CHAPTER IX

HAAR WAVELETS

APPROACH FOR SOLVING

ONE-DIMENSIONAL

BURGERS’ EQUATIONS‡

‡ The part of this chapter has been accepted as “Haar wavelets approach for solving one-dimensional Burgers’ equations” in Intl. J.Appl. Math. Sci., (IJAMS) (2010).
Chapter 9

Haar Wavelets approach for solving one-dimensional Burgers’ equations

9.1 Introduction

The Burgers’ equation, which is a nonlinear partial differential equation of second order, is used in disciplines as a simplified model for turbulence, boundary layer behavior, shock wave formation and mass transport. The equation serves as a nonlinear analog of the fluid mechanics equations because it has terms, which closely duplicate the physical properties, i.e., a convective term, a diffusive term, and a time dependent term. Nonlinear Partial differential equations (NLPDEs) arise in many fields of science, particularly in physics, engineering, chemistry and finance, and are fundamental for the mathematical formulation of continuum models. Systems of NLPDEs have attracted much attention in studying evolution equations describing wave propagation, in investing the shallow water waves
[128], and in examining the chemical reaction-diffusion model of Brusselator [8]. Fletcher [65] using the Hopf-Cole transformation gave the analytical solution for the system of two-dimensional Burgers’ equations. Several numerical methods to solve this system have been given such as algorithms based on cubic spline function technique [[104],[118]] the explicit-implicit method [205]. Alice Gorguis [68] using a comparison between Cole-Hopf transformation and the decomposition method for solving Burgers’ equations. We consider a one-dimensional quasi-linear parabolic partial differential equation

\[
\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} = v \frac{\partial^2 U}{\partial x^2} \tag{9.1}
\]

with the initial condition

\[ u(x, 0) = \phi(x), a < x < b. \tag{9.2} \]

and the boundary conditions

\[ U(a, t) = f(t), U(b, t) = g(t), t > 0. \tag{9.3} \]

where \( v > 0 \) is the coefficient of kinematic viscosity, \( f \) and \( g \) are the prescribed functions of the variables.

A numerical solution of Eq. (9.1) has been discussed in many papers. The conventional methods are i) Fourier spectral method

ii) Galerkin and collocation methods

iii) Finite difference and element methods

Historically, equation (9.1) was first introduced by Bateman [40] who gave its steady solutions. It was later treated by Burgers [173] as a mathematical model
for turbulence and after when such an equation is widely referred to as Burger’s equation. The distinctive feature of the Burgers equation is that it is the simplest mathematical formulation of the competition between convection and diffusion. Another feature of the Burgers equation is that although it does not have a pressure gradient term it still is a good approximation of the propagation of one-dimensional disturbances. Performance of a numerical method can be judged from its ability to resolve the large gradient region that develops in the solution. Many problems can be modeled by the Burger’s equation. For example, the Burger’s equation can be considered as an approach to the Navier-Stokes equation since both contain nonlinear terms of the type: unknown functions multiplied by a first derivative and both contain higher-order terms multiplied by a small parameter.

The exact solutions of the one-dimensional, Burgers’ equation have been surveyed by Benton and Platzman [21]. Many other authors have used a variety of numerical techniques based on finite difference, finite element and boundary element methods in attempting to solve the equation particularly for small values of the kinematic viscosity, which correspond to steep fronts in the propagation of dynamic waveforms were designed and well documented in the literature [9], [10], [19], [36], [68], [151], [152], [168]. Abd-el-Malek and El-Mansi [2] have used the group-theoretic methods for calculating the solution of Burgers’ equation with appropriate boundary and initial conditions. Joaquin Zueco [106] established a network thermodynamic method for the numerical solution of Burgers’ equation.

