A Haar wavelet quasilinearization approach for numerical simulation of Burgers’ equation

Ram Jiwari*

School of Mathematics & Computer Applications, Thapar University, Patiala 147004, India

A R T I C L E   I N F O

Article history:
Received 19 April 2012
Received in revised form 31 May 2012
Accepted 16 June 2012
Available online 19 June 2012

Keywords:
Haar wavelets
Burgers’ equation
Multi-resolution analysis
Collocation points

A B S T R A C T

In this paper, an efficient numerical scheme based on uniform Haar wavelets and the quasilinearization process is proposed for the numerical simulation of time dependent nonlinear Burgers’ equation. The equation has great importance in many physical problems such as fluid dynamics, turbulence, sound waves in a viscous medium etc. The Haar wavelet basis permits to enlarge the class of functions used so far in the collocation framework. More accurate solutions are obtained by wavelet decomposition in the form of a multi-resolution analysis of the function which represents a solution of boundary value problems. The accuracy of the proposed method is demonstrated by three test problems. The numerical results are compared with existing numerical solutions found in the literature. The use of the uniform Haar wavelet is found to be accurate, simple, fast, flexible, convenient and has small computation costs.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Consider the one-dimensional Burgers’ equation

\[
\frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial x} = 0, \quad (x, t) \in \Omega \times (0, T),
\]

(1)

with the initial condition

\[
u(x, 0) = f(x), \quad 0 \leq x \leq 1,
\]

(2a)

and the boundary conditions

\[
u(0, t) = f_1(t), \quad \nu(1, t) = f_2(t), \quad 0 \leq t \leq T
\]

(2b)

where \(\Omega = (0, 1), \nu > 0\) is the coefficient of kinematic viscosity and the prescribed function \(f(x)\) is sufficiently smooth.

The nonlinear homogeneous quasilinear parabolic partial differential equation is the simplest nonlinear model equation for diffusive waves in fluid dynamics. Burgers’ equation arises in many physical problems including one-dimensional turbulence, sound waves in a viscous medium, shock waves in a viscous medium, waves in fluid filled viscous elastic tubes, and magnetohydrodynamic waves in a medium with finite electrical conductivity. Such type of equation was first introduced by Bateman [1] in 1915 and he proposed the steady-state solution of the problem. In 1948, Burgers [2,3] introduced this equation to capture some features of turbulent fluid in a channel caused by the interaction of the opposite effects of convection and diffusion. Therefore, Eq. (1) is referred to as “Burgers’ equation”. The structure of Burgers’ equation is roughly similar to that of Navier–Stoke’s equations due to the presence of the nonlinear convection term and the occurrence of the diffusion term with viscosity coefficient. So this equation can be considered as a simplified form of Navier–Stoke’s equations. The study of the general properties of Burgers’ equation has attracted attention of scientific community due to its applications in various fields such as gas dynamics, heat conduction, elasticity, etc.

The study of the solution of Burgers’ equation has been carried out for last half Century and still it is an active area of research to develop some better numerical scheme to approximate its solution. So far various numerical algorithms such as the automatic differentiation method [4], Galerkin finite element method [5], cubic B-splines collocation method [6], spectral collocation method [7,8], Sinc Differential Quadrature Method [9], Polynomial based differential quadrature method [10], Quartic B-splines Differential Quadrature Method [11], Quartic B-splines collocation method [12], Quadratic B-splines finite element method [13], finite element method [14], Fourth-order finite difference method [15], a parameter-uniform implicit difference scheme [16], A novel numerical scheme [17], explicit and exact-explicit finite difference methods [18], least-squares quadratic B-splines finite element method [19], implicit fourth-order compact finite difference scheme [20], some implicit methods [21], Adomian–Padé technique [22], variational iteration method [23], homotopy analysis method [24], differential transform method and the homotopy analysis method [25], Semi-Implicit Finite Difference Schemes [26], modified cubic B-splines collocation method [27] etc.
Wavelets have been used for the solution of partial differential equations (PDEs) since the 1980s. The good features of this approach are possibility to detect singularities, irregular structure and transient phenomena exhibited by the analyzed equations. Most of the wavelets algorithms for solving PDE are based on the collocation [28,29] or the Galerkin techniques method [28,30,31].

