Numerical solution of Burgers equation with nonlinear damping using non-polynomial tension spline

H.S.Shekarabi∗ and J.Rashidinia†‡

Abstract
Numerical solution of Burgers equation with nonlinear damping term has been investigated. We developed new approach based on non-polynomial cubic tension spline approximation. The proposed approach depends on the parameters involving in tension spline. By choosing suitable values of such parameters the optimal local truncation error of the scheme can be obtained. Convergence analysis of presented method has been discussed in details and we have shown under appropriate condition the method convergence. The method tested on two problems, numerical results have been compared with the exact solution to justify the usefulness and accurate nature of proposed method.

Keywords: Burgers equation, Non-polynomial tension spline, Convergence analysis.

AMS subject Classification 2010: 65M06, 12, 99.

1. Introduction

The Burgers equation introduced by Burgers [1] provide fundamental pedagogical examples for many important equation in nonlinear Partial Differential equations such as traveling waves, shock formation, similarity solutions and singular perturbations [14, 27, 40, 43], it appears in some of condensed matter and statistical Physics and non-physics problems such as vehicular traffic [7], The Kardar-Parisi-Zhang or KPZ equation [23, 2], traffic flow, shallow water waves, gas dynamics, and fluids with the dissipative viscous behavior [28, 29, 30]. Furthermore, Burgers equation is studying in directed polymers [24, 3] and has found interesting applications in cosmology, such as ”Zel’dovich approximation” [46] and ”adhesion model” [16]. Another application of Burgers equation is in the theory of turbulence and field [34, 37, 15, 31].

Hofe[20] and Cole[8] have shown the Burgers equation can reduce to heat conduction equation.

In this paper, we investigate the solution of generalized one-dimensional Burgers equation with nonlinear damping term [38] of the form

$$u_t + auu_x - u_{xx} = g(u), \quad x \in \Omega = [c, d], \quad t \geq 0,$$

∗Department of Mathematics, College of basic science, Karaj Branch, Islamic Azad University, Alborz, Iran Email: hoda.shekarabi@kiau.ac.ir
†School of Mathematics, Iran University of science and Technology Email: rashidinia@iust.ac.ir
‡Corresponding Author.
with the initial condition
\[(1.2) \quad u(x, t_0) = \phi(x),\]
and the boundary conditions
\[(1.3) \quad u(c, t) = p_0(t), \quad u(d, t) = p_1(t),\]
where \(u(x, t)\) indicates the velocity for the space \(x\) and time \(t\), \(a\) is parameter and \(g(u)\) is damping term.

The Burgers equation in the first term is an unsteady term, the second and third term represents nonlinear convection and diffusion problem, with nonlinearity term, can be survived by many researcher.

Different numerical technique have been used for solving Burgers equation. Finite difference methods have been given by Biringen et al[4] and Kuthay et al.[25] and recently, by Inan et al.[21]. Finite elements methods have been given by Caldwell et al.[9] and Vargh et al.[42] and Ozis et al.[35]. Spectral methods have been developed by Bar-yoseph et al.[5] and Mansell et al.[32]. Pseudo-spectral method has been used by Darvishi et al.[12] and distributed approximation function approach has studied by Zang et al.[47] and Wei et al.[44]. Boundary elements methods is given by Bahadir et al.[6]. A wavelet collocation method has used by Garba[17], furthermore quasi wavelet based numerical method has been suggested by Wan et al.[45]. Fast adaptive diffusion wavelet method have been survived by Goyal et al.[18]. Least square quadratic B-spline finite elements has been given by Kuthay et al.[26]. Various B-spline have been proposed by Dag et al.[13,41]. B-spline and multi-quadratic quasi-interpolation have been described by Zhu et al. and Chen et al.[48,10].

The present work attempts to use cubic non-polynomial spline.[36,19,33]. One of the important ability of this approximation is the tension parameters involving definition of non-polynomial cubic spline which can be chosen in such a way that the local truncation error of the proposed method can be optimal. Hence, it has been demonstrate that tension spline give better result. This paper is organized as follows: In section 2, derivation and formulation of the cubic non-polynomial tension spline along with consistency relation of second derivatives discussed in details. In section 3, the derivation of two level scheme based on non-polynomial tension spline has been described. In section 4, convergence analysis of the present method has been discussed in detail and we have shown under appropriate condition the method converges. At the end, we illustrate the accuracy and efficiency of the proposed method by testing this approach on two test problems. Comparison of the numerical result are given.