The detailed relationship between Eq.(9.1) and both turbulence theory and shock wave theory was described by Cole. He also gave an exact solution of Burgers’ equation. Benton and Platzman [21] have demonstrated about 35 distinct exact solutions of Burgers-like equations and their classifications. It is well
known that the exact solution of Burgers’ equation can only be computed for restricted values of which represent the kinematics viscosity of the fluid motion. Because of this fact, various numerical methods were employed to obtain the solution of Burgers’ with small values. It is not our purpose to exhaust all existing numerical schemes for Burgers’ equation but to mention some of them especially which use variational approaches and finite elements. For example, Varoglu and Finn [182] proposed in isoparametric space-time finite element method for solving Burgers’ equation, utilizing the hyperbolic differential equation associated with Burgers’ equation. Ozis and Ozdes [151] applied a directional method to generate limiting form of the solution of Burgers’ equation. Ozis et al. [152] applied a simple finite element approach with linear elements to Burgers’ equation reduced by Hopf-Cole transformation. Aksan and Ozdes [9] have reduced Burgers’ equation to the system to the system of non-linear ordinary differential equations by discretization in time and solved each nonlinear ordinary differential equation by Galerkin method in each time step. As they claimed, for moderately small kinematics viscosity, their approach can provide high accuracy while using a small number of grid points (that is, $N=5$) and this makes the approach very economical computational wise. In the case where the kinematics viscosity is small enough, That is, the exact solution is not available and a discrepancy exists in the literature, their results clarify the behavior of the solution for small times, That is, $T = t_{max} \leq 0.15$ In this chapter, again, the reduced Burgers’ equation by the method of discretization in time has been solved by Haar wavelet method in each time step. It is, now, aimed that establishing the existence of the solution in this case, it can be made some improvement on the values of kinematics viscosity and maximum time values.
We introduce a Haar wavelet method for solving one-dimensional Burgers’ equations, which will exhibit several advantageous features:

i) Very high accuracy fast transformation and possibility of implementation of fast algorithms compared with other known methods.

ii) The simplicity and small computation costs, resulting from the sparsity of the transform matrices and the small number of significant wavelet coefficients.

iii) The method is also very convenient for solving the boundary value problems, since the boundary conditions are taken care of automatically.

Beginning from 1980’s, wavelets have been used for solution of partial differential equations (PDE). The good features of this approach are possibility to detect singularities, irregular structure and transient phenomena exhibited by the analyzed equations. Evidently all attempts to simplify the wavelet solutions for PDE are welcome. One possibility for this is to make use of the Haar wavelet family.

Haar wavelets (which are Daubechies of order 1) consists of piecewise constant functions and are therefore the simplest orthonormal wavelets with a compact support. 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. There are two possibilities for getting out of this situation. One way is to regularize the Haar wavelets with interpolating splines (for example, B-splines or Deslaurier-Dabuc interpolating wavelets). This approach has been applied by Cattani [34], but the regularization process considerably complicates the solution and the main advantage of the Haar wavelets—the simplicity gets to some extent lost. The other way is to make use of the integral
method, which was proposed by Chen and Hsiao [37]. Computational complexity of the discretization methods appears typically with an increasing number of integration points (mesh). Lepik [[120],[121],[122]] had solved higher order as well as nonlinear ODE by using Haar wavelet method. There are discussions by other researchers [[77],[77],[80],[92],[93],[108]].

9.2 Solution of one-dimensional Burgers’ equation

Hopf and Cole discovered the Hopf-Cole transformation independently around 1950. It changes Burgers’ equation (9.1) into the heat equation $\phi_t = v\phi_{xx}$. The solution of grid representation of heat equation involves matrix functions. Approximating the exponential matrix functions in this recurrence relation develops numerical techniques.