Evidently, all attempts to simplify the wavelet solutions for PDE are welcome; one of the possibilities for this is to make use of the Haar wavelet family. Haar wavelets which are Daubechies of order 1 consist of piecewise constant functions and are therefore the simplest orthonormal wavelets with a compact support. The Haar wavelets have gained popularity among researchers for their useful properties such as simple applicability, orthogonality and compact support. Compact support of the Haar wavelet basis permits straight inclusion of the different types of boundary conditions in the numeric algorithms. A drawback of the Haar wavelets is their discontinuity. Since the derivatives do not exist in the breaking points it is not possible to apply the Haar wavelets for solving PDE directly. Lepik [32–34] had solved higher order as well as nonlinear ODEs by using the Haar wavelet method. Recently, Hariharan and his associates [35,36] have used Haar wavelets to enlarge the class of functions used so far in the collocation [28,29] or the Galerkin techniques method [28,30,31].

In this paper, an efficient numerical scheme based on uniform Haar wavelets and quasilinearization process is proposed for the numerical simulation of the non-linear homogeneous quasilinear parabolic Burger’s equation. The Haar wavelet basis permits to enlarge the class of functions used so far in the collocation framework. More accurate solutions are obtained by wavelet decomposition in the form of a multi-resolution analysis of the function which represents solutions of boundary value problems. The accuracy of the proposed scheme is demonstrated by three test problems. The numerical results are compared with existing numerical and exact solutions and it is found that the proposed scheme produce better results. The use of uniform Haar wavelet is found to be accurate, simple, fast, flexible, convenient and has small computation costs.

2. Haar wavelets

Wavelet transform or wavelet analysis is a recently developed mathematical tool for many problems. One of the popular families of wavelet is Haar wavelets. The Haar function is in fact the Daubechies wavelet of order 1. Due to its simplicity, the Haar wavelet had become an effective tool for solving many problems arising in many branches of sciences. Haar functions have been used since 1910. It was introduced by the Hungarian mathematician Alfred Haar. The Haar function is an odd rectangular pulse pair, is the simplest and oldest orthonormal wavelet with compact support. There are different definitions for the Haar function and various generalizations have been used and published. Haar showed that certain square wave function could be translated and scaled to create a basis set that span L^2. Year later, it was seen that the system of Haar is a particular wavelet system. If we choose scaling function to have compact support over 0 ≤ x ≤ 1, that is the Haar wavelet family for x ∈ [0, 1] is defined as

\[ h_i(x) = \begin{cases} 
1 & x \in [\xi_1, \xi_2), \\
-1 & x \in [\xi_2, \xi_3), \\
0 & \text{otherwise}
\end{cases} \] (3)

where

\[ \xi_1 = \frac{k}{m}, \quad \xi_2 = \frac{k + 0.5}{m}, \quad \xi_3 = \frac{k + 1}{m}. \] (4)

In the above definition the integer \( m = 2^j, \) \( j = 0, 1, \ldots, J \) indicates the level of the wavelet and the integer \( k = 0, 1, \ldots, m - 1 \) is the translation parameter. The maximal level of resolution is \( J. \)

The index \( i \) in Eq. (4) is calculated from the formula \( i = m + k + 1. \) In the case of minimal values \( m = 1, k = 0, \) we have \( i = 2. \) The maximum value of \( i \) is \( i = 2^J = 2^{J+1}. \) For \( i = 1, \) the function \( h_1(x) \) is the scaling function for the family of the Haar wavelets which is defined as

\[ h_1(x) = \begin{cases} 
1 & x \in [0, 1), \\
0 & \text{otherwise}.
\end{cases} \] (5)

If we want to solve partial differential equations of any order, we need the following integrals

\[ p_{i,1}(x) = \int_0^x h_i(x')dx', \]
\[ p_{i,v+1}(x) = \int_0^x p_{i,v}(x')dx', \quad v = 1, 2, 3, \ldots \]

Taking into account (3) the integrals (6) can be calculated analytically; by doing so we obtained the following

\[ p_{i,1}(x) = \begin{cases} 
\frac{(x - \xi_1)^2}{2} & x \in [\xi_1, \xi_2), \\
\frac{1}{4m^2} - \frac{(\xi_1 - x)^2}{2} & x \in [\xi_2, \xi_3), \\
0 & \text{otherwise}
\end{cases} \] (7)
\[ p_{i,2}(x) = \begin{cases} 
\frac{(x - \xi_1)^2}{2} & x \in [\xi_1, \xi_2), \\
\frac{1}{4m^2} - \frac{(\xi_1 - x)^2}{2} & x \in [\xi_2, \xi_3), \\
0 & \text{otherwise}
\end{cases} \] (8)