2. Non-polynomial tension spline

Following our earlier works, let \(s(x)\) of class \(c^2[c, d]\) be non-polynomial tension spline interpolating the function \(u(x)\) at the grid point \(x_i, \; i = 0, 1, 2, ..., n\). For each
segment \([x_l, x_{l+1}], l = 0, 1, ..., n - 1\), the non-polynomial \(s(x)\) defined by

\[
s(x) = a_l + b_l(x - x_l) + c_l(e^{\omega(x-x_l)} - e^{-\omega(x-x_l)}) + d_l(e^{\omega(x-x_l)} + e^{-\omega(x-x_l)}),
\]

where the \(a_l, b_l, c_l, d_l\) are unknown coefficients and \(\omega\) is arbitrarily parameter. To determine the unknown coefficient in (2.1) we denote the following relations

\[
s(x_l) = u_l, \quad s(x_{l+1}) = u_{l+1}, \quad s'(x_l) = m_l, \quad s'(x_{l+1}) = m_{l+1},
\]

(2.2)

\[
s''(x_l) = M_l, \quad s''(x_{l+1}) = M_{l+1}.
\]

The first and second derivatives of non-polynomial tension spline function \(s(x)\) are

\[
s'(x_l) = m_l, \quad s'(x_{l+1}) = m_{l+1},
\]

(2.3)

\[
s''(x_l) = M_l, \quad s''(x_{l+1}) = M_{l+1}.
\]

Now using (2.2)-(2.4) and after some algebraic manipulation, we can determine the unknown coefficients in (2.1) as

\[
a_l = u_l - \frac{M_l}{\omega^2}, \quad b_l = \frac{u_{l+1} - u_l}{h} + \frac{M_l - M_{l+1}}{\omega h},
\]

(2.5)

\[
c_l = \frac{2M_{l+1} - (e^\theta + e^{-\theta})M_l}{2\omega^2(e^\theta - e^{-\theta})}, \quad d_l = \frac{M_l}{2\omega^2},
\]

where \(h = \frac{d-c}{n}, \theta = \omega h\).

Using the continuity of the first derivative at \((x_l, u_l)\), that is \(s'(x_l^-) = s'(x_l^+)\). We obtain the following equation for \(l = 1, ..., n\).

\[
\frac{u_{l+1} - 2u_l + u_{l-1}}{h^2} = \alpha M_{l+1} + 2\beta M_l + \alpha M_{l-1},
\]

(2.6)

where

\[
\alpha = \frac{1}{\theta^2} (1 - \frac{2\theta}{e^\theta - e^{-\theta}}), \quad \beta = \frac{1}{\theta^2} \left( \frac{\theta(e^\theta + e^{-\theta})}{e^\theta - e^{-\theta}} - 1 \right).
\]

When \(\omega \to 0\), that \(\theta \to 0\), then \((\alpha, \beta) \to \left( \frac{1}{6}, \frac{1}{3} \right)\), and the relations defined by (2.5) reduced into construction relation of conventional cubic spline.

Now by using the continuous of the first derivative, we have

\[
s'(x_l^+) = u_{x_l} = \frac{u_{l+1} - u_l}{h} - h[\alpha M_{l+1} + \beta M_l]
\]

(2.7)

\[
s'(x_l^-) = u_{x_l} = \frac{u_l - u_{l-1}}{h} + h[\beta M_l + \alpha M_{l-1}]
\]

combining (2.6) and (2.7), we obtain

\[
m_l = s'(x_l) = u_{x_l} = \frac{u_{l+1} - u_{l-1}}{2h} - \frac{\alpha h}{2}[M_{l+1} + M_{l-1}]
\]

(2.8)

Similarly, we have

\[
m_{l+1} = s'(x_{l+1}) = u_{x_{l+1}} = \frac{u_{l+1} - u_l}{h} + h[\beta M_{l+1} + \alpha M_l]
\]

(2.9)

\[
m_{l-1} = s'(x_{l-1}) = u_{x_{l-1}} = \frac{u_l - u_{l-1}}{h} - h[\beta M_{l-1} + \alpha M_l]
\]

(2.10)
3. The method based on tension spline

The notation \( u^j_l \) is used for the discrete approximation value of \( u(x_l, t_j) \), \( l = 0, 1, \ldots, n \) and \( j = 0, 1, \ldots, m \), in which \( n \) and \( m \) are integer and \( x_l = c + lh \) and \( t_j = t_0 + jk \), where \( k \) is the step size in \( t \) direction.

