1}+U_{j-1}^{n})=\alpha(S_{j+1}^{n+1}+S_{j+1}^{n})\\&+\beta(S_j^{n+1}+S_j^n)+\alpha(S_{j-1}^{n+1}+S_{j-1}^{n}),\\&(V_{j+1}^{n+1}+V_{j+1}^{n})-2(V_j^{n+1}+V_i^n)+(V_{j-1}^{n+1}+V_{j-1}^{n})=\alpha(\delta_{j+1}^{n+1}+\delta_{j+1}^{n})\\&+\beta(\delta_j^{n+1}+\delta_j^n)+\alpha(\delta_{j-1}^{n+1}+\delta_{j-1}^{n}),\ j=1,...,N.\end{aligned}\tag{2.9}$$

On the other hand, we rewrite (1.1) and (1.2) as

$$\begin{aligned}\frac{\partial^2u(x,t)}{\partial x^2}&=\frac{\partial u(x,t)}{\partial t}+k_1u(x,t)\frac{\partial u(x,t)}{\partial x}+k_2\left[u(x,t)v(x,t)\right]_x,\\\frac{\partial^2v(x,t)}{\partial x^2}&=\frac{\partial v(x,t)}{\partial t}+k_1v(x,t)\frac{\partial v(x,t)}{\partial x}+k_3\left[u(x,t)v(x,t)\right]_x,\\\frac{\partial^2u(x,t)}{\partial x^2}&=\frac{\partial u(x,t)}{\partial t}+k_1u(x,t)\frac{\partial u(x,t)}{\partial x}+k_2\left[u(x,t)v(x,t)\right]_x,\\\frac{\partial^2v(x,t)}{\partial x^2}&=\frac{\partial v(x,t)}{\partial t}+k_1v(x,t)\frac{\partial v(x,t)}{\partial x}+k_3\left[u(x,t)v(x,t)\right]_x.\end{aligned}$$

Take the approximation $u(x,t)=U_j^n$ and $v(x,t)=V_j^n$ then $\frac{\partial^2u(x,t)}{\partial x^2}=S_j^n$ and $\frac{\partial^2v(x,t)}{\partial x^2}=\delta_j^n$, and from famous Cranck–Nicolson scheme and forward finite difference approximation for the derivative t [17], we get.

$$\left[\frac{S_j^{n+1}+S_j^n}{2}\right]=\frac{U_j^{n+1}-U_j^n}{k}+k_1\left[\frac{(UU_x)_j^{n+1}+(UU_x)_j^n}{2}\right]+k_2\left[\frac{(UV)_{x_j}^{\ n+1}+(UV)_{x_j}^{\ n}}{2}\right],\tag{2.10}$$

$$\left[\frac{\delta_j^{n+1}+\delta_j^n}{2}\right]=\frac{V_j^{n+1}-V_j^n}{k}+k_1\left[\frac{(VV_x)_j^{n+1}+(VV_x)_j^n}{2}\right]+k_3\left[\frac{(UV)_{x_j}^{\ n+1}+(UV)_{x_j}^{\ n}}{2}\right],\tag{2.11}$$

we take linearization of the nonlinear term as follows

$$\begin{aligned}&(UU_x)_j^{n+1}=U_j^nU_{x_j}^{\ n+1}+U_j^{n+1}U_{x_j}^{\ n}-U_j^nU_{x_j}^{\ n},\\&(VV_x)_j^{n+1}=V_j^nV_{x_j}^{\ n+1}+V_j^{n+1}V_{x_j}^{\ n}-V_j^nV_{x_j}^{\ n}.\end{aligned}\tag{2.12}$$

From (2.10), (2.11) and (2.12), the following difference equations can be extracted:

$$\begin{aligned}
&S_j^{n+1}+S_j^n=\tfrac{2}{k}(U_j^{n+1}-U_j^n)-\tfrac{k_1}{2h}(U_j^{n+1}(U_{j+1}^n-U_{j-1}^n)+U_j^n(U_{j+1}^{n+1}-U_{j-1}^{n+1}))\\
&+\tfrac{k_2}{2h}(U_j^{n+1}(V_{j+1}^n-V_{j-1}^n)+U_j^n(V_{j+1}^{n+1}-V_{j-1}^{n+1})+V_j^{n+1}(U_{j+1}^n-U_{j-1}^n)\\
&+V_j^n(U_{j+1}^{n+1}-U_{j-1}^{n+1})),\qquad j=1,...,N.
\end{aligned}\tag{2.13}$$

$$\begin{aligned}
&\delta_j^{n+1}+\delta_j^n=\tfrac{2}{k}(V_j^{n+1}-V_j^n)-\tfrac{k_1}{2h}(V_j^{n+1}(V_{j+1}^n-V_{j-1}^n)+V_j^n(V_{j+1}^{n+1}-V_{j-1}^{n+1}))\\
&+\tfrac{k_3}{2h}(U_j^{n+1}(V_{j+1}^n-V_{j-1}^n)+U_j^n(V_{j+1}^{n+1}-V_{j-1}^{n+1})+V_j^{n+1}(U_{j+1}^n-U_{j-1}^n)\\
&+V_j^n(U_{j+1}^{n+1}-U_{j-1}^{n+1})),\qquad j=1,...,N.
\end{aligned}\tag{2.14}$$