3. Function approximation

Any function \( y(x) \) which is square integrable in the interval \((0, 1)\) can be expressed in the following form of Haar wavelets

\[ y(x) = \sum_{i=1}^{\infty} a_i h_i(x). \] (9)

The above series terminates at finite terms if \( y(x) \) is piecewise constant or can be approximated as piecewise constant during each subinterval, then \( y(x) \) will be terminated at finite terms, that is

\[ y(x) = \sum_{i=1}^{2M} a_i h_i(x) = a_{2M}^T h_{2M}(x), \] (10)

where the coefficients \( a_{2M}^T \) and the Haar function vector \( h_{2M}(x) \) are defined as

\[ a_{2M}^T = [a_1, a_2, \ldots, a_{2M}] \quad \text{and} \quad h_{2M}(x) = [h_{1}(x), h_{2}(x), \ldots, h_{2M}(x)]^T, \]

where \( T \) denotes the transpose and \( M = 2^j. \)

The best way to understand wavelets is through a multi-resolution analysis. Given a function \( y(x) \in L^2(R) \) a multi-resolution analysis (MRA) of \( L^2(R) \) produces a sequence of subspaces \( U_j, U_{j+1}, \ldots \) such that the projections of \( y(x) \) onto these spaces give finer and finer approximations of the function \( y(x) \) as \( j \to \infty. \) The details of MRA is as follows.

3.1. Multi-resolution analysis

A multi-resolution analysis of \( L^2(R) \) is defined as a sequence of closed subspaces \( U_j \subset L^2(R), \) \( j \in \mathbb{Z} \) with the following properties.

(i) \( \cdots \subset U_{-1} \subset U_0 \subset U_1 \subset \cdots \)
(ii) The space \( U_j \) satisfying \( \bigcup_{j \in \mathbb{Z}} U_j \) is dense in \( L^2(R) \) and \( \bigcap_{j \in \mathbb{Z}} U_j = 0. \)
Fig. 1. Physical behavior of numerical solutions of Example 1 for \( \nu = 0.1 \) at different times \( t \) with \( \Delta t = 0.001 \).

Fig. 2. Physical behavior of numerical solutions of Example 1 for \( \nu = 0.01 \) at different times \( t \) with \( \Delta t = 0.001 \).

Fig. 3. Physical behavior of numerical solutions of Example 1 in 3D and contour for \( \nu = 0.1 \) at different times \( t \) with \( \Delta t = 0.001 \).

(iii) If \( g(x) \in U_0, g(2^j x) \in U_j \) i.e. the space \( U_j \) is a scaled version of the central space \( U_0 \).

(iv) If \( g(x) \in U_0, g(2^j x - k) \in U_j \) i.e. all the \( U_j \) are invariant under translation.

(v) There exists \( \Phi \in U_0 \) such that \( \Phi(x - k), k \in \mathbb{Z} \) is a Riesz basis in \( U_0 \). The space \( U_j \) is used to approximate general functions by defining appropriate projection of these functions onto these spaces. Since the union of all the \( U_j \) is dense in \( L^2(\mathbb{R}) \), it guarantees that any function in \( L^2(\mathbb{R}) \) can be approximated arbitrarily close by such projections. As an example the space \( U_j \) can be defined like

\[
U_j = W_{j-1} \oplus U_{j-1} = W_{j-1} \oplus W_{j-2} \oplus V_{j-2} = \cdots = \bigoplus_{j=1}^{j+1} W_{j} \oplus V_0
\]

then the scaling function \( h_t(x) \) generates an MRA for the sequence of spaces \( \{U_j, j \in \mathbb{Z}\} \) by translation and dilation as defined in Eqs. (3) and (5). For each \( j \) the space \( W_j \) serves as the orthogonal
complement of $U_j$ in $U_{j+1}$. The space $W_j$ includes all the functions in $U_{j+1}$ that are orthogonal to all those in $U_j$ under some chosen inner product. The set of functions which form basis for the space $W_j$ are called wavelets [37,38].

4. Uniform Haar wavelet based scheme for Burgers’ equation

This section presents the uniform Haar wavelet and quasilinearization approach based scheme for Burgers’ equation (1) with initial and boundary conditions (2).

