Numerical Solutions of the Modified Burgers Equation by a Cubic B-spline Collocation Method

Selcuk Kutluay, Yusuf Ucar, N. Murat Yagmurlu*
Department of Mathematics, İnönü University, Malatya, 44280, TURKEY.
(*corresponding author) e-mail: murat.yagmurlu@inonu.edu.tr

Abstract

In this paper, a numerical solution of the modified Burgers equation is obtained by a cubic B-spline collocation method. In the solution process, a linearization technique based on quasi-linearization has been applied to deal with the non-linear term appearing in the equation. The computed results are compared with others selected from the available literature. The error norms L_2 and L_∞ are computed and found to be sufficiently small. A Fourier stability analysis of the method is also investigated.

Keywords: Modified Burgers equation, Finite element method, Collocation, B-Spline.
AMS 2010 : 97N40, 65L60, 65D07, 74S05

1 Introduction

The Burgers equation

$$u_t + uu_x - \nu u_{xx} = 0$$

where ν is a positive constant and the subscrits x and t denote space and time derivatives respectively, was first introduced by [1] and has been widely used since then. The equation is the simplest nonlinear model equation for diffusive waves in fluid dynamics [2]. Hence it is of great interest for many scientists and mathematicians. Therefore, its analytical and numerical solutions are found by many authors using various methods. The analytical solution of the equation has been obtained by Benton and Platzman [3]. Miller [4] has also obtained infinite series solutions of the problem. As for the numerical one, among others Dag et al. have used B-spline collocation methods for numerical solutions of the Burgers equation, see [5] and references therein. In this paper, we will deal with the modified Burgers equation given in the form

$$u_t + u^2 u_x - \nu u_{xx} = 0, \qquad a \le x \le b \tag{1}$$

where $u = u(x,t)$ is the dependent variable, ν denotes the viscosity parameter, and t and x are the independent time and space parameters, respectively. In (1), when the term u^2u_x is substitued with the term uu_x, then the modified Burgers equation turns into the classical Burgers equation. In the present study, the numerical solutions of Eq. (1) will be sought with the following boundary conditions

$$\begin{array}{ll} u(a,t)=0, & u(b,t)=0, \\ u_x(a,t)=0, & u_x(b,t)=0, \end{array} \qquad t \geq t_0. \tag{2}$$

The main purpose of this study is to apply the cubic B-spline collocation finite element method to develop a numerical technique for solving the modified Burgers equation. Eq. (1) has been solved by several authors using various methods and techniques. As for the numerical one, among others Ramadan and El-Danaf [6] have considered the solution of the modified Burgers equation by using the collocation method with quintic splines. Ramadan et al. [7] have solved the modified Burgers equation numerically using the collocation method with septic splines. Saka and Dag [10] have applied time and space splitting techniques to the Burgers and modified Burgers equations and then employed the quintic B-spline collocation procedure to approximate the resulting systems. Irk [11] has used Crank-Nicolson central differencing scheme for the time integration and sextic B-spline functions for the space integration to the modified and time splitted modified Burgers equation. Temsah [12] has proposed a numerical solution for the convection-diffusion equation using El-Gendi method with interface points and then shown numerical results for Burgers and modified Burgers equations. Grienwank and El-Danaf [13] have proposed a non-polynomial spline based method to obtain numerical solutions of the non-linear modified Burgers equation. Bratsos [14] has used a finite-difference scheme based on rational approximations to the matrix-exponential term in a two-time level recurrence relation for the numerical solution of the modified Burgers equation. Bratsos [15] has presented a finite-difference scheme based on fourth-order rational approximants to the matrix-exponential term in a two-time level recurrence relation for the numerical solution of the modified Burgers equation. Bratsos and Petrakis [16] have used an explicit finite difference scheme based on second-order rational approximations to the matrix-exponential term for the numerical solution of the modified Burgers equation. Roshan and Bhamra [17] have solved the modified Burgers equation numerically by the Petrov-Galerkin method using a linear hat function as the trial function and a cubic B-spline function as the test function, see [17] and references therein. One can find several articles using different methods and techniques to solve various forms of Burgers equation. Among others, Mittal and Arora have proposed a numerical method for the numerical solution of a coupled system of viscous Burgers equation with appropriate initial and boundary conditions, see [18] and references therein.