Substituting (2.13) and (2.14) in (2.9) and doing some calculations, we get

$$\begin{aligned}
&A_1U_{j+1}^{n+1}+A_2U_j^{n+1}+A_3U_{j-1}^{n+1}+A_4V_{j+1}^{n+1}+A_5V_{j-1}^{n+1}\\
&=A_6U_{j+1}^n+A_7U_j^n+A_8U_{j-1}^n+A_9V_{j+1}^n+A_{10}V_{j-1}^n,
\end{aligned}\tag{2.15}$$

$$\begin{aligned}
&B_1V_{j+1}^{n+1}+B_2V_j^{n+1}+B_3V_{j-1}^{n+1}+B_4U_{j+1}^{n+1}+B_5U_{j-1}^{n+1}f\\
&=B_6V_{j+1}^n+B_7V_j^n+B_8V_{j-1}^n+B_9U_{j+1}^n+B_{10}U_{j-1}^n.
\end{aligned}\tag{2.16}$$

Where

$$\begin{aligned}
&A_1=1-\tfrac{2\alpha}{k}-\tfrac{\alpha k_1}{2h}(U_{j+1}^n+U_{j-1}^n)-\tfrac{\alpha k_2}{2h}(V_{j+1}^n+V_{j-1}^n)-\tfrac{\beta k_1}{2h}U_j^n-\tfrac{\beta k_2}{2h}V_j^n,\\
&A_2=-2-\tfrac{2\beta}{k},\\
&A_3=1-\tfrac{2\alpha}{k}+\tfrac{\alpha k_1}{2h}(U_{j+1}^n+U_{j-1}^n)+\tfrac{\alpha k_2}{2h}(V_{j+1}^n+V_{j-1}^n)+\tfrac{\beta k_1}{2h}U_j^n+\tfrac{\beta k_2}{2h}V_j^n,\\
&A_4=-\tfrac{\alpha k_2}{2h}(U_{j+1}^n+U_{j-1}^n)-\tfrac{\beta k_2}{2h}U_j^n,\\
&A_5=\tfrac{\alpha k_2}{2h}(U_{j+1}^n+U_{j-1}^n)+\tfrac{\beta k_2}{2h}U_j^n,\\
&A_6=-1-\tfrac{2\alpha}{k}+\tfrac{\alpha k_1}{2h}(U_{j+1}^{n+1}+U_{j-1}^{n+1})+\tfrac{\alpha k_2}{2h}(V_{j+1}^{n+1}+V_{j-1}^{n+1})+\tfrac{\beta k_1}{2h}U_j^{n+1}+\tfrac{\beta k_2}{2h}V_j^{n+1},\\
&A_7=2-\tfrac{2\beta}{k},\\
&A_8=-1-\tfrac{2\alpha}{k}-\tfrac{\alpha k_1}{2h}(U_{j+1}^{n+1}+U_{j-1}^{n+1})-\tfrac{\alpha k_2}{2h}(V_{j+1}^{n+1}+V_{j-1}^{n+1})-\tfrac{\beta k_1}{2h}U_j^{n+1}-\tfrac{\beta k_2}{2h}V_j^{n+1},\\
&A_9=\tfrac{\alpha k_2}{2h}(U_{j+1}^{n+1}+U_{j-1}^{n+1})+\tfrac{\beta k_2}{2h}U_j^{n+1},\\
&A_{10}=-\tfrac{\alpha k_2}{2h}(U_{j+1}^{n+1}+U_{j-1}^{n+1})-\tfrac{\beta k_2}{2h}U_j^{n+1},\\
&B_1=1-\tfrac{2\alpha}{k}-\tfrac{\alpha k_1}{2h}(V_{j+1}^n+V_{j-1}^n)-\tfrac{\alpha k_2}{2h}(U_{j+1}^n+U_{j-1}^n)-\tfrac{\beta k_1}{2h}V_j^n-\tfrac{\beta k_2}{2h}U_j^n,\\
&B_2=-2-\tfrac{2\beta}{k},\\
&B_3=1-\tfrac{2\alpha}{k}+\tfrac{\alpha k_1}{2h}(V_{j+1}^n+V_{j-1}^n)+\tfrac{\alpha k_2}{2h}(U_{j+1}^n+U_{j-1}^n)+\tfrac{\beta k_1}{2h}V_j^n+\tfrac{\beta k_2}{2h}U_j^n,\\
&B_4=-\tfrac{\alpha k_2}{2h}(V_{j+1}^n+V_{j-1}^n)-\tfrac{\beta k_2}{2h}V_j^n,\\
&B_5=\tfrac{\alpha k_2}{2h}(V_{j+1}^n+V_{j-1}^n)+\tfrac{\beta k_2}{2h}V_j^n,\\
&B_6=-1-\tfrac{2\alpha}{k}+\tfrac{\alpha k_1}{2h}(V_{j+1}^{n+1}+V_{j-1}^{n+1})+\tfrac{\alpha k_2}{2h}(U_{j+1}^{n+1}+U_{j-1}^{n+1})+\tfrac{\beta k_1}{2h}V_j^{n+1}+\tfrac{\beta k_2}{2h}U_j^{n+1},\\
&B_7=2-\tfrac{2\beta}{k},\\
&B_8=-1-\tfrac{2\alpha}{k}-\tfrac{\alpha k_1}{2h}(V_{j+1}^{n+1}+V_{j-1}^{n+1})-\tfrac{\alpha k_2}{2h}(U_{j+1}^{n+1}+U_{j-1}^{n+1})-\tfrac{\beta k_1}{2h}V_j^{n+1}-\tfrac{\beta k_2}{2h}U_j^{n+1},\\
&B_9=\tfrac{\alpha k_2}{2h}(V_{j+1}^{n+1}+V_{j-1}^{n+1})+\tfrac{\beta k_2}{2h}V_j^{n+1},\\
&B_{10}=-\tfrac{\alpha k_2}{2h}(V_{j+1}^{n+1}+V_{j-1}^{n+1})-\tfrac{\beta k_2}{2h}V_j^{n+1}.
\end{aligned}$$