Consider the Burgers’ equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = v \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1. \quad (9.4)$$

With the initial condition

$$u(x, 0) = \sin \pi x \quad (9.5)$$

and the homogeneous boundary conditions

$$u(0, t) = 0,$$

$$u(1, t) = 0, t > 0$$

By the Hopf-Cole transformation
The Burgers’ equation transforms to the linear heat equation

\[ \frac{\partial \theta}{\partial t} = v \frac{\partial^2 \theta}{\partial x^2} \]  

(9.7)

However, by the Hopf-Cole transformations, initial condition (9.5) and the above mentioned homogeneous boundary condition transform following conditions (9.8) and (9.9), respectively.

\[ \theta_0(x) = \theta(x, 0) = e^{-(2v\pi)^{-1}[1-\cos(\pi x)]}, \quad 0 < x < 1. \]  

(9.8)

\[ \theta_x(0, t) = \theta_x(1, t) = 0, \quad t > 0 \]  

(9.9)

where if \( \theta = \theta(x, t) \) is any solution of heat equation (9.7) is a solution of Burgers’ equation (9.4) with the initial condition (9.5) and the above boundary conditions.

Hence, using the method of separation of variables the Fourier series solution to the above problem by equations (9.7)-(9.9) can be obtained easily as

\[ \theta(x, t) = a_0 + \sum_{n=1}^{\infty} a_n e^{-n^2 \pi^2 vt} \cos(n \pi x) \]  

(9.10)

where \( a_0 \) and \( a_n(n = 1, 2, \ldots) \) are Fourier coefficients and they are evaluated in the usual manner

\[ a_0 = \int_0^1 e^{-(2v\pi)^{-1}[1-\cos(\pi x)]} \, dx \]  

(9.11)
\[ a_n = 2 \int_0^1 e^{-(2\pi^2)\frac{1}{1-\cos(\pi x)}} \cos(n \pi x) dx, \quad (n = 1, 2, \ldots) \] (9.12)

Thus, using the Hopf-Cole transformation given by equation (9.6), the (exact) Fourier solution to the problem given by equations (9.4),(9.5) and the above mentioned boundary conditions are obtained as

\[ u(x, t) = \frac{2\pi v}{a_0} \sum_{n=1}^{\infty} \frac{a_n e^{-n^2\pi^2 v t n \sin(n \pi x)}}{a_n e^{-n^2\pi^2 v t \cos(n \pi x)}} \] (9.13)

### 9.3 Method of solution

We will consider the one dimensional Burgers’ equation (9.4) with the initial condition (9.5) and the boundary conditions. Let us divide the interval (0,1] into \( N \) equal parts of length \( \Delta t = (0, 1]/N \) and denote \( t_s = (s - 1)\Delta t \). We assume that can be expanded in terms of Haar wavelets as formula

\[ \dot{u}''(x, t) = \sum_{n=0}^{m-1} c_s(n) h_n(x) = c^{T}_{(m)} h_{(m)}(x) \] (9.14)

where . and ' means differentiation with respect to \( t \) and \( x \) respectively, the row vector \( c^{T}_{(m)} \) is constant in the subinterval \( t \in (t_s, t_{s+1}] \).

Integrating formula (9.14) with respect to \( t \) from \( t_s \) to \( t \) and twice with respect to \( x \) from 0 to \( x \), we obtain

\[ u''(x, t) = (t - t_s) c^{T}_{(m)} h_{(m)}(x) + u''(x, t_s), \] (9.15)

\[ u'(x, t) = (t - t_s) c^{T}_{(m)} P_{(m)} h_{(m)}(x) + u'(x, t_s) - u'(0, t_s) + u'(0, t) \] (9.16)
\[ u(x, t) = (t - t_s)c_{(m)}^T P_m^2 h_m(x) + u(x, t) - u(0, t_s) + x[\dot{u}'(0, t) - \dot{u}'(0, t_s)] + \dot{u}(0, t) \]  

(9.17)

\[ \ddot{u}(x, t) = c_{(m)}^T P_m^2 h_m(x) + x\ddot{u}'(0, t) + \dot{u}(0, t) \]  

(9.18)

