COMBINED COMPACT FINITE DIFFERENCE TREATMENT
OF BURGERS’ EQUATION

M.H. Mousa¹§, A.A. Abadeer², M.M. Abbas³
¹Department of Mathematics
Faculty of Science
Mansoura University
Mansoura, 35516, EGYPT
²,³Department of Mathematics
Faculty of Science at Damietta
Mansoura University
New Damietta, 34517, EGYPT

Abstract: In this paper, numerical solution of nonlinear Burgers’ equation is obtained by using a combined compact finite difference method, our method is simple and avoids restrictions and complications encountered in published methods. The computed results given here are compared with the exact solution and pervious works to show the efficiency of the method.

AMS Subject Classification: 65M06, 97M50
Key Words: Burgers’ equation, combined compact finite differences, compact finite differences

1. Introduction

Recently, there is a great interest in solving the nonlinear Burgers’ equation. Various numerical methods were used for simulating solutions by solving the linearity problem using the Hopf–Cole transformation to reduce the Burger equation to a linear heat equation, and solving the heat equation by a Galerkin quadratic B-spline finite element method [1], by modified Adomian’s decomposition method [2], by a restrictive Taylor approximation [3], by a restrictive
Padé approximation [4], or by finite element approach [5]. Other research done without linearization the Burger equation such as the matched asymptotic expansion [6], discretization in time with Galerkin method [7], the modified Adomian’s decomposition method [8] and [9], using a finite difference schemes [10] and [11], the homotopy analysis method [12], a network simulation [13], or the compact finite differences [14].

Here we consider the following Burgers’ equations in the form

\[ \psi_t + \psi\psi_x = v\psi_{xx}, \quad t \in [0, \infty), \; x \in [0, 1], \]  
\[ \psi(x, 0) = f(x) \quad \forall 0 \leq x \leq 1, \]  
\[ \psi(0, t) = \psi(1, t) = 0, \quad \forall t \in (0, \infty). \]

To solve the equation, we use the Hopf–Cole transformation [5] \( \psi = -2v\frac{u_x}{u} \) to convert the problem into the following heat equation problem.

\[ u_t = vu_{xx}, \]  
with initial and boundary conditions

\[ u(x, 0) = \exp \left( \int_0^x -\frac{1}{2v} f(x) \, dx \right), \]  
\[ u_x(0, t) = u_x(1, t) = 0. \]

The accuracy and reliability of the present method is verified using exact solution with more digital than given in previous papers.

In this paper, a sixth order compact (three-point stencil) scheme in space variables is employed to solve the Burgers’ equation(1), with initial condition (2) and boundary conditions (3), is proposed. a low-storage version of the Runge-Kutta method of order two is used for time integration. The rest of the paper is organized as follows: In Section 2, we will specify the current scheme in this paper. Handling the boundary conditions will be considered in Section 3, followed by numerical experiments and results analysis in Section 4. Finally, concluding remarks are presented in Section 5.

2. The Combined Compact Finite Difference Scheme

Let the dependent variable \( u(x, t) \) be defined on the intervals \( x \in [0, 1], \; t \in [0, \infty) \). For simplicity uniform meshes in space and time are considered, consisting of \( M \) points \( 0 = x_1, x_2, ..., x_{i-1}, x_i, x_{i+1}, ..., x_M = 1 \) and \( N \) points
0 = t_1, t_2, ..., t_{n-1}, t_n, t_{n+1}, ..., t_N. The mesh sizes are denoted by \( h = x_{i+1} - x_i \) and \( k = t_{n+1} - t_n \). Let the dependent variable \( u(x, t) \) at any grid point \((x_i, t_n)\) and two neighboring points \((x_{i-1}, t_n)\) and \((x_{i+1}, t_n)\) be given by \( u^n_i, u^n_{i-1} \) and \( u^n_{i+1} \) respectively, its first and second derivatives at the grid point \((x_i, t_n)\) be given by \( F^n_i = (u_x)_i^n, G^n_i = (u_{xx})_i^n \).