In this paper, we have used a linearization technique to obtain the numerical solution of the modified Burgers equation. The performance of the method has been tested on a numerical example, and the stability analysis of the numerical scheme has also been investigated.

2 The Finite Element Solution

2.1 Analysis of the method

Let's assume that the interval $[a, b]$ is divided into N finite elements having uniform equal length by the knots x_m, $m = 0(1)N$ such that $a = x_0 < x_1 \cdots < x_{N-1} < x_N = b$ and $h = x_{m+1} - x_m$. The cubic B-splines $\phi_m(x)$, $(m = -1(1)N + 1)$ at the knots x_m are defined over the interval $[a, b]$ as [19]

$$\phi_m(x) = \frac{1}{h^3}\begin{cases} (x - x_{m-2})^3, & x \in [x_{m-2}, x_{m-1}], \\ h^3 + 3h^2(x - x_{m-1}) + 3h(x - x_{m-1})^2 - 3(x - x_{m-1})^3, & x \in [x_{m-1}, x_m], \\ h^3 + 3h^2(x_{m+1} - x) + 3h(x_{m+1} - x)^2 - 3(x_{m+1} - x)^3, & x \in [x_m, x_{m+1}], \\ (x_{m+2} - x)^3, & x \in [x_{m+1}, x_{m+2}], \\ 0, & otherwise. \end{cases}$$

The set of cubic B-splines $\left\{\phi_{-1}(x), \phi_0(x), \ldots, \phi_{N+1}(x)\right\}$ constitutes a basis for the functions to be defined over the interval $[a, b]$. Thus, an approximation solution $U(x, t)$ to analytical solution $u(x, t)$ on this interval can be written in terms of these cubic B- splines as

$$U(x, t) = \sum_{m=-1}^{N+1} \delta_m(t)\phi_m(x) \tag{3}$$

in which $\delta_m(t)$'s are unknown time dependent element parameters to be determined. Because of the fact that each cubic B-spline covers four elements, on the other hand, each element $[x_m, x_{m+1}]$ is covered by four cubic B-splines. In this paper, the finite elements are identified with the interval $[x_m, x_{m+1}]$ and the elements knots x_m and x_{m+1}. In terms of the local coordinate transformation $\xi = x - x_m$, the cubic B-splines can now be expressed in terms of the local variable ξ as follows

$$\begin{array}{l} \phi_{m-1} \\ \phi_m \\ \phi_{m+1} \\ \phi_{m+2} \end{array} = \frac{1}{h^3}\begin{cases} (h - \xi)^3, \\ h^3 + 3h^2(h - \xi) + 3h(h - \xi)^2 - 3(h - \xi)^3, \\ h^3 + 3h^2\xi + 3h\xi^2 - 3\xi^3, \\ \xi^3, \end{cases} \quad 0 \leq \xi \leq h. \tag{4}$$

Since all other cubic B-splines are identically zero over the element $[x_m, x_{m+1}]$, the variation of $U(x, t)$ in Eq. (3) over a typical element $[x_m, x_{m+1}]$ is written as

$$U(\xi, t) = \sum_{j=m-1}^{m+2} \delta_j(t)\phi_j(\xi). \tag{5}$$

If we use the Eqs. (4) and (5), then the nodal values of U_m, U'_m and U''_m at the knots $x = x_m$ can be easily found in terms of element parameter δ_m as follows

$$
\begin{aligned}
U_m &= U(x_m) = \delta_{m-1} + 4\delta_m + \delta_{m+1}, \\
U'_m &= U'(x_m) = \tfrac{3}{h}(-\delta_{m-1} + \delta_{m+1}), \\
U''_m &= U''(x_m) = \tfrac{6}{h^2}(\delta_{m-1} - 2\delta_m + \delta_{m+1}).
\end{aligned} \tag{6}
$$

In place of U_t, we use the first order difference formula, and in places of U_x and U_{xx} in Eq. (1), we use the following approximations arising from the Crank-Nicolson method. After these replacements, we easily obtain the following approximation

$$
\frac{U^{n+1} - U^n}{\Delta t} + \frac{(U^2U_x)^{n+1} + (U^2U_x)^n}{2} - \nu\frac{U^{n+1}_{xx} + U^n_{xx}}{2} = 0. \tag{7}
$$