Now to solving this system, we using initial conditions Eq. (1.4) to find the value of U_j^0 and V_j^0, where $U_j^0 = f(x_j)$ and $V_j^0 = g(x_j)$, for each $j = 0, 1, ..., N+1$. If the procedure is reapplied all the approximation U_j^1 and V_j^1 are known, the values of $U_j^2, U_j^3, U_j^4, ...$and $V_j^2, V_j^3, V_j^4, ...$ can be obtained in a similar manner .

3. Stability analysis of the method

In this section, the standard Von-Neumann concept is applied to investigate the stability analysis of the schemes. At first, we must linearize the nonlinear term of the coupled Burgers' equation by making (U) and (V) as local constants λ_1, λ_2 respectively. According to the Von- Neumann concept, we get

$$\begin{aligned} U_j^n &= A\zeta^n \exp(ij\phi),\\ V_j^n &= B\zeta^n \exp(ij\phi), \end{aligned} \tag{3.1}$$

$$g = \frac{\zeta^{n+1}}{\zeta^n},$$

where A and B are the harmonics amplitude, $\phi = kh$, k is the mode number, $i = \sqrt{-1}$ and g is the amplification factor of the schemes. Substituting (3.1) into the difference (2.15), we get

$$\begin{aligned} &\zeta^{n+1}\left[A\left[\left(2-\tfrac{4\alpha}{k}\right)\cos\phi-\left(2+\tfrac{2\beta}{k}\right)-\left[\tfrac{\alpha k_1}{h}\lambda_1+\tfrac{\alpha k_2}{h}\lambda_2+\tfrac{\beta k_1}{2h}\lambda_1+\tfrac{\beta k_2}{2h}\lambda_2\right]2i\sin\phi\right]\right]\\ &+\zeta^{n+1}\left[-B\left[\tfrac{\alpha k_2}{h}\lambda_1+\tfrac{\beta k_2}{2h}\lambda_1\right]2i\sin\phi\right]=\\ &\zeta^{n}\left[A\left[-\left(2+\tfrac{4\alpha}{k}\right)\cos\phi-\left(-2+\tfrac{2\beta}{k}\right)+\left[\tfrac{\alpha k_1}{h}\lambda_1+\tfrac{\alpha k_2}{h}\lambda_2+\tfrac{\beta k_1}{2h}\lambda_1+\tfrac{\beta k_2}{2h}\lambda_2\right]2i\sin\phi\right]\right]\\ &+\zeta^{n}\left[B\left[\tfrac{\alpha k_2}{h}\lambda_1+\tfrac{\beta k_2}{2h}\lambda_1\right]2i\sin\phi\right], \end{aligned}$$

we get

$$g = \frac{X_2 + iY}{X_1 - iY}, \tag{3.2}$$

where $X_1 = A\left[\left(2-\frac{4\alpha}{k}\right)\cos\phi-\left(2+\frac{2\beta}{k}\right)\right]$, $X_2 = A\left[\left(-2-\frac{4\alpha}{k}\right)\cos\phi-\left(-2+\frac{2\beta}{k}\right)\right]$and

$$Y = \left[B\left[\frac{\alpha k_2}{h}\lambda_1+\frac{\beta k_2}{2h}\lambda_1\right]+A\left[\frac{\alpha k_1}{h}\lambda_1+\frac{\alpha k_2}{h}\lambda_2+\frac{\beta k_1}{2h}\lambda_1+\frac{\beta k_2}{2h}\lambda_2\right]\right]2i\sin\phi.$$

From (3.2) we get $|g| \le 1$, hence the scheme is unconditionally stable. It means that there is no restriction on the grid size, i.e. on h and Δt, but we should choose them in such a way that the accuracy of the scheme is not degraded.

Similar results can be obtained from the difference (2.16), due to symmetric $u$and$v$.

4. Numerical Tests and Results of coupled Burgers' equation

In this section, we present two numerical examples to test the validity of our scheme for solving coupled Burgers' equations.

The norms L_2-norm and L_∞-norm are used to compare the numerical solution with the analytical solution [18].

$$\begin{aligned} \left\|u^E - u^N\right\|_2 &= \sqrt{h\sum_{i=0}^{N}(u_j^E - u_j^N)^2}, \\ \left\|u^E - u^N\right\|_\infty &= \max_j \left|u_j^E - u_j^N\right|, j = 0, 1, \cdots, N, \end{aligned} \tag{4.1}$$

where u^E is the exact solution u and u^N is the approximation solution U_N.