### 2.1. Spatial Discretization

In the interior points spatial derivatives are evaluated by the combined compact finite difference scheme as given in Chu et al. [15]. Below we summarize the method used in this paper; more details can be found in Chu et al. [15].

The first and second derivatives can be computed at internal nodes as follows:

\[
\alpha_1 \left( F^n_{i-1} + F^n_{i+1} \right) + F^n_i + h\beta_1 \left( G^n_{i+1} - G^n_{i-1} \right) = \frac{a_1}{h} \left( u^n_{i+1} - u^n_{i-1} \right) \quad (7)
\]

\[
G^n_i + \alpha_2 \left( G^n_{i-1} + G^n_{i+1} \right) + \frac{\beta_2}{h} \left( F^n_{i+1} - F^n_{i-1} \right) = \frac{a_2}{h^2} \left( u^n_{i+1} - 2u^n_i + u^n_{i-1} \right) \quad (8)
\]

where \( \alpha_1, \beta_1, a_1, \alpha_2, \beta_2 \) and \( a_2 \) are constants. This gives rise to a family of twin-tridiagonal system of algebraic equations. In [15] the sixth-order twin-tridiagonal system is obtained by using local Hermitian polynomials but we can obtain the same scheme by using Taylor expansion and matching the Taylor series coefficients of various orders. The following choice of parameters

\[
\begin{align*}
\alpha_1 &= \frac{7}{16}, \quad \beta_1 = -\frac{1}{16}, \quad a_1 = \frac{15}{16}, \\
\alpha_2 &= -\frac{1}{8}, \quad \beta_2 = \frac{9}{8}, \quad a_2 = 3
\end{align*}
\]

will yield

\[
\frac{7}{16} F^n_{i-1} + F^n_i + \frac{7}{16} F^n_{i+1} + \frac{h}{16} G^n_{i+1} - \frac{h}{16} G^n_{i-1} = \frac{15}{16h} \left( u^n_{i+1} - u^n_{i-1} \right) \quad (9)
\]

\[
-\frac{9}{8h} F^n_{i-1} + \frac{9}{8h} F^n_{i+1} - \frac{1}{8} G^n_{i+1} + G^n_i - \frac{1}{8} G^n_{i+1} = \frac{3}{h^2} \left( u^n_{i+1} - 2u^n_i + u^n_{i-1} \right) \quad (10)
\]

The technique of solving the twin-tridiagonal system is illustrated in appendix.

### 2.2. Temporal Discretization

In the current work, the equations are integrated in time using low storage Runge-Kutta scheme (ORK25-6) [16]. Assuming that the governing equation is \( \frac{\partial u}{\partial t} = \phi(t, u, F, G) \), where \( \phi \) stands for a differentiable function in space,
Table 1: The coefficients for ORK25-6 scheme

<table>
<thead>
<tr>
<th>stage</th>
<th>$\rho_j$</th>
<th>$\sigma_j$</th>
<th>$c_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>-1.0</td>
<td>0.83204</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>-1.55798</td>
<td>0.6</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>-1.0</td>
<td>0.35394</td>
<td>0.8</td>
</tr>
<tr>
<td>5</td>
<td>-0.45031</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

The ORK25-6 scheme integrates from time $t_n$ to $t_{n+1} = t_n + k$ through the operations

\[
\begin{align*}
v_1 &= u(t_n, x_i) = u^n_i, \\
T_1 &= t_n, \\
w_1 &= 0, \\
T_j &= t_n + c_j k \\
w_{j+1} &= \rho_j w_j + k \phi(T_j, v_j, F_j, G_j) \\
v_{j+1} &= v_j + \sigma_j w_{j+1} \\
u(t_{n+1}, x_i) &= u_i^{n+1} = v_{s+1}
\end{align*}
\]

where $F_j = F(T_j, v_j)$, $G_j = G(T_j, v_j)$, and the coefficients for ORK25-6 scheme are given in Table 1.