We consider the following finite difference approximation

\[
\begin{align*}
(3.1) & \quad \bar{u}_{t_i}^j = \frac{u_{i+1}^j - u_i^j}{k} = u_{t_i}^j + O(k) \\
(3.2) & \quad \bar{u}_{x_{i+1}}^j = \frac{u_{i+1}^j - u_{i-1}^j}{2h} = u_{x_{i+1}}^j + O(h^2) \\
(3.3) & \quad \bar{u}_{t_{i-1}}^j = \frac{u_{i-1}^j - u_i^j}{k} = u_{t_{i-1}}^j + O(k) \\
(3.4) & \quad \bar{u}_{x_i}^j = \frac{u_{i+1}^j - u_{i-1}^j}{2h} = u_{x_i}^j + O(h^2) \\
(3.5) & \quad \bar{u}_{x_{i+1}}^j = \frac{3u_{i+1}^j - 4u_i^j + u_{i-1}^j}{2h} = u_{x_{i+1}}^j + O(h^2) \\
(3.6) & \quad \bar{u}_{x_{i-1}}^j = \frac{-3u_{i-1}^j + 4u_i^j - u_{i+1}^j}{2h} = u_{x_{i-1}}^j + O(h^2),
\end{align*}
\]

By replacing space derivatives by non-polynomial tension spline

\[
\begin{align*}
(3.7) & \quad \bar{u}_{xx_{i}}^j = s''(x_i, t_j) = M_i^j + O(h^2) \\
(3.8) & \quad \bar{u}_{x_i}^j = s'(x_i, t_j) = m_i^j + O(h^3)
\end{align*}
\]

By using the relations of (3.1), (3.7) and (3.8), we can obtain the new approximate solution of equation (1) as

\[
(3.9) \quad \left( \frac{u_{i+1}^j - u_i^j}{k} \right) + au_{t_i}^j.m_i^j - g_i^j = M_i^j,
\]

where \( m_i^j \) similar to (2.8) in \( j\text{th} \) time level and \( g_i^j = g(u_i^j) \), \( l = 1(1)n-1 \). Furthermore, similar to (2.5) in \( j\text{th} \) time level, we get

\[
(3.10) \quad \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} = \alpha M_{i+1}^j + 2\beta M_i^j + \alpha M_{i-1}^j.
\]

We substitute (3.9) in (3.10) and by the help of (3.1)-(3.3), we obtain

\[
\begin{align*}
\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} & = \alpha \left( \frac{u_{i+1}^j - u_i^j}{k} \right) + \alpha au_{t_i}^j.m_i^j - \alpha g_{t_i}^j \\
& + 2\beta \left( \frac{u_{i+1}^j - u_i^j}{k} \right) + 2\beta au_{t_i}^j.m_i^j - 2\beta g_{t_i}^j \\
& + \alpha \left( \frac{u_{i+1}^j - u_{i-1}^j}{k} \right) + \alpha au_{t_{i-1}}^j.m_{i-1}^j - \alpha g_{t_{i-1}}^j.
\end{align*}
\]
Now by using equations of (3.4)-(3.6) in (2.8)-(2.10), we have

\[ m^j_i = \frac{u^j_{i+1} - u^j_{i-1}}{2h} - \frac{a h}{2} \left( \frac{u^j_{i+1} - u^j_{i-1}}{k} \right) - \frac{a h a}{2} u^j_{i+1} \left( \frac{3u^j_{i+1} - 4u^j_i + u^j_{i-1}}{2h} \right) \]

\[ + \frac{a h}{2} \left( \frac{u^j_{i+1} - u^j_{i-1}}{k} \right) + \frac{a h a}{2} u^j_{i-1} \left( \frac{-3u^j_{i-1} + 4u^j_i - u^j_{i-1}}{2h} \right) + \frac{a h g^j_{i+1} - a h g^j_{i-1}}{2} \]

\[ m^j_{i+1} = \frac{u^j_{i+1} - u^j_i}{h} + \beta h \left( \frac{u^j_{i+1} - u^j_i}{k} \right) + \beta h a u^j_i \left( \frac{3u^j_{i+1} - 4u^j_i + u^j_{i-1}}{2h} \right) \]

\[ + \beta h \left( \frac{u^j_{i+1} - u^j_i}{k} \right) + \alpha h a u^j_i \left( \frac{u^j_{i+1} - u^j_i}{2h} \right) - \beta h g^j_{i+1} - \alpha h g^j_i \]

\[ m^j_{i-1} = \frac{u^j_i - u^j_{i-1}}{h} - \beta h \left( \frac{u^j_{i+1} - u^j_i}{k} \right) - \beta h a u^j_i \left( \frac{-3u^j_{i-1} + 4u^j_i - u^j_{i+1}}{2h} \right) \]

\[ - \beta h \left( \frac{u^j_{i+1} - u^j_i}{k} \right) - \beta h a u^j_i \left( \frac{u^j_{i+1} - u^j_i}{2h} \right) + \beta h g^j_{i-1} + \alpha h g^j_i \]