Now we consider two test problems.

Test problem (1.1):

Consider the coupled Burgers' equations (1.1) and (1.2) with the following initial and boundary conditions:

$$u(x,0) = v(x,0) = \sin(x), \quad -\pi \le x \le \pi$$

and

$$u(-\pi,t) = u(\pi,t) = 0, \quad 0 \le t \le T,$$
$$v(-\pi,t) = v(\pi,t) = 0, \quad 0 \le t \le T.$$

The exact solution is

$$u(x,t) = v(x,t) = e^{-t}\sin(x), \ -\pi \le x \le \pi, \ 0 \le t \le T.$$

We compute the numerical solutions using the selected values $k_1 = -2$, $k_2 = 1$ and $k_3 = 1$ with different values of the time step length Δt. In our first computation, we compute L_2- norm and L_∞- norm at $t = 0.1$, $k = 0.001$ while the number of partition N changes. The corresponding results are presented in table 1. In our second computation, we compute L_2- norm and L_∞- norm at time level $t = 1$ for the same parameters in first computation with different decreasing values of Δt. The corresponding results are reported in table 2. In both computations, the results are same for $u(x,t)$ and $v(x,t)$ because of symmetric initial and boundary conditions. And also we make a comparison of our numerical results of the problem (1.1) with the results obtained from [15] and [5] for $N = 50$, $k = 0.01$, $k_1 = -2$, $k_2 = k_3 = 1$ with different time$t$. The corresponding results are presented in table 3.

Table 1: L_2- norm and L_∞- norm for $\omega \to 0$, $t = 0.1$, $k = 0.001$ at different N

N	$u(x,t)$		$v(x,t)$		[7]
	L_2- norm	L_∞- norm	L_2- norm	L_∞- norm	L_∞- norm
50	6.28469 E-4	8.93728 E-4	6.28469 E-4	8.93728 E-4	-
100	1.33123 E-4	1.79992 E-4	1.33123 E-4	1.79992 E-4	-
128	8.61281E-5	9.40987E-5	8.61281E-5	9.40987E-5	1.8178E-5
200	6.90219 E-5	4.56693 E-5	6.90219 E-5	4.56693 E-5	-

Table 2: L_2- norm and L_∞- norm for $\omega \to 0$, $t = 1$, $k = 0.01, 0.001$ at different $N = 200$

k	$u(x,t)$		$v(x,t)$	
	L_2- norm	L_∞- norm	L_2- norm	L_∞- norm
0.01	4.06918E-3	5.01558E-3	4.06918E-3	5.01558E-3
0.001	2.70771E-4	3.47104E-4	2.70771E-4	3.47104E-4

Table 3: Comparison of numerical results of the problem (1.1) with the results obtained from [15] and [5] for the variable u and v with, $N = 50$, $k = 0.01$. $\omega \to 0$,

t	$u(x,t)$		$v(x,t)$		[15]	[5]
	L_2- norm	L_∞- norm	L_2- norm	L_∞- norm	L_∞- norm	L_∞- norm
0.5	1.7887E-3	1.0301E-3	1.7887E-3	1.0301E-3	1.1031 E-4	-
1	1.9529E-3	1.1556E-3	1.9529E-3	1.1556E-3	1.3369 E-4	1.8471E-3

In table 3 we show that our results are related to the results in [15] and [5]. The corresponding graphical illustrations are presented in figures 1 computed solutions of $u(x,t)$ and $v(x,t)$ for $k_1 = -2,\ k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.5, 1$. In figure 2 computed solutions of $u(x,t)$ and $v(x,t)$ for $k_1 = -2, k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.05, 0.1$. In figure 3 computed solutions (exact and approximation) of $u(x,t)$and $v(x,t)$ for $k_1 = -2,\ k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0.1$. In figures 4-6, computed solutions of $u(x,t)$ and $v(x,t)$ at $t = 0.1$, $\Delta t = k = 0.001$ and $N = 200$ for $k_1,\ k_2,\ ,k_1, k3$ and $k_2, k3$ fixed respectively.

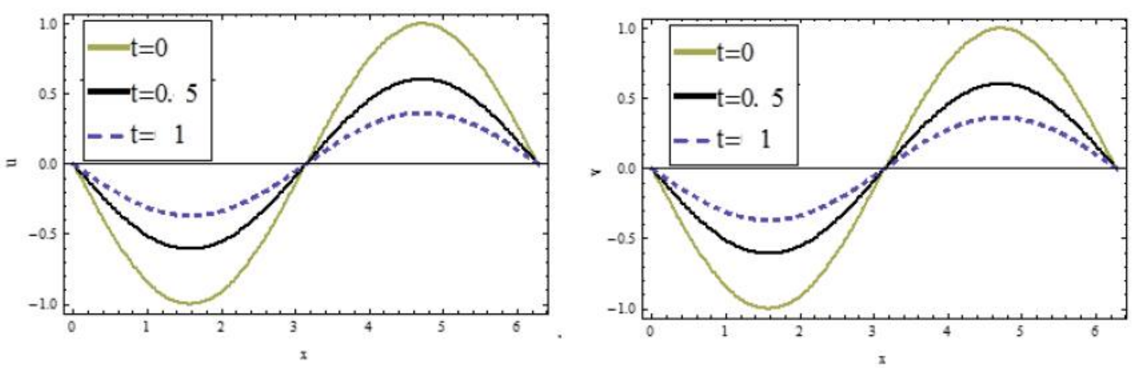

Figure 1: Computed approximation solutions of u and v for $k_1 = -2,\ k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.5, 1$.