### 3. Boundary Treatment

For the nodes at the boundary, we derive the sixth-order second derivatives formulae at boundary points 1, $M$ without changing the twin-tridiagonal shape, respectively, as follows:

\[
G^n_1 + \beta G^n_2 + \delta F^n_2 = \frac{1}{h^2} (a u^n_1 + b u^n_2 + c u^n_3 + d u^n_4 + e u^n_5)
\]

(11)

The relations between the coefficients $a, b, c, d, e, \beta$ and $\delta$ are obtained by matching the Taylor series coefficients of various orders and the first unmatched coefficients determine the formal truncation error. We have the sixth-order forward combined finite difference

\[
G^n_1 + \frac{37}{3} G^n_2 + \frac{35}{9h} F^n_2 = \frac{1}{h^2} \left( \frac{53}{4} u^n_1 - \frac{1753}{54} u^n_2 + \frac{43}{2} u^n_3 - \frac{5}{2} u^n_4 + \frac{23}{108} u^n_5 \right)
\]

(12)

as $a = \frac{53}{4h^2}, b = -\frac{1753}{54h^2}, c = \frac{43}{2h^2}, d = -\frac{5}{2h^2}, e = \frac{23}{108h^2}, \beta = \frac{37}{3}, \delta = \frac{35}{9h}$
In the same way we can calculate the sixth-order backward combined finite difference and the formula becomes

\[
G^n_M + \frac{37}{3} G^n_{M-1} - \frac{35}{9h} F^n_{M-1} = \frac{1}{h^2} \left( \frac{53}{4} u^n_M - \frac{1753}{54} u^n_{M-1} + \frac{43}{2} u^n_{M-2} - \frac{5}{2} u^n_{M-3} + \frac{23}{108} u^n_{M-4} \right) \tag{13}
\]

Similarly the sixth-order first derivatives formulas at the boundary points 1, M

\[
F^n_1 + \frac{2}{3} F^n_2 - 2hG^n_2 = \frac{1}{h} \left( -\frac{49}{12} u^n_1 + \frac{61}{9} u^n_2 - 3u^n_3 + \frac{1}{3} u^n_4 - \frac{1}{36} u^n_5 \right) \tag{14}
\]

\[
F^n_M + \frac{2}{3} F^n_{M-1} + 2hG^n_{M-1} = \frac{1}{h} \left( \frac{49}{12} u^n_M - \frac{61}{9} u^n_{M-1} + 3u^n_{M-2} - \frac{1}{3} u^n_{M-3} + \frac{1}{36} u^n_{M-4} \right) \tag{15}
\]

But we only use here the sixth-order for the second derivatives at the boundary and for the first derivative we replace it by the boundary conditions to complete the system.

4. Numerical Experiments and Results Analysis

In this section, we present the numerical results of the new method on several problems compared with the exact solution and previous works.

Example 1: Here we will solve the one-dimensional Burgers’ equation (1) with initial condition

\[
\psi(x,0) = \sin (\pi x), \tag{16}
\]

By using Hopf-Cole transformation, the initial condition transform to

\[
u(x,0) = \exp \left( -\frac{1}{2\pi v} (1 - \cos (\pi x)) \right) \tag{17}
\]

The exact solution [3] is given by

\[
\psi(x,t) = 2\pi v \sum_{n=1}^{\infty} \frac{a_n n \exp \left( -n^2 \pi^2 vt \right) \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \exp \left( -n^2 \pi^2 vt \right) \cos(n\pi x)} \tag{18}
\]
with the Fourier coefficients

\begin{align*}
a_0 &= \int_0^1 \exp \left( - (2\pi v)^{-1} (1 - \cos (\pi x)) \right) dx \\
a_n &= 2 \int_0^1 \exp \left( - (2\pi v)^{-1} (1 - \cos (\pi x)) \right) \cos (n\pi x) dx, \quad n = 1, 2, 3, \ldots \tag{19}
\end{align*}

We will use the sixth order central combined finite difference equations (9) and (10) for the internal space point, the sixth order one sided combined approximation for the second derivative (12) and (13) on the each boundary points. To complete the system, we also will use the boundary conditions (6) as approximation of first derivative at left and the right side as follows:-

\begin{align*}
F_1^n &= 0 \quad \tag{20} \\
F_M^n &= 0 \quad \tag{21}
\end{align*}

In the process, we use the approximation of the second derivative for the solution of the heat equation but only the value of the first derivative approximation at the first stage of ORK25=6 will be used to evaluate the variable \( \psi = -2v \frac{ux}{u} \).