Let us divide the interval $(0, 1]$ into $N$ equal parts of length $\Delta t = 1/N$ and denote $t_s = (s-1)\Delta t$, $s = 1, 2, \ldots, N$. We assume the following for the solution of Eq. (1), assumed by Lepik [33]

$$\dot{u}^r(x, t) = \sum_{i=1}^{2M} a_r(i) h_i(x) = a_r^{(2M)} h_{(2M)}(x)$$  \hspace{1cm} (11)

where dot and dash denote the differentiation with respect to $t$ and $x$, respectively, the row vector $a_r$ is constant in the subinterval $t \in [t_s, t_{s+1}]$. 

Table 1
Comparison with exact and existing numerical methods of Example 1 for $\nu = 0.1$ at different times $t$ and $x$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$t$</th>
<th>[18] $\Delta t = 0.001$</th>
<th>[14] $\Delta t = 0.01$</th>
<th>[16] $\Delta t = 0.01$</th>
<th>Present method $\Delta t = 0.001$</th>
<th>Exact solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.30891</td>
<td>0.31429</td>
<td>0.30881</td>
<td>0.30887</td>
<td>0.30889</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4</td>
<td>0.24075</td>
<td>0.24373</td>
<td>0.24069</td>
<td>0.24070</td>
<td>0.24074</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4</td>
<td>0.19568</td>
<td>0.19758</td>
<td>...</td>
<td>0.19566</td>
<td>0.19568</td>
</tr>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>0.16257</td>
<td>0.16391</td>
<td>0.16254</td>
<td>0.16255</td>
<td>0.16256</td>
</tr>
<tr>
<td>3.0</td>
<td>0.4</td>
<td>0.02720</td>
<td>0.02743</td>
<td>0.02720</td>
<td>0.02721</td>
<td>0.02720</td>
</tr>
<tr>
<td>0.25</td>
<td>0.6</td>
<td>0.56964</td>
<td>0.57636</td>
<td>0.56955</td>
<td>0.56956</td>
<td>0.56963</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6</td>
<td>0.44721</td>
<td>0.45169</td>
<td>0.44714</td>
<td>0.44715</td>
<td>0.44721</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6</td>
<td>0.35924</td>
<td>0.36245</td>
<td>...</td>
<td>0.35920</td>
<td>0.35924</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.29192</td>
<td>0.29437</td>
<td>0.29188</td>
<td>0.29188</td>
<td>0.29192</td>
</tr>
<tr>
<td>3.0</td>
<td>0.6</td>
<td>0.04021</td>
<td>0.04057</td>
<td>0.04021</td>
<td>0.04022</td>
<td>0.04021</td>
</tr>
<tr>
<td>0.25</td>
<td>0.75</td>
<td>0.62542</td>
<td>0.62592</td>
<td>0.62540</td>
<td>0.62540</td>
<td>0.62544</td>
</tr>
<tr>
<td>0.6</td>
<td>0.75</td>
<td>0.48721</td>
<td>0.49034</td>
<td>0.48715</td>
<td>0.48716</td>
<td>0.48721</td>
</tr>
<tr>
<td>0.8</td>
<td>0.75</td>
<td>0.37392</td>
<td>0.37713</td>
<td>...</td>
<td>0.37389</td>
<td>0.37392</td>
</tr>
<tr>
<td>1.0</td>
<td>0.75</td>
<td>0.28748</td>
<td>0.29015</td>
<td>0.28744</td>
<td>0.28743</td>
<td>0.28747</td>
</tr>
<tr>
<td>3.0</td>
<td>0.75</td>
<td>0.02977</td>
<td>0.03434</td>
<td>0.02978</td>
<td>0.02978</td>
<td>0.02977</td>
</tr>
</tbody>
</table>
Integrating Eq. (11) with respect to \( t \) from \( t_{s_0} \) to \( t \), we get the following
\[
\overline{u}(x, t) = (t - t_{s_0}) \overline{a}_{2M}^2 h_{2M}(x) + u'(x, t_{s_0}). \tag{12}
\]