Using the boundary conditions, we obtain

\[ u(0, t) = u(0, t_s) = 0 \]
\[ \dot{u}(0, t) = 0 \]
\[ u(1, t) = \dot{u}(1, t) = 0 \]

give

\[ \dot{u}(0, t) - \dot{u}(0, t_s) = -(t - t_s)c_{(m)}^T P_m h_m(x) \]  

(9.19)

\[ \ddot{u}'(0, t) = -c_{(m)}^T P_m^2 f \]  

(9.20)

where the vector \( f \) is defined as

\[ f = \begin{bmatrix} 1, 0, \cdots, 0 \end{bmatrix}^{T} \]

\((m-1)\text{elements})\]

substituting Eq.(9.19) and (9.20) into the equations (9.15)-(9.18), and discretizing the results by assuming \( x \to x_l, t \to t_{s+1} \) we obtain

\[ u''(x_l, t_{s+1}) = (t_{s+1} - t_s)c_{(m)}^T h_m(x_l) + u''(x_l, t_s) \]  

(9.21)
\[ u'(x_l, t_{s+1}) = (t_{s+1} - t_s) w^T_m P_m h_m(x_l) + u'(x_l, t_s) - (t_{s+1} - t_s) w^T_m P_m f \]

\[ u(x_l, t_{s+1}) = (t_{s+1} - t_s) w^T_m P_m h_m(x_l) + u(x_l, t_s)[(t_{s+1} - t_s) c^T_m P_m f] \]

\[ \dot{u}(x_l, t_{s+1}) = w^T_m P_m^2 h_m(x_l) + x_l[-c^T_m P_m f] \]

There are several possibilities for treating the nonlinearity in equations (9.4) and (9.5). In the following scheme

\[ \dot{u}(x_l, t_{s+1}) = -u(x_l, t_s)u'(x_l, t_{s+1}) + vu''(x_l, t_{s+1}) \]

which leads us from the time layer \( t_s \) to \( t_{s+1} \) is used.

Substituting equations (9.21)-(9.24) into the equation (9.25), we gain

\[ c^T_m P_m^2 h_m(x_l) + x_l[-c^T_m P_m f] = -u(x_l, t_s)u'(x_l, t_s) - vu''(x_l, t_s) \]

From Eq.(9.26) the wavelet coefficients \( C^T_m \) can be successively calculated. This process is started with

\[ u(x_l, t_l) = \sin[\pi x(l)] u'(x_l, t_l) = \pi \cos[\pi x(l)] u''(x_l, t_l) = -\pi^2 \sin[\pi x(l)] \]

For estimating the efficiency of the solution it is expedient to calculate the
maximum value of the gradient $|\partial u/\partial x|$ at $x = 0.5$.

According to the analytic solution, for $v = (400\pi)^{-1}$ the theoretical maximum is $|\partial u/\partial x| = 304.0$ and takes place at $t_{max} = 0.51$. Computer simulation was carried out for $v = (400\pi)^{-1}$. Results for $J = 5(m = 32$ collocation points$)$ and $J = 6(m = 64$ collocation points$), \Delta t = 0.001, t = 0.35$ are plotted in figure 9.2. It follows from this Figure that in spite of the number of collocation points the solution describes quite well the saw-tooth effect. With increasing $t$, $u$ oscillates near the point $x = 0.5$ appear in Figure 9.2. For getting more results $m$ must be increased.

In this case $Max|\partial u/\partial x| = 400.4$. If we calculate the angle $\delta = \arctan(Max|\partial u/\partial x|)$, then our solution gives $\delta = 89.86$, while in the case of the analytic solution we have $\delta = 89.81$. Some results of computation are presented in Table.9.5. Comparison with these algorithms shows that the Haar wavelet method is competitive and efficient. The advantages of our method are its simplicity and speed of convergence [172].