If the nonlinear term $(U^2U_x)^{n+1}$ in Eq. (7) is linearized by the approximation

$$
(U^2U_x)^{n+1} = 2U^{n+1}U^nU^n_x + U^nU^nU^{n+1}_x - 2U^nU^nU^n_x
$$

which is similar to the one first introduced by Rubin and Graves [20], then we finally obtain the following formula

$$
\begin{aligned}
& U^{n+1} + \frac{\Delta t}{2}(2U^{n+1}U^nU^n_x + U^nU^nU^{n+1}_x) - \nu\frac{\Delta t}{2}U^{n+1}_{xx} \\
= \; & U^n - \frac{\Delta t}{2}(U^2U_x)^n + \nu\frac{\Delta t}{2}U^n_{xx} + \Delta t(U^nU^nU^n_x).
\end{aligned} \tag{8}
$$

Using the nodal values given by Eq. (6) in the Eq. (8), we obtain the following iterative system

$$
\begin{aligned}
& \delta^{n+1}_{m-1}(1 + \frac{3\Delta t}{h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})(\delta^n_{m+1} - \delta^n_{m-1}) - \\
& \frac{3\Delta t}{h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})^2 - \nu\frac{3\Delta t}{h^2}) + \\
& \delta^{n+1}_m(4 + \frac{12\Delta t}{h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})(\delta^n_{m+1} - \delta^n_{m-1}) + \nu\frac{6\Delta t}{h^2}) + \\
& \delta^{n+1}_{m+1}(1 + \frac{3\Delta t}{h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})(\delta^n_{m+1} - \delta^n_{m-1}) + \\
& \frac{3\Delta t}{h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})^2 - \nu\frac{3\Delta t}{h^2}) \\
= \; & (\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1}) - \frac{3\Delta t}{2h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})^2(\delta^n_{m+1} - \delta^n_{m-1}) \\
& + \nu\frac{3\Delta t}{h^2}(\delta^n_{m-1} - 2\delta^n_m + \delta^n_{m+1}) + \frac{3\Delta t}{h}(\delta^n_{m-1} + 4\delta^n_m + \delta^n_{m+1})^2(\delta^n_{m+1} - \delta^n_{m-1}).
\end{aligned} \tag{9}
$$

This iterative system (9) consists of $N+1$ equations and $N+3$ unknown parameters $(\delta_{-1}, \delta_0, \ldots, \delta_N, \delta_{N+1})^T$. In order for this system to have a unique solution, we need two additional constraints. These two additional constraints are obtained from the boundary conditions and then are used to eliminate δ_{-1} and δ_{N+1} from the system (9). Then, this system of equations becomes a matrix equation with the $N+1$ unknowns $\mathbf{d} = [\delta_0, \delta_1, \ldots, \delta_N]^T$ in the form

$$\mathbf{A}\mathbf{d}^{n+1} = \mathbf{B}\mathbf{d}^n. \tag{10}$$

Here, both of the matrices A and B are tridiagonal $(N+1) \times (N+1)$ matrices and therefore are easily solved using a variant of Thomas algorithm [8].

2.2 Initial state

2.2.1 The initial vector

To proceed with the iterative formula (10), first of all, we do need the initial vector d^0 which is going to be determined from the initial and boundary conditions. To achieve this, the approximation (3) ought to be rewritten for the initial condition as

$$U(x,t_0)= \sum_{m=-1}^{N+1} \delta_m(t_0)\phi_m(x)$$

where the δ_m's are unknown element parameters. Now, if we force the initial numerical approximation $U_N(x,t_0)$ comply with the following boundary conditions to discard δ_{-1} and δ_{N+1}

$$\begin{array}{c} U(x_m,t_0) = u(x_m,t_0), \quad m = 0, 1, ..., N \\ (U)_x(a,t_0) = 0, \qquad (U)_x(b,t_0) = 0, \end{array}$$

we obtain the matrix form for the initial vector $\mathbf{d}^0$ as

$$\mathbf{W}\mathbf{d}^0 = \mathbf{b}$$

where

$$\mathbf{W} = \begin{bmatrix} 4 & 2 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & & & \ddots & & \\ & & & & 1 & 4 & 1 \\ & & & & & 2 & 4 \end{bmatrix}$$

$$\mathbf{d}^0 = (\delta_0, \delta_1, \delta_2, \ldots, \delta_{N-2}, \delta_{N-1}, \delta_N)^T$$

and

$$\mathbf{b} = (u(x_0,t_0), u(x_1,t_0), u(x_2,t_0), \ldots, u(x_{N-2},t_0), u(x_{N-1},t_0), u(x_N,t_0))^T.$$