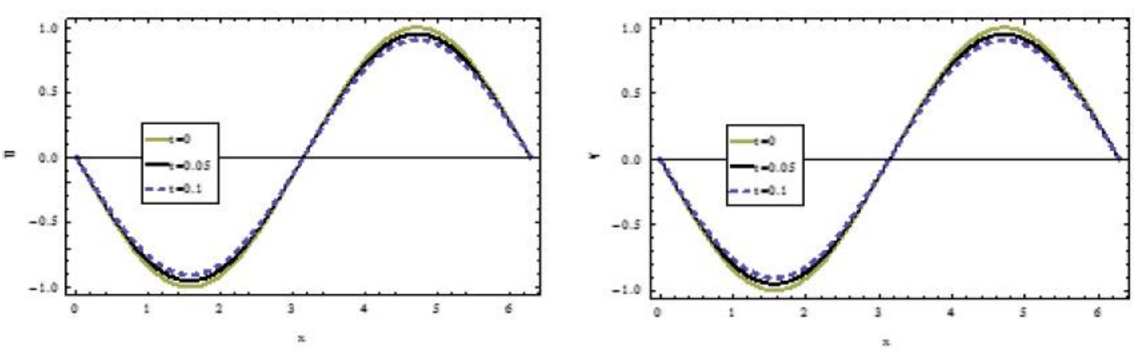

Figure 2: Computed approximation solutions of u and v for $k_1 = -2,\ k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.05, 0.1$.

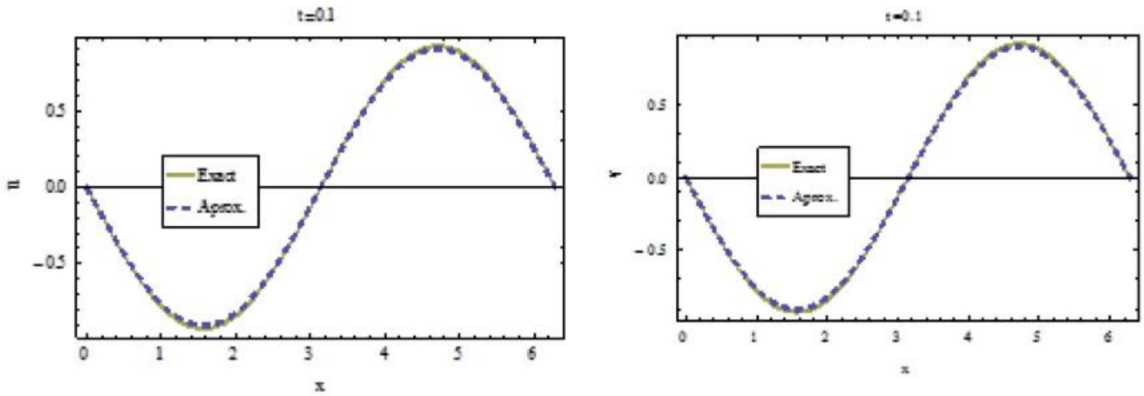

Figure 3: Computed solutions (exact and approximation) of u and v for $k_1 = -2,\ k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0.1$.

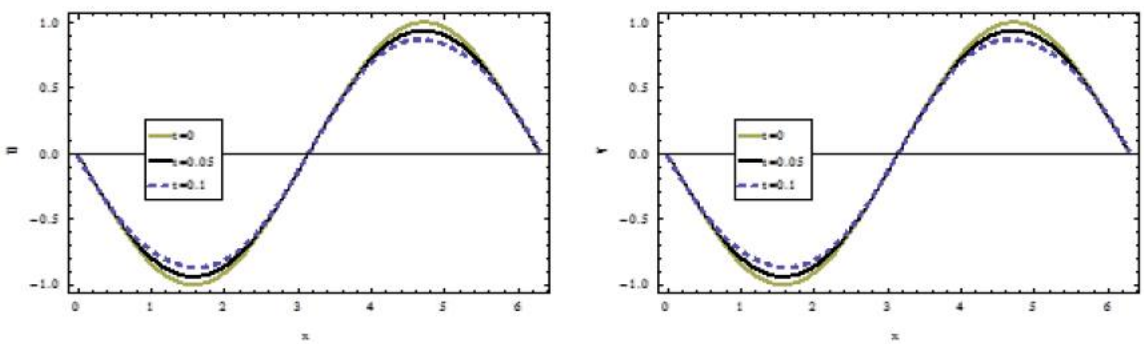

Figure 4: Computed approximation solutions of u and v for $k_1 = -2,\ k_2 = 1,\ k_3 = 8,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.05, 0.1$.

Figure 5: Computed approximation solutions of u and v for $k_1 = -2,\ k_2 = 8,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.05, 0.1$.