Now, integrating Eq. (12) twice with respect to \( x \) from 0 to \( x \), we obtain
\[
\overline{u}(x, t) = (t - t_{s_0}) \overline{a}_{2M}^2 P_{2,2M} h_{2M}(x) + u'(x, t_{s_0}) + \frac{u'(0, t) - u'(0, t_{s_0})}{2} \tag{13}
\]
\[
\overline{u}(x, t) = (t - t_{s_0}) \overline{a}_{2M}^2 P_{2,2M} h_{2M}(x) + u(0, t) + x u'(0, t) - \frac{u(0, t)}{2} \tag{14}
\]
\[
\overline{u}(x, t) = \overline{a}_{2M}^2 P_{2,2M} h_{2M}(x) + \frac{u(0, t)}{2} + x u'(0, t). \tag{15}
\]
By using the boundary conditions (2b), we obtain
\[
\overline{u}(0, t) = f_1(t), \quad u(1, t) = f_2(t). \tag{16}
\]
Putting \( x = 1 \) in Eqs. (14) and (15) and using the conditions in (16), we obtain
\[
\overline{u}'(0, t) = f_1'(t) - f_2(t), \quad \overline{u}'(1, t) = f_2'(t). \tag{17}
\]
Substituting Eqs. (16)–(18) into Eqs. (13)–(15), and discretizing the results by \( x \rightarrow x_i, \quad t \rightarrow t_{s_0} \), we have
\[
\overline{u}'(x_i, t_{s_0}) = \Delta t \overline{a}_{2M}^2 h_{2M}(x_i) + u'(x_i, t_{s_0}). \tag{19}
\]
Table 5
Comparison with exact and existing numerical methods of Example 2 for \( v = 0.01 \) at different times \( t \) and \( x \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t )</th>
<th>[19] ( \Delta t = 0.0001 )</th>
<th>[16] ( \Delta t = 0.01 )</th>
<th>Present method ( \Delta t = 0.001 )</th>
<th>Exact solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.36911</td>
<td>0.39273</td>
<td>0.36217</td>
<td>0.36226</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.28905</td>
<td>0.28212</td>
<td>0.28197</td>
<td>0.28204</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.23703</td>
<td>...</td>
<td>0.23040</td>
<td>0.23045</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.20609</td>
<td>0.19467</td>
<td>0.19465</td>
<td>0.19469</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.18705</td>
<td>0.17631</td>
<td>0.17631</td>
<td>0.17631</td>
</tr>
<tr>
<td>0.50</td>
<td>0.4</td>
<td>0.68818</td>
<td>0.61816</td>
<td>0.68357</td>
<td>0.68368</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.55425</td>
<td>0.55125</td>
<td>0.54822</td>
<td>0.54832</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.46011</td>
<td>...</td>
<td>0.45363</td>
<td>0.45371</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.39206</td>
<td>0.38627</td>
<td>0.38561</td>
<td>0.38568</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.33501</td>
<td>0.33021</td>
<td>0.32966</td>
<td>0.32973</td>
</tr>
<tr>
<td>0.75</td>
<td>0.4</td>
<td>0.92194</td>
<td>0.94940</td>
<td>0.92050</td>
<td>0.92050</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.78676</td>
<td>0.79399</td>
<td>0.78299</td>
<td>0.78299</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.66777</td>
<td>...</td>
<td>0.66264</td>
<td>0.66272</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.57491</td>
<td>0.57170</td>
<td>0.56924</td>
<td>0.56932</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>0.49305</td>
<td>0.48993</td>
<td>0.48747</td>
<td>0.48751</td>
</tr>
</tbody>
</table>