- For \( v = 1, h = 0.0125, k = 0.00001 \), the comparison of the current results \textbf{HCCFD6} with the exact solution as well the results of [3],[13] in Table 2 showed that the presented results are very accurate in intervals \( x \in [0,1] \) and \( t \in [0,0.25] \).

- For \( v = 0.1, h = 0.0125, k = 0.0001 \), the comparison of the current results with the exact solution as well the results of [3],[13] in Table 3 showed that the presented results are very accurate in intervals \( x \in [0,1] \) and \( t \in [0,3] \).

- For \( v = 0.01, h = 0.01, k = 0.001 \), the comparison of the current results with the exact solution as well the results of [17] in Table 4 showed that the presented results are very accurate in intervals \( x \in [0,1] \) and \( t \in [0,4] \).

- For \( v = 1, h = 0.1, k = 0.0001 \), the comparison of the current results with the exact solution as well the results of [7] in Table 5 showed that the presented results are very accurate in intervals \( x \in [0,1] \) and \( t = 0.1 \).
Example 2: Here we will solve the one-dimensional Burgers’ equation (1) with initial condition

\[ \psi(x, 0) = 4x (1 - x) , \]  

\textit{Table 2}: Comparison of results for Problem 1 at different times for \( v = 1, h = 0.0125, k = 0.00001 \)

<table>
<thead>
<tr>
<th>x</th>
<th>t</th>
<th>Exact</th>
<th>HCCFD6</th>
<th>RHC[3]</th>
<th>[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.308894227876420</td>
<td>0.30889422800835</td>
<td>0.317062</td>
<td>0.308902</td>
</tr>
<tr>
<td>0.6</td>
<td>0.240739023290827</td>
<td>0.24073902337374</td>
<td>0.248472</td>
<td>0.240750</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.195675570103439</td>
<td>0.19567557015980</td>
<td>0.202953</td>
<td>0.202953</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.162564857110670</td>
<td>0.16256485715196</td>
<td>0.169527</td>
<td>0.169527</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x</th>
<th>t</th>
<th>Exact</th>
<th>HCCFD6</th>
<th>RHC[3]</th>
<th>[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.308894227876420</td>
<td>0.30889422800835</td>
<td>0.317062</td>
<td>0.308902</td>
</tr>
<tr>
<td>0.6</td>
<td>0.240739023290827</td>
<td>0.24073902337374</td>
<td>0.248472</td>
<td>0.240750</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.195675570103439</td>
<td>0.19567557015980</td>
<td>0.202953</td>
<td>0.202953</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.162564857110670</td>
<td>0.16256485715196</td>
<td>0.169527</td>
<td>0.169527</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Comparison of results for Problem 1 at different times for \( v = 0.1, h = 0.0125, k = 0.0001 \)

