International Journal of Pure and Applied Mathematics
Volume 75 No. 2 2012, 169-184
ISSN: 1311-8080 (printed version)
url: http://www.ijpam.eu

ijpam.eu

COMBINED COMPACT FINITE DIFFERENCE TREATMENT OF BURGERS' EQUATION

M.H. Mousa[1] §, A.A. Abadeer[2], M.M. Abbas[3]

[1]Department of Mathematics
Faculty of Science
Mansoura University
Mansoura, 35516, EGYPT

[2,3]Department 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

Received: June 18, 2011

© 2012 Academic Publications, Ltd.
url: www.acadpubl.eu

§Correspondence author

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

$$\psi_t + \psi\psi_x = v\psi_{xx}, \quad t \in [0,\infty),\ x \in [0,1], \tag{1}$$

$$\psi(x,0) = f(x) \quad \forall 0 \le x \le 1, \tag{2}$$

$$\psi(0,t) = \psi(1,t) = 0, \quad \forall t \in (0,\infty). \tag{3}$$

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.

$$u_t = vu_{xx}, \tag{4}$$

with initial and boundary conditions

$$u(x,0) = \exp\left(\int_0^x \frac{-1}{2v} f(x)dx\right), \tag{5}$$

$$u_x(0,t) = u_x(1,t) = 0. \tag{6}$$

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_i^n, u_{i-1}^n and u_{i+1}^n respectively, its first and second derivatives at the grid point (x_i, t_n) be given by$F_i^n = (u_x)_i^n$, $G_i^n = (u_{xx})_i^n$.

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:

$$\alpha_1 \left(F_{i-1}^n + F_{i+1}^n\right) + F_i^n + h\beta_1 \left(G_{i+1}^n - G_{i-1}^n\right) = \frac{a_1}{h}\left(u_{i+1}^n - u_{i-1}^n\right) \tag{7}$$

$$G_i^n + \alpha_2 \left(G_{i-1}^n + G_{i+1}^n\right) + \frac{\beta_2}{h}\left(F_{i+1}^n - F_{i-1}^n\right) = \frac{a_2}{h^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{8}$$

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

$$\alpha_1 = \tfrac{7}{16}, \beta_1 = -\tfrac{1}{16}, a_1 = \tfrac{15}{16},$$
$$\alpha_2 = -\tfrac{1}{8}, \beta_2 = \tfrac{9}{8}, a_2 = 3$$

will yield

$$\frac{7}{16}F_{i-1}^n + F_i^n + \frac{7}{16}F_{i+1}^n + \frac{h}{16}G_{i-1}^n - \frac{h}{16}G_{i+1}^n = \frac{15}{16h}\left(u_{i+1}^n - u_{i-1}^n\right) \tag{9}$$

$$-\frac{9}{8h}F_{i-1}^n + \frac{9}{8h}F_{i+1}^n - \frac{1}{8}G_{i-1}^n + G_i^n - \frac{1}{8}G_{i+1}^n = \frac{3}{h^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{10}$$

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 $\frac{\partial u}{\partial t} = \phi\left(t, u, F, G\right)$, where ϕ stands for a differentiable function in space,

stage	ρ_j	σ_j	c_j
1	0.	0.2	0
2	-1	0.83204	0.2
3	-1.55798	0.6	0.2
4	-1	0.35394	0.8
5	- 0.45031	0.2	0.8

Table 1: The coefficients for ORK25-6 scheme

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

$$\begin{array}{l} v_1 = u\left(t_n, x_i\right) = u_i^n, \\ T_1 = t_n, \\ w_1 = 0, \\ \left.\begin{array}{l} T_j = t_n + c_j k \\ w_{j+1} = \rho_j w_j + k\phi\left(T_j, v_j, F_j, G_j\right) \\ v_{j+1} = v_j + \sigma_j w_{j+1} \end{array}\right\} \quad j = 1, 2, ..., 5 \\ u\left(t_{n+1}, x_i\right) = u_i^{n+1} = v_{s+1} \end{array},$$

where$F_j = F\left(T_j, v_j\right), G_j = G\left(T_j, v_j\right)$, 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:

$$G_1^n + \ \beta G_2^n + \delta F_2^n = \frac{1}{h^2}\left(a u_1^n + b u_2^n + c u_3^n + d u_4^n + e u_5^n\right) \tag{11}$$