By substituting equations (3.12)-(3.14) in (3.11), we can obtain

\[ (b_0)^j_i u^j_{i+1} + (b_1)^j_i u^j_{i-1} + (b_2)^j_i u^j_{i+1} = b_3(u^j_{i+1} + u^j_{i-1}) + b_4 u^j_i \]

\[ + b_5(g^j_{i+1} + g^j_{i-1}) + b_{10}g^j_i + b_6((u^j_{i+1})^2 - (u^j_{i-1})^2) - b_7 u^j_i (u^j_{i+1} - u^j_{i-1}) \]

\[ - b_8 u^j_i (B^j_i) - b_9(u^j_{i+1}(A^j_i) + u^j_{i-1}(C^j_i)) \]

\[ l = 1(1)n - 1, \]

where

\[ (b_0)^j_i = h^3 \alpha a(u^j_{i+1} - u^j_i) + h^2 \alpha, \]

\[ (b_1)^j_i = h^3 \alpha a(u^j_i - u^j_{i-1}) + h^2 \alpha, \]

\[ (b_2)^j_i = h^3 \alpha a(u^j_{i+1} - u^j_{i-1}) + 2 \beta h^2, \]

\[ b_3 = h^2 \alpha + k, \]

\[ b_4 = 2 \beta h^2 - 2k, \]

\[ b_5 = kh^2 \alpha, \]

\[ b_6 = h^3 \alpha a, \]

\[ b_7 = h^3 \alpha a(\alpha - \beta), \]

\[ b_8 = 2 \beta k h^2 a, \]

\[ b_9 = ak h^2 a, \]

\[ b_{10} = 2 \beta k h^2, \]
and
\[ A_j^i = \frac{u_{j+1}^i - u_j^i}{h} + \beta h a u_{i+1}^j \left( \frac{3u_{j+1}^i - 4u_j^i + u_{j-1}^i}{2h} \right) \\
+ \alpha h a u_j^i \left( \frac{u_{j+1}^i - u_{j-1}^i}{2h} \right) - (h \beta g_{j+1}^i + \alpha h g_j^i) \]
\[ B_j^i = \frac{u_{j+1}^i - u_j^i}{2h} - \frac{\alpha h a u_{i+1}^j}{2} \left( \frac{3u_{j+1}^i - 4u_j^i + u_{j-1}^i}{2h} \right) \\
+ \frac{\alpha h}{2} a u_{j-1}^j \left( -3u_{j-1}^i + 4u_j^i - u_{j+1}^i \right) + (h \beta g_{j+1}^i - \frac{\alpha h}{2} g_j^i), \]
\[ C_j^i = \frac{u_j^i - u_{j-1}^i}{h} - \beta h a u_{i-1}^j \left( \frac{-3u_{j-1}^i + 4u_j^i - u_{j+1}^i}{2h} \right) \\
- \alpha h a u_j^i \left( \frac{u_{j+1}^i - u_{j-1}^i}{2h} \right) + (h \beta g_{j+1}^i + \alpha h g_j^i). \]

The above system can be associated with boundary conditions. By solving this system the approximate solution can be obtain.

3.1. The appropriate parameters. Using Taylor expansion about the grid point \( u(x_i, t_j) \) finally we obtain the local truncation error
\[ T_j^i = k^2 h^2 \left( \alpha + \beta \right) \frac{\partial^2 u}{\partial t^2} + h^4 k \left( \alpha - \frac{1}{12} \right) \frac{\partial^4 u}{\partial x^4} + h^4 k^2 \frac{\alpha}{2} \frac{\partial^4 u}{\partial t^2 \partial x^2} + \ldots \]

The consistency relation for (2.5) lead to the equation \( 2\alpha + 2\beta = 1 \), by simplifying the above equation and choose \( \alpha = \frac{1}{12} \) and \( \beta = \frac{5}{12} \) obtain the scheme of \( O(k^2 + h^4 + h^2 k^2) \).

4. Convergence of the method

Here we analyze the convergence of the system, we can write system (3.15) in the matrix form
\[ (4.1) \quad P U^{j+1} = Q U^j + G(U^j) \]

\( P \) is tri-diagonal matrix with variable entries, \( Q \) is coefficient matrix of \( U^j \) with constant entries and \( G(U^j) \) is nonlinear terms in this system.

to prove convergence, we suppose \( a > 0, \rho = \max|u_l^i|, l = 1(1)n - 1, j = 0(1)m \).

In this paper \( ||.|| \) means \( ||.||_\infty \).