<table>
<thead>
<tr>
<th>x</th>
<th>t</th>
<th>Exact</th>
<th>HCCFD6</th>
<th>RHC[3]</th>
<th>[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.625437896424913</td>
<td>0.62543789645580</td>
<td>0.638847</td>
<td>0.625426</td>
</tr>
<tr>
<td>0.6</td>
<td>0.48721497483946</td>
<td>0.48721497484945</td>
<td>0.506429</td>
<td>0.487279</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.373921753209456</td>
<td>0.37392175327646</td>
<td>0.393565</td>
<td>0.374005</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.287474405916976</td>
<td>0.28747440597921</td>
<td>0.305862</td>
<td>0.287555</td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>0.02977212685877</td>
<td>0.02977212686875</td>
<td>0.034848</td>
<td>0.029786</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>t</td>
<td>Exact</td>
<td>HCCFD6</td>
<td>[7]</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>0.121143531499446</td>
<td>0.12114354280213</td>
<td>0.12079</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.04296377803806</td>
<td>0.04296377803806</td>
<td>0.04300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.02310423297530</td>
<td>0.02310423297530</td>
<td>0.02324</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>0.36027105847179</td>
<td>0.36027105847179</td>
<td>0.36113</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.12883989287318</td>
<td>0.12883989287318</td>
<td>0.12877</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.06930829113796</td>
<td>0.06930829113796</td>
<td>0.06935</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.58869578627457</td>
<td>0.58869578627457</td>
<td>0.59559</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.21455805679446</td>
<td>0.21455805679446</td>
<td>0.21468</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.11549475744843</td>
<td>0.11549475744843</td>
<td>0.11550</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0.5</td>
<td>0.793493405966195</td>
<td>0.793493405966195</td>
<td>0.81257</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.29999776882587</td>
<td>0.29999776882587</td>
<td>0.30075</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.1612146556121</td>
<td>0.1612146556121</td>
<td>0.16125</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.5</td>
<td>0.9381095337230</td>
<td>0.9381095337230</td>
<td>0.97184</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.37327762855064</td>
<td>0.37327762855064</td>
<td>0.37452</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.16605872299879</td>
<td>0.16605872299879</td>
<td>0.16515</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison of results for Problem 1 at different times for $v = 0.01, h = 0.01, k = 0.001$

<table>
<thead>
<tr>
<th>x</th>
<th>exact</th>
<th>HCCFD6</th>
<th>[7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.109538151270509</td>
<td>0.10954222226587</td>
<td>0.10958</td>
</tr>
<tr>
<td>0.2</td>
<td>0.209792148910037</td>
<td>0.20979568495887</td>
<td>0.20989</td>
</tr>
<tr>
<td>0.3</td>
<td>0.291896350825530</td>
<td>0.29190012701347</td>
<td>0.29199</td>
</tr>
<tr>
<td>0.4</td>
<td>0.347923912365551</td>
<td>0.34792707785181</td>
<td>0.34809</td>
</tr>
<tr>
<td>0.5</td>
<td>0.371577476146794</td>
<td>0.37157989686308</td>
<td>0.37173</td>
</tr>
<tr>
<td>0.6</td>
<td>0.359045579984961</td>
<td>0.35904713273930</td>
<td>0.35920</td>
</tr>
<tr>
<td>0.7</td>
<td>0.30990507631105</td>
<td>0.30990591035177</td>
<td>0.31003</td>
</tr>
<tr>
<td>0.8</td>
<td>0.227817406627376</td>
<td>0.2278178262166</td>
<td>0.22792</td>
</tr>
<tr>
<td>0.9</td>
<td>0.120686691089410</td>
<td>0.12068759038988</td>
<td>0.12071</td>
</tr>
</tbody>
</table>

Table 5: Comparison of results for Problem 1 for $v = 1, h = 0.01, k = 0.0001$ and $t = 0.1$

By using Hopf-Cole transformation, the initial condition transform to

$$u(x,0) = \exp \left( -x^2 (3v)^{-1} (3-2x) \right)$$

(23)

The exact solution [3] is also given by (12).
With the Fourier coefficients

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

(24)

The method was solved using the above technique with the same difference equations (9), (10), (12), (13), (20) and (21)