The relations between the coefficients a, b, c, d, e, β and δ 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

$$G_1^n + \ \frac{37}{3} G_2^n + \frac{35}{9h} F_2^n = \frac{1}{h^2}\left(\frac{53}{4} u_1^n - \frac{1753}{54} u_2^n + \frac{43}{2} u_3^n - \frac{5}{2} u_4^n + \frac{23}{108} u_5^n\right) \tag{12}$$

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

$$G_M^n + \frac{37}{3}G_{M-1}^n - \frac{35}{9h}F_{M-1}^n = \frac{1}{h^2}\left(\frac{53}{4}u_M^n - \frac{1753}{54}u_{M-1}^n + \frac{43}{2}u_{M-2}^n - \frac{5}{2}u_{M-3}^n + \frac{23}{108}u_{M-4}^n\right) \quad (13)$$

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

$$F_1^n + \frac{2}{3}F_2^n - 2hG_2^n = \frac{1}{h}\left(-\frac{49}{12}u_1^n + \frac{61}{9}u_2^n - 3u_3^n + \frac{1}{3}u_4^n - \frac{1}{36}u_5^n\right) \quad (14)$$

$$F_M^n + \frac{2}{3}F_{M-1}^n + 2hG_{M-1}^n = \frac{1}{h}\left(\frac{49}{12}u_M^n - \frac{61}{9}u_{M-1}^n + 3u_{M-2}^n - \frac{1}{3}u_{M-3}^n + \frac{1}{36}u_{M-4}^n\right) \quad (15)$$

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 pervious works.

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

$$\psi(x,0) = \sin(\pi x), \quad (16)$$

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

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

The exact solution [3] is given by

$$\psi(x,t) = 2\pi v \frac{\sum_{n=1}^{\infty} a_n n \exp\left(-n^2\pi^2 vt\right)\sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \exp\left(-n^2\pi^2 vt\right)\cos(n\pi x)}. \quad (18)$$

with the Fourier coefficients

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

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:-

$$F_1^n = 0 \tag{20}$$

$$F_M^n = 0 \tag{21}$$

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 $\psi = -2v\frac{u_x}{u}$.

- For $v = 1, h = 0.0125, k = 0.00001$, the comparison of the current results **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 $t = 0.1$.

x	t	Exact	HCCFD6	RHC[3]	[13]
0.25	0.1	0.253637576456303	0.25363757647186	0.264126	0.253622
	0.15	0.156600928504537	0.15660092851768	0.165683	0.156600
	0.2	0.096441887057574	0.09644188706816	0.101617	0.0964475
	0.25	0.059217810173447	0.05921781018154	0.059113	0.059225
0.5	0.1	0.371577476146794	0.37157747616732	0.393354	0.371553
	0.15	0.226823808094687	0.22682380811336	0.251788	0.226823
	0.2	0.138473474504236	0.13847347451939	0.163931	0.138482
	0.25	0.084537611850818	0.08453761186225	0.120967	0.084548
0.75	0.1	0.272581718686695	0.27258171870068	0.285579	0.272563
	0.15	0.164369265924170	0.16436926593776	0.176957	0.164369
	0.2	0.099435368504681	0.09943536851574	0.111020	0.099441
	0.25	0.060347132211840	0.06034713222022	0.068569	0.060354

Table 2: Comparison of results for Problem 1 at different times for $v = 1, h = 0.0125, k = 0.00001$

x	t	Exact	HCCFD6	RHC[3]	[13]
0.25	0.4	0.308894227876420	0.30889422800835	0.317062	0.308902
	0.6	0.240739023290827	0.24073902337374	0.248472	0.240750
	0.8	0.195675570103439	0.19567557015980	0.202953	0.195687
	1	0.162564857110670	0.16256485715196	0.169527	0.162579
0.5	0.4	0.569632450880106	0.56963245100236	0.583408	0.569649
	0.6	0.447205521198856	0.44720552130214	0.461714	0.447239
	0.8	0.359236058515669	0.35923605859791	0.373800	0.359278
	1	0.291915957125836	0.29191595719234	0.306184	0.291962
0.75	0.4	0.625437896424913	0.62543789645580	0.638847	0.625426
	0.6	0.487214974883946	0.48721497494596	0.506429	0.487279
	0.8	0.373921753209456	0.37392175327646	0.393565	0.374005
	1	0.287474405916976	0.28747440597921	0.305862	0.287555
	3.00	0.02977212685877	0.02977212686875	0.034484	0.029786