4.1. Lemma. \( P \) is nonsingular.

proof. It is sufficient to solve that \( P \) is strictly diagonally dominant. Therefore we must prove
\[ (4.2) \quad | (b_0)^i_j + (b_1)^i_j | \leq | (b_2)^i_j | \]

We have
\[ (4.3) \quad \geq 2h^2 \alpha - \left| h^3 \alpha \beta a (u_{j+1}^i - u_j^i) + h^3 \alpha \beta a (u_j^i - u_{j-1}^i) + 2h^2 \alpha \right| \]
By using inequality (4.3) in left hand side of equation (4.2), we obtain
\[ 2h^2 \alpha - |h^3 \alpha \beta a(u_{i+1}^j - u_i^j)| \leq |h^3 \alpha^2 a(u_{i+1}^j - u_i^j)| + 2\beta h^2 \]
(4.4)
\[ 2h^2(\alpha - \beta) \leq a(h^3 \alpha \beta + h^3 \alpha^2) | u_{i+1}^j - u_i^j | . \]

We know \( \alpha - \beta = -\frac{1}{3} \), therefore inequality (4.4) is obvious and proof complete.

4.1. Theorem. The discrete numerical scheme defined by (3.15) is convergent, provided that \( \|N\| \leq h^2(2 + h(\frac{a^2}{24})) \).

Proof. We assume that \( U^{j+1} \) and \( \hat{U}^{j+1} \) are exact and approximation solution of (4.1), respectively. The error in the solution is:
\[ U^{j+1} - \hat{U}^{j+1} = P^{-1}(Q^{j} - \hat{Q}^{j}) + P^{-1}[G(U^{j}) - G(\hat{U}^{j})] \]
(4.5)
where \( E = (e_1, e_2, ..., e_n)^T \)

Following [11] we have
\[ G(U^{j}) - G(\hat{U}^{j}) = E^j N \]
(4.6)

\( N \) is the coefficient matrix of the nonlinear term.

Now by using of equation (4.6) in (4.5) we obtain
\[ E^{j+1} = P^{-1}Q E^j + P^{-1}[G(U^{j}) - G(\hat{U}^{j})] \]
(4.7)

Using the infinity norm, we can write
\[ \| E^{j+1} \| \leq \| P^{-1} Q \| \| E^j \| \]
\[ \| E^{j+1} \| \leq \| P^{-1}(Q + N) \| \| E^j \| \]
\[ \| E^{j+1} \| \leq \| P^{-1} \| \| (Q + N) \| \| E^j \| \]
\[ \leq \left( \| P^{-1} \| \| (Q + N) \| \right)^j \| E^0 \| \]
\[ \vdots \]
\[ \| E^{j+1} \| \leq \left( \| P^{-1} \| \| (Q + N) \| \right)^{j+1} \| E^0 \| \]

The method is convergent if
\[ \| P^{-1} \| \| (Q + N) \| \leq 1 \]
\[ \| (Q + N) \| \leq \frac{1}{\| P^{-1} \|} \]
(4.8)
\[ \| N \| - \| Q \| \leq \| (Q + N) \| \leq \frac{1}{\| P^{-1} \|} \]

Since \( \| P \| \| P^{-1} \| \geq 1 \), we have
\[ \| N \| - \| Q \| \leq \frac{1}{\| P^{-1} \|} \leq \| P \| \]
(4.9)
\[ \| N \| \leq \| Q \| + \| P \| \]
By simple calculation, we achieve

\[(4.10) \quad \|Q\| = h^2,\]
\[(4.11) \quad \|P\| \leq h^2 + h^3 a \left(\frac{P}{24}\right)\]

By substitute (4.10) and (4.11) in (4.9), the proof complete.

5. Numerical illustrations

To illustrate accuracy and ability of the proposed method, we considered two examples. Note that, the proposed non-linear tension spline is a two-level scheme therefore the starting level can be determined by the given initial condition. Finally we solve the arising system.

**Example 1.**

We consider equation

\[u_t + auu_x = u_{xx}, \quad t \geq 0, \quad 0 \leq x \leq 1\]

with the following initial and boundary conditions

\[u(x, 0) = \sin(\pi x), \quad 0 \leq x \leq 1\]
\[u(0, t) = 0,\]
\[u(1, t) = 0, \quad t \geq 0.\]

The exact solution of the above equation is taken

\[u(x, t) = \frac{2\pi}{a_0 + \sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2 t) \sin(n\pi x)}\]

\[a_0 = \int_0^1 \exp\{(-\pi)^{-1}[1 - \cos(\pi x)]\} dx\]

\[a_n = 2 \int_0^1 \exp\{(-\pi)^{-1}[1 - \cos(\pi x)]\} \cos(n\pi x) dx \quad n = 1, 2, 3, ...\]