2.3 Stability analysis

To investigate stability analysis of the scheme, it is convenient to use the Fourier method in which the growth factor of a typical Fourier mode is defined as [9]

$$\delta_j^n = \hat{\delta}^n e^{ij\phi}, \tag{11}$$

where ϕ is a real number ($i = \sqrt{-1}$). To apply the von Neumann stability analysis, on very small time increments, if we take $U^2 \simeq U^n U^n \simeq U^{n+1} U^n$ as a local constant $\widehat{U}$ in Eq. (8) from which iterative equation (9) has been obtained, then the equation becomes

$$\begin{aligned} & \delta_{m-1}^{n+1}(1 - \frac{3\Delta t}{2h}\widehat{U} - v\frac{3\Delta t}{h^2}) + \delta_m^{n+1}(4 + v\frac{6\Delta t}{h^2}) + \delta_{m+1}^{n+1}(1 + \frac{3\Delta t}{2h}\widehat{U} - v\frac{3\Delta t}{h^2}) \qquad (12) \\ = \; & \delta_{m-1}^{n}(1 + \frac{3\Delta t}{2h}\widehat{U} + v\frac{3\Delta t}{h^2}) + \delta_m^{n}(4 - v\frac{6\Delta t}{h^2}) + \delta_{m+1}^{n}(1 - \frac{3\Delta t}{2h}\widehat{U} + v\frac{3\Delta t}{h^2}). \end{aligned}$$

for $m = 1(1)N-1$.

Substituting the Fourier mode (11) into the iterative formula (12) and writing $\hat{\delta}^{n+1} = g\hat{\delta}^n$, the linearized recurrence relationship results in the growth factor g as follows:

$$g = \frac{a - ib}{c - id}$$

where

$$\begin{aligned} a &= 4h^2 - 6\nu\Delta t + 2(h^2 + 3\nu\Delta t)\cos\phi, \\ b &= 3h\widehat{U}\Delta t \sin\phi, \\ c &= 4h^2 + 6\nu\Delta t + 2(h^2 - 3\nu\Delta t)\cos\phi \\ d &= -3h\widehat{U}\Delta t \sin\phi. \end{aligned}$$

Since

$$c^2 + d^2 - a^2 - b^2 = 96h^2\nu\Delta t(2 + \cos\phi)\sin[\frac{\phi}{2}]^2 \geq 0$$

the von Neumann necessary criterion for stability condition $|g| \leq 1$ is satisfied. Therefore the linearized scheme is unconditionally stable.

3 Numerical examples and results

In this section, numerical results of the test problem considered in the below have been obtained and all computations have been executed on a Pentium i7 PC in the Fortran code using double precision arithmetic. The accuracy of the method is measured by the error norms L_2 and L_∞ defined as

$$\begin{aligned} L_2 &= \|u-U\|_2 = \sqrt{h\sum_{j=0}^{N}\left|u_j-(U)_j\right|^2}, \\ L_\infty &= \|u-U\|_\infty = \max_j \left|u_j-(U)_j\right| \end{aligned}$$

respectively. In this study, to implement the performance of the scheme, as a test problem we consider the modified Burgers (1) equation with the boundary conditions (2) and the initial condition

$$u(x,1) = \frac{x}{1+(1/c_0)\exp(x^2/4\nu)}, \qquad t \geq 1, \qquad 0 \leq x \leq 1$$

where c_0 is a constant, $0 < c_0 < 1$.

The analytical solution of this problem is [7]

$$u(x,t) = \frac{x/t}{1+\sqrt{t}/c_0\exp(x^2/4\nu t)}, \qquad t \geq 1, \qquad 0 \leq x \leq 1.$$