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

Example 2: Here we will solve the one-dimensional Burgers' equation (1) with initial condition

$$\psi(x, 0) = 4x\,(1 - x)\,, \tag{22}$$

x	t	Exact	HCCFD6	[17]
0.1	0.5	0.121143531499446	0.12114354280213	0.12079
	2.0	0.042963776898489	0.04296377803806	0.04300
	4.0	0.023104232693404	0.02310423297530	0.02324
0.3	0.5	0.360271055868959	0.36027105847179	0.36113
	2.0	0.128839890319171	0.12883989287318	0.12877
	4.0	0.069308290365404	0.06930829113796	0.06935
0.5	0.5	0.588695773501889	0.58869578627457	0.59559
	2.0	0.214558054270833	0.21455805679446	0.21468
	4.0	0.115494756336886	0.11549475744843	0.11550
0.7	0.5	0.793493405966195	0.79349427051027	0.81257
	2.0	0.299997767670026	0.29999776882587	0.30075
	4.0	0.161214654320313	0.16121465556121	0.16125
0.9	0.5	0.938108282222430	0.93810995337230	0.97184
	2.0	0.373277628825965	0.37327762855064	0.37452
	4.0	0.166058721641256	0.16605872299879	0.16515

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

x	exact	HCCFD6	[7]
0.1	0.109538151270509	0.10954222226587	0.10958
0.2	0.209792148910037	0.20979568495887	0.20989
0.3	0.291896350825530	0.29190012701347	0.29199
0.4	0.347923912365551	0.34792707785181	0.34809
0.5	0.371577476146794	0.37157989686308	0.37173
0.6	0.359045579984961	0.35904713273930	0.35920
0.7	0.309905000631105	0.30990591035177	0.31003
0.8	0.227817406627376	0.22781782862166	0.22792
0.9	0.120686691089410	0.12068759038988	0.12071

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

$$u(x,0) = \exp\left(-x^2 (3v)^{-1} (3 - 2x)\right) \tag{23}$$

The exact solution [3] is also given by (12).

x	t	exact	HCCFD6	RHC[3]
0.25	0.1	0.261479814192645	0.26147981422297	0.245579
	0.15	0.161477615167447	0.16147761518815	0.147050
	0.2	0.099469553053455	0.09946955306850	0.086989
	0.25	0.061087582313049	0.06108758232394	0.053258
0.5	0.1	0.383422416438965	0.38342241647595	0.365055
	0.15	0.234055329438479	0.23405532946748	0.221493
	0.2	0.142888087801198	0.14288808782274	0.141403
	0.25	0.087232703460768	0.08723270347635	0.106944
0.75	0.1	0.281572641339867	0.28157264136270	0.263296
	0.15	0.169738279579551	0.16973827960028	0.152572
	0.2	0.102655433756969	0.10265543377266	0.088794
	0.25	0.062289848924514	0.06228984893571	0.054354

Table 6: Comparison of results for Problem 2 at different times for $v = 1, h = 0.0125, k = 0.00001$

With the Fourier coefficients

$$\begin{aligned} a_0 &= \int_0^1 \exp\left(-x^2 (3v)^{-1} (3-2x)\right) dx \\ a_n &= 2\int_0^1 \exp\left(-x^2 (3v)^{-1} (3-2x)\right) \cos(n\pi x)\, dx, \quad n = 1, 2, 3, \cdots \end{aligned} \tag{24}$$

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 $t \in [0, 4]$.

x	t	exact	HCCFD6	RHC[3]
0.25	0.4	0.317522880346768	0.31752288093206	0.306529
	0.6	0.246138455741545	0.24613845606064	0.236051
	0.8	0.199555307716945	0.19955530791867	0.190181
	1	0.165598631696975	0.16559863183902	0.156646
0.5	0.4	0.584537259423137	0.58453725996466	0.565994
	0.6	0.457976404556937	0.45797640497895	0.438926
	0.8	0.367398193136396	0.36739819345298	0.348328
	1	0.298343106946419	0.29834310719240	0.280038
0.75	0.4	0.645615507508048	0.64561550757305	0.626990
	0.6	0.502675751374800	0.50267575164533	0.477908
	0.8	0.385335518826973	0.38533551911836	0.360630
	1	0.295856684503934	0.29585668475873	0.272623
	3.0	0.03043964524096	0.03043964526907	0.024748