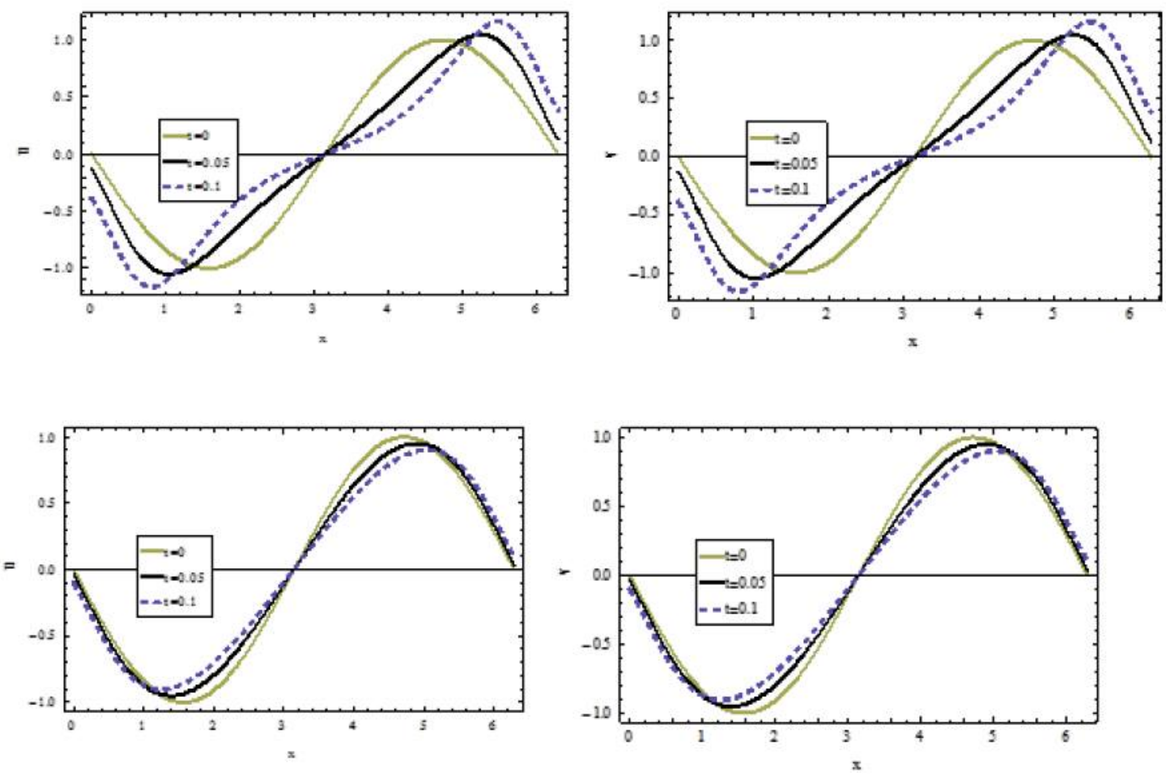

Figure 6: Computed approximation solutions of u and v for $k_1 = 2,\ k_2 = 1,\ k_3 = 1,\ N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.05, 0.1$. **Testproblem(1.2):** Numerical solutions of considered coupled Burgers' equations are obtained for $k_1 = 2$ with different values of k_2 and k_3 at different time levels. In this situation the exact solution is

$$u(x,t) = a_0 - 2A\left[\frac{2k_2-1}{4k_2k_3-1}\right]\tanh(A(x-2At)),$$
$$v(x,t) = a_0\left[\frac{2k_3-1}{2k_2-1}\right] - 2A\left[\frac{2k_2-1}{4k_2k_3-1}\right]\tanh(A(x-2At)).$$

Thus, the initial and boundary conditions are taken from the exact solution is

$$u(x,0) = a_0 - 2A\left[\frac{2k_2-1}{4k_2k_3-1}\right]\tanh\left(A(x)\right),$$
$$v(x,0) = a_0\left[\frac{2k_3-1}{2k_2-1}\right] - 2A\left[\frac{2k_2-1}{4k_2k_3-1}\right]\tanh\left(A(x)\right).$$

Thus, the initial and boundary conditions are extracted from the exact solution. Where $a_0 = 0.05$ and $A = \frac{1}{2}\left[\frac{a_0(4k_2k_3-1)}{2k_2-1}\right]$. The numerical solutions for $u(x,t)$ and $v(x,t)$ have been computed for the domain $x \in [-10, 10]$, $k = 0.01$ and a number of partitions $N = 10$, $N = 100$ and $N = 200$. L_2- norm and L_∞- norm have been computed in table 4 for $t = 1,\ k_1 = 2,\ k_2 = 0.1$ and $k_3 = 0.3$. In tables 5 and 6 we make comparison of our numerical results of the problem (1.2) with the results obtained from [3] and [5] for the variables $u(x,t)$ and $v(x,t)$ with $a_0 = 0.05$, $N = 16$, $k = 0.01$ at different time t and different values of k_2, k_3. In tables 7 and 8, we make comparison of our numerical results of the problem (1.2) with the results obtained from [10] and [7] for the variables $u(x,t)$ and $v(x,t)$ with $a_0 = 0.05$, $N = 21$, $k = 0.01$ at different time t and different values of k_2, k_3. **Table 4:** L_2- norm and L_∞-norm for $t = 1$, $k = 0.01$ at different values of N

$k_1 = 2,\ k_2 = 0.1$ and $k_3 = 0.3,\ \omega \to 0$.