u’(x, t_{i+1}) = \Delta t a^0_{(2M)} P_1(2M) h_{(2M)}(x_i) - \Delta t a^0_{(2M)} P_2(2M) F
+ f'_2(t_{i+1}) - f_1(t_{i+1}) + u(x, x_i)

(20)

u(x, t_{i+1}) = \Delta t a^0_{(2M)} P_2(2M) h_{(2M)}(x_i) + u(x, x_i)
- f_1(t_i) + f_1(t_{i-1}) + x_i (\Delta t a^0_{(2M)} P_2(2M) F)
+ f_2(t_{i+1}) - f_1(t_{i+1}) - f_2(t_i) + f_1(t_i))

(21)

\hat{u}(x, t_{i+1}) = \Delta t a^0_{(2M)} P_2(2M) h_{(2M)}(x_i) + f'_1(t_{i+1})
+ x_i (\Delta t a^0_{(2M)} P_2(2M) F) + f'_2(t_{i+1}) - f'_1(t_{i+1}))

(22)

where the vector \( F = [1, 0, \ldots, 0]^T \).

5. Numerical experiments

In this section, three test examples are considered to check the efficiency and accuracy of the proposed scheme. In order to measure the accuracy of the numerical scheme error norm \( L_0 \) and \( L_2 \) are calculated. Lagrange’s interpolation is used to find the solution at specified points. The whole computational work has been done with the help of MATLAB software.

Example 1. In this example, the author considered Burgers’ equation (1) with initial and boundary conditions in the following form [5,16]

\( u(x, 0) = \sin \pi x, \quad 0 \leq x \leq 1 \),

(26)

\( u(0, t) = u(1, t) = 0, \quad t > 0 \).

(27)

The exact solution of example is obtained by Hopf–Cole transformation and given by

\[
\begin{align*}
\hat{u}(x, t) &= \frac{2\pi v}{A_0 + \sum_n A_n \exp(-n^2\pi^2 v t) \cos(n\pi x)}
\end{align*}
\]

(28)

where

\[
A_0 = \int_0^1 \exp\left(-\frac{1}{2\pi v} (1 - \cos(\pi x)) \right) \, dx,
\]

(29)

\[
A_n = 2 \int_0^1 \exp\left(-\frac{1}{2\pi v} (1 - \cos(\pi x)) \right) \, dx.
\]

The numerical solutions of the example are presented for \( v = 0.1, 0.01, 0.005, 0.004, 0.003 \) with \( \Delta t = 0.001, M = 64 \) in Tables 1–3 and Figs. 1–5. The results are compared with [14–16,18,19] and it is found that these are much better than the results presented in [14,18,19]. The figures are depicted up to time \( t \leq 3 \), which exhibit correct physical behavior of the problem. Figs. 3 and 4 show the physical behavior of numerical solutions in 3D and contour which is similar to the 2D figures. Fig. 5 compares the exact and numerical solutions at different times.

Example 2. Consider Burgers’ equation (1) with the following initial and boundary conditions [4,13,17]

\( u(x, 0) = 4x(1 - x), \quad 0 \leq x \leq 1 \),

(30)

\( u(0, t) = u(1, t) = 0, \quad t > 0 \).

(31)
Table 6
Comparison of exact and numerical solutions of Example 2 for \( \nu = 0.005, 0.004, 0.003 \) at different times \( t \) and \( x \) with \( \Delta t = 0.001 \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t = 0.005 )</th>
<th>( t = 0.004 )</th>
<th>( t = 0.003 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Present method</td>
<td>Exact solution</td>
<td>Present method</td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.19604</td>
<td>0.19609</td>
<td>0.19636</td>
</tr>
<tr>
<td>5</td>
<td>0.04741</td>
<td>0.04741</td>
<td>0.04744</td>
</tr>
<tr>
<td>10</td>
<td>0.02433</td>
<td>0.02434</td>
<td>0.02434</td>
</tr>
<tr>
<td>15</td>
<td>0.01636</td>
<td>0.01636</td>
<td>0.01637</td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.38795</td>
<td>0.38797</td>
<td>0.38842</td>
</tr>
<tr>
<td>5</td>
<td>0.09481</td>
<td>0.09482</td>
<td>0.09491</td>
</tr>
<tr>
<td>10</td>
<td>0.04866</td>
<td>0.04868</td>
<td>0.04869</td>
</tr>
<tr>
<td>15</td>
<td>0.03255</td>
<td>0.03255</td>
<td>0.03270</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.57248</td>
<td>0.57250</td>
<td>0.57312</td>
</tr>
<tr>
<td>5</td>
<td>0.14215</td>
<td>0.14215</td>
<td>0.14224</td>
</tr>
<tr>
<td>10</td>
<td>0.07152</td>
<td>0.07151</td>
<td>0.07258</td>
</tr>
<tr>
<td>15</td>
<td>0.04433</td>
<td>0.04432</td>
<td>0.04696</td>
</tr>
</tbody>
</table>

Fig. 6. Physical behavior of numerical solutions of Example 2 for \( \nu = 0.1 \) at different times \( t \) with \( \Delta t = 0.001 \).

Fig. 7. Physical behavior of numerical solutions of Example 2 for \( \nu = 0.01 \) at different times \( t \) with \( \Delta t = 0.001 \).