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

x	t	exact	HCCFD6	[17]
0.1	0.5	0.128462159628175	0.12846218814552	0.12808
	2.0	0.043813854221127	0.04381385673479	0.04388
	4.0	0.023345001323585	0.02334500195258	0.02351
0.3	0.5	0.378489128693062	0.37848913727667	0.37956
	2.0	0.131345187649400	0.13134519372199	0.13129
	4.0	0.070027182347423	0.07002718412010	0.07009
0.5	0.5	0.609886125745511	0.60988615631258	0.61768
	2.0	0.218588014961054	0.21858802177950	0.21873
	4.0	0.116682021753212	0.11668202440477	0.11671
0.7	0.5	0.809781655757175	0.80978270679221	0.83022
	2.0	0.305348153994317	0.30534815892762	0.30614
	4.0	0.162878300446152	0.16287830362214	0.16293
0.9	0.5	0.946034889114793	0.94601653581617	0.98068
	2.0	0.380273645420979	0.38027364827726	0.38163
	4.0	0.168577409006530	0.16857741288584	0.16766

Table 8: Comparison of results for Problem 2 at different times for $v = 0.01, h = 0.01, k = 0.001$

x	exact	HCCFD6	[18]
0.1	0.112892245268291	0.11289667966323	0.11295
0.2	0.216252142417222	0.21625600407506	0.21662
0.3	0.300965859903397	0.30096984538218	0.30208
0.4	0.358863061468998	0.35886619149367	0.36056
0.5	0.383422416438965	0.38342456283725	0.38543
0.6	0.370657835501244	0.37065896281781	0.37270
0.7	0.320065690908233	0.32006616126201	0.32185
0.8	0.235371149338849	0.23537123354175	0.23668
0.9	0.124718046630702	0.12471878902863	0.12541

Table 9: Comparison of results for Problem 2 for $v = 1, h = 0.1, k = 0.00001$ and $t = 0.1$

- For $v = 1, h = 0.1, k = 0.00001$, the comparison of the current results with the exact solution as well 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

[1] T. Özis, A. Esen, S. Kutluay, Numerical solution of Burgers' equation by quadratic B-spline finite elements, *Applied Mathematics and Computation*, **165** (2005), 237-249.

[2] S. Abbasbandy, M.T. Darvishi, A numerical solution of Burgers' equation by time discretization of Adomian's decomposition method, *Applied Mathematics And Computation*, **170**, No. 1 (2005), 95-102.

[3] M. Gulsu, T. Özis, Numerical solution of Burgers' equation with restrictive Taylor approximation, *Applied Mathematics and Computation*, **171**, No. 2 (2005), 1192-1200.

[4] M. Gulsu, A finite difference approach for solution of Burgers' equation, *Applied Mathematics and Computation*, **175**, No. 2 (2006), 1245-1255.

[5] T. Özis, E.N. Aksan, A. Özdes, A finite element approach for solution of Burgers' equation, *Applied Mathematics and Computation*, **139**, No-s: 2-3 (2003), 417-428.

[6] T. Özis, Y. Aslan, The semi-approximate approach for solving Burgers' equation with high Reynolds' number, *Applied Mathematics and Computation*, **163** (2005), 131-145.

[7] En. Aksan, A. Özdes, T. Özis, A numerical solution of Burgers' equation, *Applied Mathematics and Computation*, **156** (2004), 395-402.

[8] S. Abbasbandy, M.T. Darvishi, A numerical solution of Burgers' equation by modified Adomian method, *Applied Mathematics and Computation*, **163**, No. 3 (2005), 1265-1272.

[9] M. Inc, On numerical solutions of one-dimensional nonlinear Burgers' equation and convergence of the decomposition method, *Applied Mathematics and Computation*, **170**, No. 1 (2005), 76-85.

[10] I.A. Hassanien, A.A. Salama, H.A. Hosham, Fourth-order finite difference method for solving Burgers' equation, *Applied Mathematics and Computation*, **170**, No. 2 (2005), 781-800.

[11] M.K. Kadalbajoo, K.K. Sharma, A. Awasthi, A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equations, *Applied Mathematics and Computation*, **170**, No. 2 (2005), 1365-1393.

[12] M. Inc, On numerical solution of Burgers' equation by homotopy analysis method, *Physics Letters A*, **372**, No. 4 (2008), 356-360.

[13] J. Zueco, A network thermodynamic method for the numerical solution of Burgers' equation, *Mathematical and Computer Modeling*, **47** (2008), 401-410.

[14] M. Sari, G. Gürarslan, A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation, *Applied Mathematics and Computation*, **208**, No. 2 (2009), 475-483.