N	$u(x,t)$		$v(x,t)$	
	L_2- norm	L_∞- norm	L_2- norm	L_∞- norm
10	3.34934 E-4	8.70937 E-5	1.3654 E-4	4.93669 E-5
100	4.32057 E-4	1.19748 E -4	3.92223E-4	1.112661E-4
200	4.30722 E-4	1.14653 E-4	3.90891E-4	1.12554 E-4

Table 5. Comparison of numerical results of the problem (1.2) with the results obtained from [3] and [5] for the variable u with $\omega \to 0, /, a_0 = 0.05$, $N = 16$, $k = 0.01$.

t	k_2	$k3$	$u(x,t)$		[3]	[5]
			L_2- norm	L_∞- norm	L_∞- norm	L_∞- norm
0.5	0.1	0.3	1.64456 E-4	4.32274 E-5	1.44 E-3	9.619E-4
	0.3	0.3	2.23953E-4	5.97431 E-5	-	-
1.0	0.1	0.3	3.29426 E-4	8.61447 E-5	1.27E-3	1.153E-3
	0.3	0.3	4.49034 E-4	1.19369 E-4	-	-

Table 6: Comparison of numerical results of the problem (1.2) with the results obtained from [3] and [5] for the variable v with $\omega \to 0$, $a_0 = 0.05$, $N = 16$, $k = 0.01$

t	k_2	$k3$	$v(x,t)$		[3]	[5]
			L_2- norm	L_∞- norm	L_∞- norm	L_∞- norm
0.5	0.1	0.30	6.60178 E-5	2.42852 E-5	5.42E-4	3.332E-4
	0.3	0.30	2.23953 E-4	5.97431 E-5	-	-
1.0	0.1	0.30	1.32006 E-4	4.78615 E-5	1.29E-3	1.162E-3
	0.3	0.30	4.49034 E-4	1.19369 E-4	-	-

In tables 5 and 6 we show that our results are related to the results in [3] and [5].

Table 7. Comparison of numerical results of the problem (1.2) with the results obtained from [10] and [7] for the variable u with $\omega \to 0$, $a_0 = 0.05$, $N = 21$, $k_1 = 2$, $k = 0.01$

t	k_2	$k3$	$u(x,t)$		[10]	[7]
			L_2- norm	L_∞- norm	L_∞- norm	L_∞- norm
0.5	0.1	0.30	1.63355 E-4	4.30493 E-5	4.173 E-5	4.167E-5
	0.3	0.30	2.22563 E-4	5.95221 E-5	-	-
1.0	0.1	0.30	8.27221 E-4	8.58752 E-5	8.275E-5	8.258E-5
	0.3	0.30	4.46234 E-4	1.19032 E-4	-	-

8: Comparison of numerical results of the problem (1.2) with the results obtained from [10] and [7] for the variable v with $a_0 = 0.05$, $N = 21$, $k = 0.01$, $\omega \to 0$.

t	k_2	$k3$	$v(x,t)$		[10]	[7]
			L_2- norm	L_∞- norm	L_∞- norm	L_∞- norm
0.5	0.1	0.30	6.51922E-5	2.39969 E-5	5.418E-5	1.480E-4
	0.3	0.30	2.22563 E-4	5.95221 E-5	-	-
1.0	0.1	0.30	1.30437 E-4	4.74225 E-5	1.074E-4	4.770E-4
	0.3	0.30	4.46234 E-4	1.19032 E-4	-	-

In tables 7 and 8, we show that our results are related to the results in [10] and [7].

Now we take the test problem (1.2) at the domain $x \in [0,1]$, $k = 0.001$ and $k_1 = 2, k_2 = 1$, $k_3 = 0.3$. L_2-norm and L_∞- norm have been computed see Table 10 for $t = 1$ with different values of N.

Table 9: L_2- norm and L_∞- norm for $t = 1$, $k = 0.001$ at different values of N $k_1 = 2$, $k_2 = 1$ and $k_3 = 0.3$, $\omega \to 0$.

N	$u(x,t)$		$v(x,t)$	
	L_2- norm	L_∞- norm	L_2- norm	L_∞- norm
10	3.13532E-5	2.99042 E-5	1.08304E-5	1.0324E-5
100	3.00432E-5	2.99041E-5	1.07781E-5	1.03202E-5
200	3.00234E-6	2.98034E-5	1.06245E-5	1.03101E-5

The corresponding graphical illustrations are presented in figure 7 computed approximation solutions of $u(x,t)$ and $v(x,t)$ for $k_1 = 2$, $k_2 = 1$, $k_3 = 0.3$, $N = 200$ and $\Delta t = k = 0.001$ at $t = 0, 0.5, 1, x \in [0,1]$.