Example 1 is the Burgers equation without damping terms. The proposed scheme (3.15) applied on example 1, with \(a = 0.1\) and \(0.01, k = 0.00001\) and values of step size \(h = 0.02\) and \(h = 0.01\) for \(t_f = 0.1\). The computed solution are compare with exact solution, the maximum absolute errors are tabulated in table 1. In table 2, we take \(h = 0.1\) and \(k = 0.001\), the results are computed for different time levels and different \(a\). The maximum absolute error are tabulated in table 2. In Figures 1–3, we show the graphs between exact and numerical solutions at \(t = 1, t = 3\) and \(t = 5\) in different \(a\).
Table 1. Maximum absolute error for example 1

<table>
<thead>
<tr>
<th>$x$</th>
<th>$a = 0.1$</th>
<th>$a = 0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h = 0.02$</td>
<td>$h = 0.01$</td>
</tr>
<tr>
<td>0.10</td>
<td>8.45119($-4$)</td>
<td>2.07841($-5$)</td>
</tr>
<tr>
<td>0.20</td>
<td>1.44193($-4$)</td>
<td>3.56293($-5$)</td>
</tr>
<tr>
<td>0.30</td>
<td>1.83155($-4$)</td>
<td>4.54565($-5$)</td>
</tr>
<tr>
<td>0.40</td>
<td>2.04631($-4$)</td>
<td>5.09991($-5$)</td>
</tr>
<tr>
<td>0.50</td>
<td>2.10730($-4$)</td>
<td>5.27339($-5$)</td>
</tr>
<tr>
<td>0.60</td>
<td>2.02309($-4$)</td>
<td>5.08367($-5$)</td>
</tr>
<tr>
<td>0.70</td>
<td>1.78959($-4$)</td>
<td>4.51683($-5$)</td>
</tr>
<tr>
<td>0.80</td>
<td>1.39121($-4$)</td>
<td>3.52934($-5$)</td>
</tr>
<tr>
<td>0.90</td>
<td>6.59886($-4$)</td>
<td>2.05162($-5$)</td>
</tr>
</tbody>
</table>

Table 2. Maximum absolute error for example 1

<table>
<thead>
<tr>
<th>$x$</th>
<th>$a = 0.1$</th>
<th>$a = 0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t = 1$</td>
<td>$t = 3$</td>
</tr>
<tr>
<td>0.1</td>
<td>3.05996($-5$)</td>
<td>3.06791($-5$)</td>
</tr>
<tr>
<td>0.01</td>
<td>3.07903($-7$)</td>
<td>3.07924($-7$)</td>
</tr>
<tr>
<td>0.001</td>
<td>3.08122($-9$)</td>
<td>2.08135($-9$)</td>
</tr>
<tr>
<td>0.5</td>
<td>8.37130($-5$)</td>
<td>8.39331($-5$)</td>
</tr>
<tr>
<td>0.01</td>
<td>8.45321($-7$)</td>
<td>8.45385($-7$)</td>
</tr>
<tr>
<td>0.001</td>
<td>8.46212($-9$)</td>
<td>8.46255($-9$)</td>
</tr>
<tr>
<td>0.9</td>
<td>2.70077($-5$)</td>
<td>2.70793($-5$)</td>
</tr>
<tr>
<td>0.01</td>
<td>2.73069($-7$)</td>
<td>2.73089($-7$)</td>
</tr>
<tr>
<td>0.001</td>
<td>2.73389($-9$)</td>
<td>2.73402($-9$)</td>
</tr>
</tbody>
</table>

Figure 1: Approximate and exact solution for example 1 at $t = 1$, with different $a$. 
Figure 2: Approximate and exact solution for example 1 at $t = 3$, with different $a$

Figure 3: Approximate and exact solution for example 1 at $t = 5$, with different $a$
Example 2. We consider nonlinear damping equation
\[ u_t + auu_x = u_{xx} + bu(1 - u), \quad t \geq 0, \quad 0 \leq x \leq 1 \]
with the following initial and boundary conditions
\[
\begin{align*}
u(x, 0) &= \frac{1}{2} + \frac{1}{2} \tanh \left( \frac{-a}{4} x \right), \quad 0 \leq x \leq 1 \\
u(0, t) &= \frac{1}{2} + \frac{1}{2} \tanh \left[ \frac{-a}{4} \left( -\left( \frac{a}{2} + \frac{2b}{a} \right) t \right) \right], \\
u(1, t) &= \frac{1}{2} + \frac{1}{2} \tanh \left[ \frac{-a}{4} \left( 1 - \left( \frac{a}{2} + \frac{2b}{a} \right) t \right) \right], \quad t \geq 0.
\end{align*}
\]
The exact solution is
\[ u(x, t) = \frac{1}{2} + \frac{1}{2} \tanh \left[ \frac{-a}{4} \left( x - \left( \frac{a}{2} + \frac{2b}{a} \right) t \right) \right] \quad t \geq 0. \]