During the solution process, various viscosity constants $\nu = 0.01, 0.001, 0.005$, space steps $h = 0.005$, time steps $\Delta t = 0.01$ and $c_0 = 0.5$ will be taken over the problem domain $[0,1]$. First of all, the program has been run until the time $t = 11$ and then the error norms L_2 and L_∞ are computed and presented in Table 1 for different values of viscosity ν. As it is obviously seen from the table, both of the error norms L_2 and L_∞ are small enough. In Table 2, we have listed the numerical and analytical solutions of the problem for various values of ν, h and Δt at times $t = 2, 6$ and 10. From the table, it is clearly seen that for $\nu = 0.01$ there exists a discrepancy between the numerical and exact solutions around the right hand-side of boundary when the solution domain is taken as $[0,1]$. However, when the solution domain is taken as $[0,1.3]$, the discrepancy between the numerical and analytical solutions smooths out. The computed error norms L_2 and L_∞ have been compared with those of some other authors for various values of h and ν in Table 3. The table clearly shows that both of the error norms are better or as good as the others found in the literature. If we consider the fact that the present method uses a lower degree base function, namely cubic B-splines, we can say that the present method yields much better results at much lower costs.

Then the obtained numerical results together with their errors are graphed in Figures 1-3 for various values of ν at different time levels. The graphs of the errors have been drawn at time $t = 10$. It can be seen that the maximum error happens at

Table 1: Comparison of the computed error norms L_2 and L_∞ with $h = 0.005$ and $\Delta t = 0.01$ for various values of ν.

	$\nu = 0.01$		$\nu = 0.005$		$\nu = 0.001$	
t	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.378848	0.816262	0.225949	0.579150	0.067286	0.259046
3	0.344560	0.709949	0.205429	0.503503	0.061409	0.225358
4	0.317165	0.605190	0.188055	0.429111	0.056346	0.192213
5	0.307896	0.526341	0.175034	0.372673	0.052504	0.166926
6	0.326003	0.525791	0.164589	0.329766	0.049394	0.147826
7	0.369938	0.755043	0.155888	0.296209	0.046765	0.132819
8	0.427983	0.963399	0.148688	0.269279	0.044486	0.120756
9	0.489147	1.139612	0.143137	0.247219	0.042478	0.110820
10	0.547020	1.281253	0.139607	0.228838	0.040688	0.102585
11	0.598717	1.390450	0.138473	0.213415	0.039078	0.095516

the right-hand boundary of the solution domain for $\nu = 0.01$. However, the errors for $\nu = 0.005$ and $\nu = 0.001$ have been recorded around the points where the waves get their highest amplitudes.

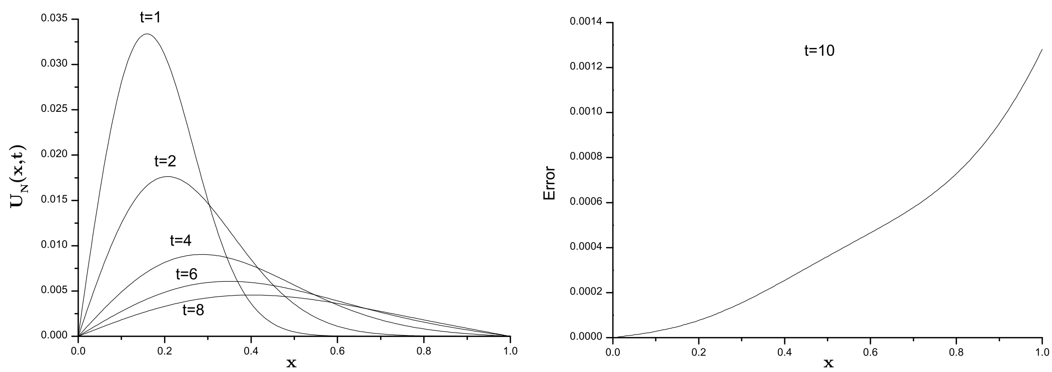

Figure 1: The numerical solutions at different times with $\nu = 0.01$.