[15] P.C. Chu, C. Fan, A three-point combined compact difference scheme, *Journal of Computational Physics*, **140** (1998), 370-399.

[16] M. Bernardini, S. Pirozzoli, A general strategy for the optimization of Runge-Kutta schemes for wave propagation phenomena, *Journal of Computational Physics*, **228** (2009), 4182-4199.

[17] A.R. Bahadır, M. Sağlam, A mixed finite difference and boundary element approach to one-dimensional Burgers' equation, *Applied Mathematics and Computation*, **160** (2005), 663-673.

[18] S. Kutluay, A. Esen, A linearized numerical scheme for Burgers-like equations, *Applied Mathematics and Computation*, **156** (2004), 295-305.

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:

$$\begin{pmatrix} b_1 & c_1 & & & & \\ a_1 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{l-2} & b_{l-1} & c_{l-1} \\ & & & & a_{l-1} & b_l \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_{l-1} \\ F_l \end{pmatrix}$$

$$+\begin{pmatrix} y_1 & z_1 & & & & & \\ x_1 & y_2 & z_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & x_{l-2} & y_{l-1} & z_{l-1} \\ & & & & x_{l-1} & y_l \end{pmatrix}\begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ \vdots \\ G_{l-1} \\ G_l \end{pmatrix}=\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{l-1} \\ d_l \end{pmatrix},$$

$$\begin{pmatrix} bb_1 & cc_1 & & & & & \\ aa_1 & bb_2 & cc_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & aa_{l-2} & bb_{l-1} & cc_{l-1} \\ & & & & aa_{l-1} & bb_l \end{pmatrix}\begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_{l-1} \\ F_l \end{pmatrix}$$

$$+\begin{pmatrix} yy_1 & zz_1 & & & & & \\ xx_1 & yy_2 & zz_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & xx_{l-2} & yy_{l-1} & zz_{l-1} \\ & & & & xx_{l-1} & yy_l \end{pmatrix}\begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ \vdots \\ G_{l-1} \\ G_l \end{pmatrix}=\begin{pmatrix} dd_1 \\ dd_2 \\ \vdots \\ \vdots \\ dd_{l-1} \\ dd_l \end{pmatrix},$$

or in the form

$$
\begin{array}{lllllllllll}
b_1F_1 & +c_1F_2 & + & & & +y_1G_1 & +z_1G_2 & + & & & = d_1 \\
bb_1F_1 & +cc_1F_2 & + & & & +yy_1G_1 & +zz_1G_2 & + & & & = dd_1 \\
a_1F_1 & +b_2F_2 & +c_2F_3 & + & & +x_1G_1 & +y_2G_2 & +z_2G_3 & & & = d_2 \\
aa_1F_1 & +bb_2F_2 & +cc_2F_3 & + & & +xx_1G_1 & +yy_2G_2 & +zz_2G_3 & & & = dd_2 \\
 & +a_2F_2 & +b_3F_3 & +c_3F_4 & + & & +x_2G_2 & +y_3G_3 & +z_3G_4 & & = d_3 \\
 & +aa_2F_2 & +bb_3F_3 & +cc_3F_4 & + & & +xx_2G_2 & +yy_3G_3 & +zz_3G_4 & & = dd_3 \\
 & & \ddots & & & & & \ddots & & & \\
 & & & \ddots & & & & & \ddots & & \\
 & & +a_{l-2}F_{l-2} & +b_{l-1}F_{l-1} & +c_{l-1}F_l & & & +x_{l-2}G_{l-2} & +y_{l-1}G_{l-1} & +z_{l-1}G_l & = d_{l-1} \\
 & & +aa_{l-2}F_{l-2} & +bb_{l-1}F_{l-1} & +cc_{l-1}F_l & & & +xx_{l-2}G_{l-2} & +yy_{l-1}G_{l-1} & +zz_{l-1}G_l & = dd_{l-1} \\
 & & & +a_{l-1}F_{l-1} & +b_lF_l & + & & & +x_{l-1}G_{l-1} & +y_lG_l & = d_l \\
 & & & +aa_{l-1}F_{l-1} & +bb_lF_l & + & & & +xx_{l-1}G_{l-1} & +yy_lG_l & = dd_l
\end{array}
$$

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 $F_l, G_{l-1}, F_{l-1}, G_{l-2}, F_{l-2}, \cdots, G_2, F_2, G_1, F_1$can then be found in turn by back-substitution.