**Problem: 2**

As another example, we consider the Burgers’ equation with initial condition

$$u(x, 0) = \sin(2\pi x), \quad 0 < x < 1$$

and boundary conditions

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

This example is usually used for simulating the shock formation. Using the Hopf-Cole transformation, we have

$$a_0 = \int_0^1 e^{(-(4\pi v)^{-1}(1-\cos(2\pi x)))} dx$$

and
\[
an_n = 2 \int_0^1 e^{-(4\pi v)^{-1}(1-\cos(2\pi x))} dx \cos(n\pi x), \quad n=1,2,3,...
\]

From Equation (9.26) the wavelet coefficients \( C_{(m)}^T \) can be successively calculated. This process is started with

\[
\begin{align*}
\left[ u(x_l, t_1) = \sin[2\pi x(l)] \right. \\
\left[ u'(x_l, t_1) = 2\pi \cos[2\pi x(l)] \right. \\
\left[ u''(x_l, t_1) = -4\pi^2 \sin[\pi x(l)] \right.
\end{align*}
\]

**Problem: 3**

We consider Burger’s equation (9.4) with initial condition

\[
u(x, 0) = \sin(3\pi x), \quad 0 < x < 1
\]

and boundary conditions

\[
u(0, t) = u(1, t) = 0, \quad t > 0
\]

Using the Hope-Cole transformation, we have

\[
a_0 = \int_0^1 e^{-(6\pi v)^{-1}(1 - \cos(3\pi x))} dx
\]

and

\[
a_n = 2 \int_0^1 e^{-(6\pi v)^{-1}(1 - \cos(3\pi x))} dx \cos(n\pi x), \quad n=1,2,3,...
\]

From Equation (9.26) the wavelet coefficients \( C_{(m)}^T \) can be successively calculated. This process is started with

\[
\begin{align*}
\left[ u(x_l, t_1) = \sin[3\pi x(l)] \right. \\
\left[ u'(x_l, t_1) = 3\pi \cos[3\pi x(l)] \right. \\
\left[ u''(x_l, t_1) = -9\pi^2 \sin[3\pi x(l)] \right.
\end{align*}
\]
Problem: 4  As another example, we consider the Burgers’ equation (9.4) with initial condition \( u(x,0) = 4x(1-x), \ 0 < x < 1 \).

Similarly, the exact solution to the problem 4 is obtained as equation (9.13). we note that where \( a_0 \) and \( a_n (n=1,2,3...) \).

Fourier coefficients are

\[
\begin{align*}
a_0 &= \int_0^1 e^{-x^2(3v)^{-1}(3-2x)} dx \\
a_n &= 2 \int_0^1 e^{-x^2(3v)^{-1}(3-2x)} \cos(n\pi x) dx \quad (n= 1,2,3...) 
\end{align*}
\]

Table 9.1: Comparison of the exact solution and the Haar solution of Burgers’ equation with \( v = 1, \Delta t = 0.000001, h = 0.0125 & m = 16 \) at different time steps

<table>
<thead>
<tr>
<th>x</th>
<th>t_{\text{max}}</th>
<th>\text{Exact}(u)</th>
<th>\text{Haar}(u)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.1</td>
<td>0.26148</td>
<td>0.26147</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.16148</td>
<td>0.16147</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.09947</td>
<td>0.09942</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.06108</td>
<td>0.06106</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>0.38342</td>
<td>0.38344</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.23406</td>
<td>0.23400</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.14289</td>
<td>0.14288</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.08723</td>
<td>0.08724</td>
</tr>
<tr>
<td>0.75</td>
<td>0.1</td>
<td>0.28157</td>
<td>0.28153</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>0.16974</td>
<td>0.16972</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.10266</td>
<td>0.10265</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.06229</td>
<td>0.06229</td>
</tr>
</tbody>
</table>

The estimations showed that the function \( v(t) \) increases monotonically; therefore for error estimate is taken \( v(t_{\text{max}}) \). It follows from Table 9.4 that already in the cases \( m = 16, \ m = 32 \) and \( m = 64 \) we get the results pertaining to our algorithm is closer to analytic solution. The error is negligible. The time step was taken as \( \Delta t = 0.005 \) or \( \Delta t = 0.001 \) further decrease in \( \Delta t \) gave no considerable effect. If the accuracy of these results is insufficient, more precise results could be obtained by the segmentation method proposed in [120].
Figure 9.1: Solution at different times for $v=1.0$, $h=0.025$, $k=0.0001$.