- For \( v = 1, h = 0.0125, k = 0.00001 \), the comparison of the current results with the exact solution as well the results of [3] in Table 6 showed that the presented results are very accurate in intervals \( x \in [0, 1] \) and \( t \in [0, 0.25] \).
- For \( v = 0.1, h = 0.0125, k = 0.0001 \), the comparison of the current results with the exact solution as well the results of [3] in Table 7 showed that the presented results are very accurate in intervals \( x \in [0, 1] \) and \( t \in [0, 3] \).
- For \( v = 0.01, h = 0.01, k = 0.001 \), the comparison of the current results with the exact solution as well the results of [17] in Table 8 showed that the presented results are very accurate in intervals \( x \in [0, 1] \) and \( t \in [0, 4] \).
<table>
<thead>
<tr>
<th>x</th>
<th>t</th>
<th>exact</th>
<th>HCCFD6</th>
<th>RHC[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>0.317522880346768</td>
<td>0.31752288093206</td>
<td>0.306529</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.246138455741545</td>
<td>0.24613845606064</td>
<td>0.236051</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.199555307716945</td>
<td>0.19955530791867</td>
<td>0.190181</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.165598631696975</td>
<td>0.16559863183902</td>
<td>0.166646</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4</td>
<td>0.584537259423137</td>
<td>0.58453725996466</td>
<td>0.565994</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.457976404556937</td>
<td>0.45797640497895</td>
<td>0.438926</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.367398193136396</td>
<td>0.36739819345298</td>
<td>0.348328</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.298343106946419</td>
<td>0.29834310719240</td>
<td>0.280038</td>
</tr>
<tr>
<td>0.75</td>
<td>0.4</td>
<td>0.645615507508048</td>
<td>0.64561550757305</td>
<td>0.626990</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.502675751374800</td>
<td>0.50267575164533</td>
<td>0.477908</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.385335518826973</td>
<td>0.38533551911836</td>
<td>0.360630</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.295856684503934</td>
<td>0.29585668475873</td>
<td>0.272623</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>0.03043964524096</td>
<td>0.03043964526907</td>
<td>0.024748</td>
</tr>
</tbody>
</table>

Table 7: Comparison of results for Problem 2 at different times for $v = 0.1, h = 0.0125, k = 0.0001$

<table>
<thead>
<tr>
<th>x</th>
<th>t</th>
<th>exact</th>
<th>HCCFD6</th>
<th>[17]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>0.128462159628175</td>
<td>0.12846218814552</td>
<td>0.12808</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.043813854221127</td>
<td>0.04381385673479</td>
<td>0.04388</td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.023345001323585</td>
<td>0.02334500195258</td>
<td>0.02351</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>0.378489128693062</td>
<td>0.37848913727667</td>
<td>0.37956</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.131345187649400</td>
<td>0.13134519372199</td>
<td>0.13129</td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.070027182347423</td>
<td>0.07002718412010</td>
<td>0.07009</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.609886125745511</td>
<td>0.60988615631258</td>
<td>0.61768</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.218588014961054</td>
<td>0.21858802177950</td>
<td>0.21873</td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.116682021753212</td>
<td>0.11668202440477</td>
<td>0.11671</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5</td>
<td>0.809781655757175</td>
<td>0.80978206792219</td>
<td>0.83022</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.305348153994317</td>
<td>0.30534815892762</td>
<td>0.30614</td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.162878300446152</td>
<td>0.16287830622144</td>
<td>0.16293</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5</td>
<td>0.946034889114793</td>
<td>0.94601653581617</td>
<td>0.98068</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.380273645209797</td>
<td>0.38027364827726</td>
<td>0.38163</td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>0.168577409006530</td>
<td>0.16857741288584</td>
<td>0.16766</td>
</tr>
</tbody>
</table>

Table 8: Comparison of results for Problem 2 at different times for $v = 0.01, h = 0.01, k = 0.001$
For \( v = 1, h = 0.1, k = 0.00001 \), the comparison of the current results with the exact solution as well as the results of [18] in Table 9 showed that the presented results are very accurate in intervals \( x \in [0, 1] \) and \( t = 0.1 \).

### 5. Result Analysis

In our numerical experiments, to calculate the error of the numerical solution for examples 1, 2, the exact solution needs to be evaluated. In real computational, a large number of terms should be evaluated to ensure high accuracy, a number \( L \) is chosen such that the error is less than \( 1.0e-15 \).