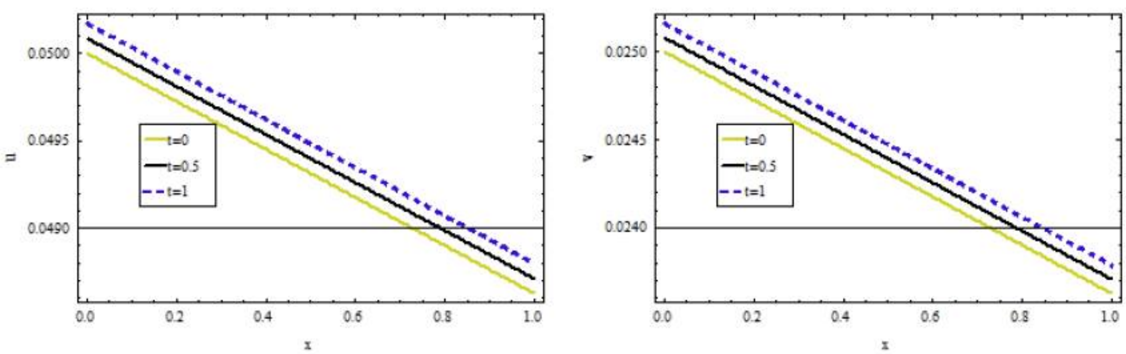

Figure 7: Computed approximation solutions of u and v for $k_1 = 2$, $k_2 = 0.1$, $k_3 = 0.3$, $N = 200$ and $\Delta t = k = 0.01$ at $t = 0, 0.5, 1$.

5. CONCLUSIONS:

In this paper, we applied non-polynomial spline method to develop a numerical method for solving coupled Burgers' equations and shown that the schemes are unconditionally stable. We take a new spline function and we make linearization for the nonlinear term. We tested our schemes through two test problems. Accuracy was shown by calculating error norms L_2 andL_∞.

References

[1] S. E. Esipov, Coupled Burgers Equations- A Model of Poly dispersive sedimentation, James Franck Institute and Department of Physics, University of Chicago, 1995.

[2] S. F. Radwan, On the Fourth-Order Accurate Compact ADI Scheme for Solving the Unsteady Nonlinear Coupled Burgers' Equations, Journal of Nonlinear Mathematical Physics, 6(1) (1999), 13–34.

[3] A. H. Khater, R. S. Temsah and M. M. Hassan, A Chebyshev spectral Collocation method for solving Burgers'-type equations, Journal of Computational and Applied Mathematics, 222 (2008), 333–350.

[4] A. Ali, A. Islam and S. Haq, A Computational Meshfree Technique for the Numerical Solution of the Two-Dimensional Coupled Burgers' Equations, International Journal for Computational Methods in Engineering Science and Mechanics, 10 (2009), 406–422.

[5] A. Rashid and A. I. B. MD. Ismail, A Fourier Pseudospectral Method for Solving Coupled Viscous Burgers Equations, Computational Method in Applied Mathematics, 9(4) (2009), 412–420.

[6] J. Liu and G. Hou, Numerical solutions of the space and time fractional Coupled Burgers equations by generalized differential transform method, Applied Mathematics and Computation, 217 (2011), 7001–7008.

[7] R. C. Mittal and G. Arora, Numerical solution of the coupled viscous Burgers' equation, Commun Nonlinear Sci. Numer. Simulat, 16 (2011), 1304–1313.

[8] R. Mokhtari, A. S. Toodar and N. G. Chengini, Application of the Generalized Differential Quadrature Method in Solving Burgers' Equations, Commun. Theor. Phys., 56 (2011), 1009–1015.

[9] I. Sadek and I. Kucuk, A robust technique for solving optimal control of Coupled Burgers' equations, IMA Journal of Mathematical Control and Information, 28 (2011), 239–250.

[10]] R.C. Mittal and Ram Jiwar, A differential quadrature method for numerical solutions of Burgers'-type equations, International Journal of Numerical Methods for Heat & Fluid, 22(7) (2012), 880–895.

[11] S. Kutluay and Y. Ucar, Numerical solutions of the coupled Burgers' equation by the Galerkin quadratic B-spline finite element method, Math. Meth. Appl. Sci., 36 (2013), 2403–2415.

[12] V.K. Srivastava, M. K. Awasthi and M. Tamsir, A fully implicit Finite difference solution to one dimensional Coupled Nonlinear Burgers' equations, International Journal of Mathematical, Computational, Physical and Quantum Engineering, 7(4) (2013), 417-422.

[13] M. Kumar and S. Pandit, A composite numerical scheme for the numerical simulation of coupled Burgers' equation, Computer Physics Communications, 185 (2014), 809–817.

[14] V. K. Srivastava, M. Tamsir, M. K. Awasthi and S. Sing, One dimensional coupled Burgers' equation and its numerical solution by an implicit logarithmic Finite difference method, Aip Advances, 4 (2014), 037119.

[15] R. C. Mittal and A. Tripathi, A Collocation Method for Numerical Solutions of Coupled Burgers' Equations, International Journal for Computational Methods in Engineering Science and Mechanics, 15 (2014), 457–471.

[16] N.G. Chegini, A. Salaripanah, R. Mokhtari and D. Isvand, "Numerical solution of the regularized long wave equation using non-polynomial splines", Nonlinear Dyn., 69 (2012), 459–471.

[17] T. S. EL-Danaf, K. R. Raslan and Khalid K. Ali," collocation method with cubic B- Splines for solving the GRLW equation", Int. J. of Num. Meth. and Appl., 15(1) (2016), 39–59.

[18] T. S. EL-Danaf, K. R. Raslan and Khalid K. Ali,"New Numerical treatment for the Generalized Regularized Long Wave Equation based on finite difference scheme", Int. J. of S. Comp. and Eng. (IJSCE)', 4 (2014), 16–24.