9.4 Features

It has been well demonstrated that in applying the nice properties of Haar wavelets, the partial differential equation can be solved conveniently by using
Figure 9.2: Solutions of Burgers’ equation for $v = (400\pi)^{-1}$, $\Delta t = 0.001$ (dashed line denotes the solution for $t=0$)

Haar wavelet method systematically. The method with far less degrees of freedom and with smaller CPU time provides better solutions than classical ones. The accuracy and effectiveness of the method are analyzed; the results obtained are
Table 9.2: Comparison of results at different positions and times for \( v = 1, \Delta x = 0.0125(N = 80), \Delta t = 1.0E - 05(m = 16) \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t )</th>
<th>HC [75]</th>
<th>RHC [75]</th>
<th>RPA [74]</th>
<th>Haar(u)</th>
<th>Exact(u)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.1</td>
<td>0.252942</td>
<td>0.264126</td>
<td>0.252875</td>
<td>0.253624</td>
<td>0.253637</td>
</tr>
<tr>
<td>0.15</td>
<td>0.156059</td>
<td>0.165833</td>
<td>0.155447</td>
<td>0.156600</td>
<td>0.156601</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>0.095889</td>
<td>0.101617</td>
<td>0.094289</td>
<td>0.096447</td>
<td>0.096440</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.056174</td>
<td>0.059113</td>
<td>0.055674</td>
<td>0.059223</td>
<td>0.059217</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>0.376474</td>
<td>0.393354</td>
<td>0.373085</td>
<td>0.371552</td>
<td>0.371579</td>
</tr>
<tr>
<td>0.15</td>
<td>0.235875</td>
<td>0.251788</td>
<td>0.228940</td>
<td>0.226824</td>
<td>0.226825</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>0.153645</td>
<td>0.163931</td>
<td>0.142127</td>
<td>0.138475</td>
<td>0.138473</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.112810</td>
<td>0.120967</td>
<td>0.091944</td>
<td>0.084535</td>
<td>0.084537</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.1</td>
<td>0.271517</td>
<td>0.285579</td>
<td>0.272368</td>
<td>0.272582</td>
<td>0.272580</td>
</tr>
<tr>
<td>0.15</td>
<td>0.162739</td>
<td>0.176957</td>
<td>0.163628</td>
<td>0.164365</td>
<td>0.164369</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>0.098431</td>
<td>0.111020</td>
<td>0.098656</td>
<td>0.099432</td>
<td>0.099434</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.057394</td>
<td>0.068659</td>
<td>0.059343</td>
<td>0.060347</td>
<td>0.060346</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.3: Comparison of results at different positions and times for \( v = 1, \Delta x = 0.0125(N = 80), \Delta t = 1.0E - 05(m = 16) \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t )</th>
<th>HC [75]</th>
<th>RHC [75]</th>
<th>RPA [74]</th>
<th>Haar(u)</th>
<th>Exact(u)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.308860</td>
<td>0.317062</td>
<td>0.308776</td>
<td>0.308894</td>
<td>0.308893</td>
</tr>
<tr>
<td>0.6</td>
<td>0.240703</td>
<td>0.248472</td>
<td>0.240654</td>
<td>0.240735</td>
<td>0.240739</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.195693</td>
<td>0.202953</td>
<td>0.195579</td>
<td>0.195678</td>
<td>0.195676</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.162561</td>
<td>0.169527</td>
<td>0.162513</td>
<td>0.162563</td>
<td>0.162564</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.4</td>
<td>0.569602</td>
<td>0.583408</td>
<td>0.569527</td>
<td>0.569633</td>
<td>0.569632</td>
</tr>
<tr>
<td>0.6</td>
<td>0.447123</td>
<td>0.461714</td>
<td>0.447117</td>
<td>0.447204</td>
<td>0.447205</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.359152</td>
<td>0.373800</td>
<td>0.359161</td>
<td>0.359235</td>
<td>0.359236</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.291961</td>
<td>0.306184</td>
<td>0.291843</td>
<td>0.291918</td>
<td>0.291916</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.4</td>
<td>0.625460</td>
<td>0.638847</td>
<td>0.625341</td>
<td>0.625436</td>
<td>0.625437</td>
</tr>
<tr>
<td>0.6</td>
<td>0.487337</td>
<td>0.506429</td>
<td>0.487089</td>
<td>0.487215</td>
<td>0.487211</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.374067</td>
<td>0.393565</td>
<td>0.373827</td>
<td>0.373925</td>
<td>0.373923</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.287525</td>
<td>0.305862</td>
<td>0.287396</td>
<td>0.287472</td>
<td>0.287473</td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>0.029532</td>
<td>0.034484</td>
<td>0.029726</td>
<td>0.029774</td>
<td>0.029771</td>
<td></td>
</tr>
</tbody>
</table>