Table 7
Absolute error of Example 3 for \( \alpha = 2, \Delta t = 0.001 \) at different times \( t \) and \( \nu \).

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( t = 0.001 )</th>
<th>( t = 0.01 )</th>
<th>( t = 0.1 )</th>
<th>( t = 0.5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>5.71475E−06</td>
<td>4.19416E−05</td>
<td>1.29318E−04</td>
<td>1.93815E−04</td>
</tr>
<tr>
<td>0.5</td>
<td>7.27810E−07</td>
<td>6.20899E−06</td>
<td>3.44228E−05</td>
<td>8.37713E−05</td>
</tr>
<tr>
<td>0.2</td>
<td>6.71062E−08</td>
<td>4.45567E−07</td>
<td>6.36510E−06</td>
<td>2.47738E−05</td>
</tr>
<tr>
<td>0.1</td>
<td>5.91078E−09</td>
<td>3.80581E−08</td>
<td>1.40456E−06</td>
<td>6.52963E−06</td>
</tr>
<tr>
<td>0.01</td>
<td>5.93092E−12</td>
<td>6.03414E−11</td>
<td>2.88838E−09</td>
<td>5.78971E−08</td>
</tr>
<tr>
<td>0.0001</td>
<td>8.13151E−18</td>
<td>8.67802E−17</td>
<td>6.56381E−15</td>
<td>8.86360E−14</td>
</tr>
<tr>
<td>0.00001</td>
<td>4.47233E−19</td>
<td>7.13201E−19</td>
<td>8.46789E−18</td>
<td>1.87445E−16</td>
</tr>
</tbody>
</table>
Fig. 8. Physical behavior of numerical solutions of Example 2 in 3D and contour for \( \nu = 0.1 \) at different times \( t \) with \( \Delta t = 0.001 \).

Fig. 9. Physical behavior of numerical solutions of Example 2 in 3D and contour for \( \nu = 0.01 \) at different times \( t \) with \( \Delta t = 0.001 \).

Fig. 10. Physical behavior of numerical (left) and exact (right) solutions of Example 2 for \( \nu = 0.01 \) at different times \( t \) with \( \Delta t = 0.001 \).

<table>
<thead>
<tr>
<th>( N )</th>
<th>Kayser [26] ( L_\infty )</th>
<th>Mittal and Jain [27] ( L_\infty )</th>
<th>Present method ( L_\infty )</th>
<th>( L_2 )</th>
<th>Present method ( L_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.8808E−07</td>
<td>3.4545E−07</td>
<td>1.60651E−08</td>
<td>3.4545E−07</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1.4305E−07</td>
<td>1.0124E−07</td>
<td>2.12402E−09</td>
<td>1.0124E−07</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>5.6677E−08</td>
<td>4.0028E−08</td>
<td>1.64798E−09</td>
<td>4.0028E−08</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>3.4992E−08</td>
<td>4.0028E−08</td>
<td>2.48655E−09</td>
<td>4.0028E−08</td>
<td></td>
</tr>
</tbody>
</table>

Comparison of \( L_\infty \) and \( L_2 \) errors with existing numerical methods of Example 3 for \( \nu = 0.01 \), \( \alpha = 100 \), \( \Delta t = 0.01 \) at \( t = 1.0 \).
Table 9
Comparison of $L_\infty$ and $L_2$ errors with existing numerical methods of Example 3 for $\nu = 0.005$, $\alpha = 100$, $\Delta t = 0.01$ at $t = 1.0$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Kaysar [26] $L_\infty$</th>
<th>Mittal and Jain [27] $L_\infty$</th>
<th>Present method $L_\infty$</th>
<th>$2M$ $L_\infty$</th>
<th>$2M$ $L_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.2458E−07</td>
<td>8.8189E−08</td>
<td>1215E−07</td>
<td>6E−08</td>
<td>1.27757E−10</td>
</tr>
<tr>
<td>20</td>
<td>3.9944E−08</td>
<td>2.4029E−08</td>
<td>3.062E−08</td>
<td>1.27757E−10</td>
<td>2.135E−08</td>
</tr>
<tr>
<td>40</td>
<td>1.1249E−08</td>
<td>7.9424E−09</td>
<td>7.644E−09</td>
<td>3.062E−08</td>
<td>5.378E−08</td>
</tr>
<tr>
<td>80</td>
<td>5.5490E−09</td>
<td>3.9178E−09</td>
<td>7.644E−09</td>
<td>2.135E−08</td>
<td>1.345E−09</td>
</tr>
</tbody>
</table>

Fig. 11. Physical behavior of numerical solutions of Example 3 at $t = 0.001$, $\alpha = 2$ (left) and for $t \leq 4$ with $\Delta t = 0.0001$, $\nu = 0.01$, $\alpha = 100$ (right).