In our computation, the computed solution are compare with exact solution. The maximum absolute error are reported in table 3. In Table 3, we take \( a = b = 0.001, \ k = 0.00001 \) and \( h = 0.02 \). The results are computed for different time levels. In table 3, results have been compared with the results in references [22]. The result show, our numerical results are more accurate in comparison to those given by Ismail et al. That result has been calculated by 5 terms in Adomian methods. In table 4, we take \( a = 0.001 \) and \( k = 0.0001 \), the results are computed for different step size and different \( b \). The maximum absolute error for time \( t = 1 \) has been computed and tabulated in table 4. In Table 5, we take \( h = 0.05 \) and \( k = 0.00001 \). The result are computed for different \( a \) and \( b \). The maximum absolute error for two time level \( t = 0.5 \) and \( t = 1 \) have been computed and tabulated in table 5. We show the graphs between exact and numerical solutions at \( t = 1 \) and \( a = 0.001 \), with different values of \( h \) and \( b \) in figures 4 – 6.

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t )</th>
<th>[22]</th>
<th>present method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.005</td>
<td>9.68765(−6)</td>
<td>1.20184(−10)</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>1.93753(−6)</td>
<td>3.04431(−11)</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>1.93752(−5)</td>
<td>2.21709(−10)</td>
</tr>
<tr>
<td>0.5</td>
<td>0.005</td>
<td>9.68661(−6)</td>
<td>1.41649(−10)</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>1.93773(−6)</td>
<td>3.05515(−11)</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>1.93773(−5)</td>
<td>2.80496(−10)</td>
</tr>
<tr>
<td>0.9</td>
<td>0.005</td>
<td>9.68619(−6)</td>
<td>1.17931(−10)</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>1.93724(−6)</td>
<td>3.02686(−11)</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>1.93724(−5)</td>
<td>1.9840(−10)</td>
</tr>
</tbody>
</table>
Table 4. Maximum absolute error for example 2

<table>
<thead>
<tr>
<th>x</th>
<th>b = 0.001</th>
<th>b = 0.01</th>
<th>b = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h = 0.1</td>
<td>h = 0.05</td>
<td>h = 0.02</td>
</tr>
<tr>
<td>0.1</td>
<td>3.08469(-8)</td>
<td>7.78978(-9)</td>
<td>1.24965(-9)</td>
</tr>
<tr>
<td>0.2</td>
<td>5.47325(-8)</td>
<td>1.30415(-8)</td>
<td>2.22143(-9)</td>
</tr>
<tr>
<td>0.3</td>
<td>7.16008(-8)</td>
<td>1.81554(-8)</td>
<td>2.91536(-9)</td>
</tr>
<tr>
<td>0.4</td>
<td>8.16355(-8)</td>
<td>2.07319(-8)</td>
<td>3.33144(-9)</td>
</tr>
<tr>
<td>0.5</td>
<td>8.46606(-8)</td>
<td>2.15716(-8)</td>
<td>3.46967(-9)</td>
</tr>
<tr>
<td>0.6</td>
<td>8.07397(-8)</td>
<td>2.06747(-8)</td>
<td>3.33005(-9)</td>
</tr>
<tr>
<td>0.7</td>
<td>6.98764(-8)</td>
<td>1.80417(-8)</td>
<td>2.91259(-9)</td>
</tr>
<tr>
<td>0.8</td>
<td>5.20745(-8)</td>
<td>1.36728(-8)</td>
<td>2.21728(-9)</td>
</tr>
<tr>
<td>0.9</td>
<td>2.73372(-8)</td>
<td>7.56845(-9)</td>
<td>1.2441(-9)</td>
</tr>
<tr>
<td></td>
<td>h = 0.1</td>
<td>h = 0.05</td>
<td>h = 0.02</td>
</tr>
<tr>
<td>0.1</td>
<td>3.09723(-8)</td>
<td>7.82111(-9)</td>
<td>1.25449(-9)</td>
</tr>
<tr>
<td>0.2</td>
<td>5.49535(-8)</td>
<td>1.38968(-8)</td>
<td>2.22997(-9)</td>
</tr>
<tr>
<td>0.3</td>
<td>7.19487(-8)</td>
<td>1.82276(-8)</td>
<td>2.9265(-9)</td>
</tr>
<tr>
<td>0.4</td>
<td>8.19626(-8)</td>
<td>2.08142(-8)</td>
<td>3.34412(-9)</td>
</tr>
<tr>
<td>0.5</td>
<td>8.49993(-8)</td>
<td>2.16571(-8)</td>
<td>3.48284(-9)</td>
</tr>
<tr>
<td>0.6</td>
<td>8.10628(-8)</td>
<td>2.07567(-8)</td>
<td>3.34269(-9)</td>
</tr>
<tr>
<td>0.7</td>
<td>7.01566(-8)</td>
<td>1.81134(-8)</td>
<td>2.92366(-9)</td>
</tr>
<tr>
<td>0.8</td>
<td>5.22838(-8)</td>
<td>1.37274(-8)</td>
<td>2.22573(-9)</td>
</tr>
<tr>
<td>0.9</td>
<td>2.7447(-8)</td>
<td>7.5988(-9)</td>
<td>1.24887(-9)</td>
</tr>
<tr>
<td></td>
<td>h = 0.1</td>
<td>h = 0.05</td>
<td>h = 0.02</td>
</tr>
<tr>
<td>0.1</td>
<td>1.53032(-7)</td>
<td>1.80708(-7)</td>
<td>1.86514(-7)</td>
</tr>
<tr>
<td>0.2</td>
<td>2.71202(-7)</td>
<td>3.19366(-7)</td>
<td>3.29698(-7)</td>
</tr>
<tr>
<td>0.3</td>
<td>3.54373(-7)</td>
<td>4.17367(-7)</td>
<td>4.03961(-7)</td>
</tr>
<tr>
<td>0.4</td>
<td>4.04031(-7)</td>
<td>4.75743(-7)</td>
<td>4.913151(-7)</td>
</tr>
<tr>
<td>0.5</td>
<td>4.20779(-7)</td>
<td>4.95119(-7)</td>
<td>5.11369(-7)</td>
</tr>
<tr>
<td>0.6</td>
<td>4.04834(-7)</td>
<td>4.75713(-7)</td>
<td>4.91322(-7)</td>
</tr>
<tr>
<td>0.7</td>
<td>3.56028(-7)</td>
<td>4.17331(-7)</td>
<td>4.30974(-7)</td>
</tr>
<tr>
<td>0.8</td>
<td>2.73812(-7)</td>
<td>3.19371(-7)</td>
<td>3.29714(-7)</td>
</tr>
<tr>
<td>0.9</td>
<td>1.57255(-7)</td>
<td>1.80824(-7)</td>
<td>1.86529(-7)</td>
</tr>
</tbody>
</table>
Table 5. Maximum absolute error for example 2