compared with the results of other authors (using classical numerical techniques) and with the exact solution, evaluating the error. The benefits of Haar wavelet
Table 9.4: Comparison of the exact solution and the Haar solution of Burgers’ equation for $t = 0.85$

<table>
<thead>
<tr>
<th>x</th>
<th>$m = 16$</th>
<th>$m = 32$</th>
<th>$m = 64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.44082</td>
<td>0.44080</td>
<td>0.44082</td>
</tr>
<tr>
<td>0.2</td>
<td>0.46846</td>
<td>0.46845</td>
<td>0.46846</td>
</tr>
<tr>
<td>0.3</td>
<td>0.49658</td>
<td>0.49656</td>
<td>0.49657</td>
</tr>
<tr>
<td>0.4</td>
<td>0.52510</td>
<td>0.52509</td>
<td>0.52510</td>
</tr>
<tr>
<td>0.5</td>
<td>0.55396</td>
<td>0.55394</td>
<td>0.55397</td>
</tr>
<tr>
<td>0.6</td>
<td>0.58320</td>
<td>0.58319</td>
<td>0.58320</td>
</tr>
<tr>
<td>0.7</td>
<td>0.61208</td>
<td>0.61204</td>
<td>0.61207</td>
</tr>
<tr>
<td>0.8</td>
<td>0.64106</td>
<td>0.64107</td>
<td>0.64106</td>
</tr>
</tbody>
</table>

Table 9.5: Some comparison results of the Haar solution

<table>
<thead>
<tr>
<th>$m$</th>
<th>$\Delta t$</th>
<th>$v(t_{max})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.005</td>
<td>0.051</td>
</tr>
<tr>
<td>16</td>
<td>0.001</td>
<td>0.038</td>
</tr>
<tr>
<td>32</td>
<td>0.005</td>
<td>0.018</td>
</tr>
<tr>
<td>32</td>
<td>0.001</td>
<td>0.0090</td>
</tr>
<tr>
<td>64</td>
<td>0.005</td>
<td>0.0096</td>
</tr>
<tr>
<td>64</td>
<td>0.001</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

approach are sparse matrices of representation, fast transformation and designing of fast algorithms and the merits of the method lie in its simplicity, computational economy and easy implementation.

For better solutions, instead of increasing the value of $m$ one can use other type of wavelets such as Mexican wavelets, spline wavelets etc. Use of spline wavelets for solving other type of Burgers’ equation is presently ongoing.