Fig. 12. Physical behavior of numerical solutions of Example 3 in 3D and contour for different values of $\nu$ with $\Delta t = 0.0001$ at $t = 0.001$.

The exact solution of the example is obtained by the Hopf–Cole transformation and is given by

$$u(x, t) = \frac{2\pi \nu \sum_{n=1}^{\infty} A_n \exp(-n^2 \pi^2 \nu t) \sin(n\pi x)}{A_0 + \sum_{n=1}^{\infty} A_n \exp(-n^2 \pi^2 \nu t) \cos(n\pi x)}$$  \hspace{1cm} (32)

where

$$A_0 = \int_0^1 \frac{-1}{3\nu} \left(3x^2 - 2x^3\right) dx,$$
$$A_n = \int_0^1 \frac{-1}{3\nu} \left(3x^2 - 2x^3\right) \cos(n\pi x) dx.$$  \hspace{1cm} (33)

In this example, numerical solutions are presented for $\nu = 0.1$, 0.01, 0.005, 0.004, 0.003 with $\Delta t = 0.001$, $M = 64$ in Tables 4–6 and Figs. 6–10. The results are compared with [14–16,19] and it is found that these are much better than the results presented in [14,16,19]. The figures are depicted up to time $t \leq 3$, which exhibit correct physical behavior of the problem. Figs. 8 and 9 show the physical behavior of numerical solutions in 3D and contour which is similar to the 2D figures while Fig. 10 compares the exact and numerical solutions at different times.

Example 3. Considering Burgers’ equation (1) with boundary conditions [4,26]

$$u(0, t) = u(1, t) = 0, \quad t > 0,$$  \hspace{1cm} (34)

and with an exact solution

$$u(x, t) = \frac{2\nu \pi e^{-\pi^2 \sigma t} \sin(\pi x)}{\sigma + e^{-\pi^2 \sigma t} \cos(\pi x)}, \quad 0 \leq x \leq 1,$$  \hspace{1cm} (35)

where $\sigma > 1$ is a parameter.

The numerical solutions of the example are presented for $\nu = 1, 0.5, 0.2, 0.1, 0.01, 0.0001, 0.00001$ with $\Delta t = 0.001$, $M = 64$.
in Tables 7–9 and Figs. 11–13. Tables 7–9 show the $L_2$ and $L_\infty$ errors at different values of $T$, $v$, $\alpha$. The results are compared with [4,26,27] and it is found that these are more better than the results presented in [4,26,27]. The errors $L_2$ and $L_\infty$ are smaller than the errors presented in [26,27]. The physical behavior of the solution depicted in Fig. 11 is similar as depicted in [26,27]. Fig. 12 shows the physical behavior of numerical solutions in 3D and contour plot while Fig. 13 compares the exact and numerical solutions at different times.

6. Conclusions

In this paper, an efficient numerical scheme based on Haar wavelets and the quasilinearization process is developed for solving nonlinear Burgers’ equation with Dirichlet’s boundary conditions. The scheme is tested on three problems and the obtained numerical results are quite satisfactory. The obtained numerical results are compared with the existing numerical and exact solutions. It is concluded that the present technique gives better accuracy in comparison to the other numerical techniques [4,14,16,18,19,26,27] available in the literature. The main advantage of the Haar wavelet based scheme is that the present scheme is able to capture the behavior of numerical solutions at a small coefficient of kinematic viscosity $v = 0.1$, $0.01$, $0.005$, $0.004$, $0.003$, where most of the numerical methods fail. The present scheme with some modifications seems to be easily extended to solve model equations including more mechanical, physical or biophysical effects, such as nonlinear convection, reaction, linear diffusion and dispersion.

Acknowledgments

The author would like to express his thanks to Prof. R.C. Mittal (Head, Department of Mathematics, Indian Institute of Technology Roorkee, India) for his illuminating advice and valuable discussion. Also, the author is very grateful to the reviewers for their valuable suggestions to improve the quality of the paper.

References