### 6. Conclusion

In this paper, we proposed a new technique for solving one-dimensional Burgers’ equations, as we combined a low-storage Runge-Kutta scheme to approximate the time integration and sixth-order combined compact finite difference scheme to approximate the space derivatives. Numerical results obtained show that the proposed method performs well and is reliable for solving any nonlinear Burger’s equations.
References


**Appendix A: Solving the Twin Tri-Diagonal Matrices**

Here we modify the method discussed by Chu et al. [15] to solve the system with simple technique by generalization of Gauss’s elimination method.

Any twin-tridiagonal systems can be written as following by using $a$, $b$, $c$, $aa$, $bb$, $cc$, $x$, $y$, $z$, $xx$, $yy$, $zz$, $d$, $dd$ as diagonals, sub diagonals, super diagonals and constants vectors respectively as follows:

\[
\begin{pmatrix}
  b_1 & c_1 \\
  a_1 & b_2 & c_2 \\
  & \ddots & \ddots & \ddots \\
  & \ddots & \ddots & \ddots \\
  & & a_{l-2} & b_{l-1} & c_{l-1} \\
  & & a_{l-1} & b_l
\end{pmatrix}
\begin{pmatrix}
  F_1 \\
  F_2 \\
  \vdots \\
  \vdots \\
  F_{l-1} \\
  F_l
\end{pmatrix}
\]
or in the form
\[ \begin{align*}
& b_1 F_1 + c_1 F_2 + \ldots + y_1 G_1 + z_1 G_2 + \ldots = d_1 \\
& bb_1 F_1 + cc_1 F_2 + \ldots + yy_1 G_1 + zz_1 G_2 + \ldots = dd_1 \\
& a_1 F_1 + b_2 F_2 + c_2 F_3 + \ldots + x_1 G_1 + y_2 G_2 + z_2 G_3 + \ldots = d_2 \\
& aa_1 F_1 + bb_2 F_2 + cc_2 F_3 + \ldots + xx_1 G_1 + yy_2 G_2 + zz_2 G_3 + \ldots = dd_2 \\
& + a_2 F_2 + b_3 F_3 + c_3 F_4 + \ldots + x_2 G_2 + y_3 G_3 + z_3 G_4 + \ldots = d_3 \\
& + aa_2 F_2 + bb_3 F_3 + cc_3 F_4 + \ldots + xx_2 G_2 + yy_3 G_3 + zz_3 G_4 + \ldots = dd_3 \\
& \ldots + a_{l-2} F_{l-2} + b_{l-1} F_{l-1} + c_{l-1} F_l + \ldots + x_{l-2} G_{l-2} + y_{l-1} G_{l-1} + z_{l-1} G_l + \ldots = d_{l-1} \\
& + aa_{l-2} F_{l-2} + bb_{l-1} F_{l-1} + cc_{l-1} F_l + \ldots + xx_{l-2} G_{l-2} + yy_{l-1} G_{l-1} + zz_{l-1} G_l + \ldots = dd_{l-1} \\
& + a_{l-1} F_{l-1} + b_l F_l + \ldots + x_{l-1} G_{l-1} + y_l G_l + \ldots = d_l \\
& + aa_{l-1} F_{l-1} + bb_l F_l + \ldots + xx_{l-1} G_{l-1} + yy_l G_l + \ldots = dd_l 
\end{align*} \]
The first equation can be used to eliminate \( F_1 \) from the second, third and fourth equations, and the new second equation can be used to eliminate \( G_1 \) from the third and fourth equations, now then the new third equation can be used to eliminate \( F_2 \) from the fourth, fifth and sixth equations, and the new fourth equation can be used to eliminate \( G_2 \) from the fifth and sixth equations and so on, until finally, the new last equation became with only one unknown \( G_l \), the other unknown \( F_l, G_{l-1}, F_{l-1}, G_{l-2}, F_{l-2}, \ldots, G_2, F_2, G_1, F_1 \) can then be found in turn by back-substitution.