<table>
<thead>
<tr>
<th>x</th>
<th>a = b = 0.001 t = 0.5</th>
<th>a = b = 0.001 t = 1</th>
<th>a = b = 0.0001 t = 0.5</th>
<th>a = b = 0.0001 t = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>7.76059(−8)</td>
<td>7.8118(−8)</td>
<td>7.73848(−10)</td>
<td>7.78809(−10)</td>
</tr>
<tr>
<td>0.20</td>
<td>1.3774(−7)</td>
<td>1.38711(−7)</td>
<td>1.37441(−9)</td>
<td>1.38383(−9)</td>
</tr>
<tr>
<td>0.30</td>
<td>1.80401(−7)</td>
<td>1.81813(−7)</td>
<td>1.802181(−9)</td>
<td>1.81513(−9)</td>
</tr>
<tr>
<td>0.40</td>
<td>2.05904(−7)</td>
<td>2.07467(−7)</td>
<td>2.05754(−9)</td>
<td>2.07274(−9)</td>
</tr>
<tr>
<td>0.50</td>
<td>2.14074(−7)</td>
<td>2.15714(−7)</td>
<td>2.14076(−9)</td>
<td>2.15672(−9)</td>
</tr>
<tr>
<td>0.60</td>
<td>2.05038(−7)</td>
<td>2.06597(−7)</td>
<td>2.05189(−9)</td>
<td>2.06705(−9)</td>
</tr>
<tr>
<td>0.70</td>
<td>1.78832(−7)</td>
<td>1.80157(−7)</td>
<td>1.79091(−9)</td>
<td>1.8038(−9)</td>
</tr>
<tr>
<td>0.80</td>
<td>1.35472(−7)</td>
<td>1.36436(−7)</td>
<td>1.35763(−9)</td>
<td>1.36699(−9)</td>
</tr>
<tr>
<td>0.90</td>
<td>7.4967(−8)</td>
<td>7.69406(−8)</td>
<td>7.51754(−10)</td>
<td>7.56673(−9)</td>
</tr>
</tbody>
</table>

Figure 4: Approximate and exact solution for example 2 at b = 1, with different h

Figure 5: Approximate and exact solution for example 2 at b = 0.01, with different h
6. conclusion

The basic goal of this work has been employed the non-polynomial tension spline as a reasonable basis for studying the approximate solutions for Burgers equations with nonlinear damping term. Finite difference approximation for time and tension spline for spatial are used. Presented scheme are of order $O(h^2 + k^2h^2 + h^4)$ and under appropriate condition the method convergence. The performance and accuracy of the method have been examined by applying in 2 examples.
References