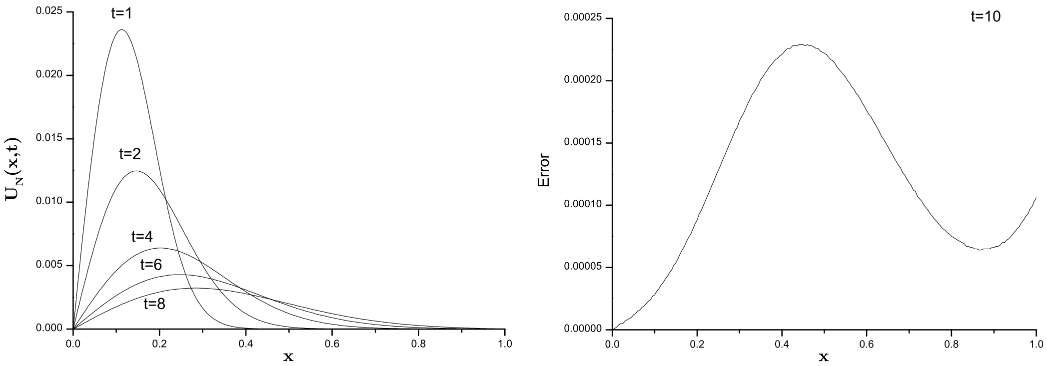

Figure 2: The numerical solutions at different times with $\nu = 0.005$.

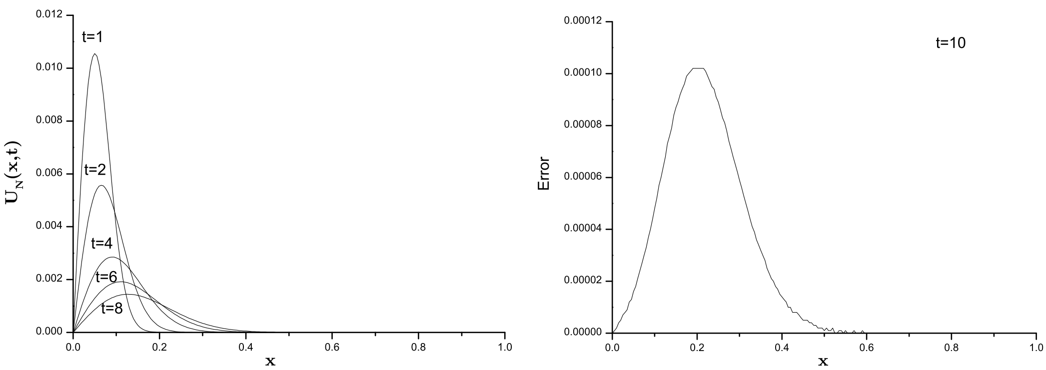

Figure 3: The numerical solutions at different times with $\nu = 0.001$.

Table 2: Comparison of the numerical and analytical solutions for various values of ν and $h = 0.005$, $\Delta t = 0.01$ at $t = 2, 6, 10$.

		$t=2$		$t=6$		$t=10$	
	x	$U_N(x,t)$	$U(x,t)$	$U_N(x,t)$	$U(x,t)$	$U_N(x,t)$	$U(x,t)$
ν =0.01	[0,1]						
	0.2	0.017612	0.017658	0.004863	0.004911	0.002426	0.002503
	0.4	0.008484	0.009133	0.005945	0.006324	0.003579	0.003833
	0.6	0.001116	0.001174	0.003911	0.004356	0.003159	0.003624
	0.8	0.000047	0.000047	0.001562	0.001865	0.001747	0.002475
	1.0	0.000000	0.000001	0.000000	0.000526	0.000000	0.001281
	[0,1.3]						
	0.2	0.017612	0.017658	0.004863	0.004911	0.002434	0.002503
	0.4	0.008484	0.009133	0.005946	0.006324	0.003616	0.003833
	0.6	0.001116	0.001174	0.003923	0.004356	0.003303	0.003624
	0.8	0.000047	0.000047	0.001653	0.001865	0.002192	0.002475
	1.0	0.000001	0.000001	0.000468	0.000526	0.001089	0.001281
	1.3	0.000000	0.000000	0.000000	0.000039	0.000000	0.000300
ν =0.005	[0,1]						
	0.2	0.011006	0.011510	0.004127	0.004253	0.002204	0.002292
	0.4	0.001215	0.001287	0.003082	0.003404	0.002432	0.002653
	0.6	0.000013	0.000013	0.000892	0.001006	0.001352	0.001528
	0.8	0.000000	0.000000	0.000118	0.000131	0.000437	0.000512
	1.0	0.000000	0.000000	0.000000	0.000008	0.000000	0.000106
ν =0.001	[0,1]						
	0.2	0.000227	0.000238	0.001109	0.001237	0.000997	0.001099
	0.4	0.000000	0.000000	0.000016	0.000017	0.000102	0.000116
	0.6	0.000000	0.000000	0.000000	0.000000	0.000001	0.000001
	0.8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Table 3: Comparison of the computed error norms L_2 and L_∞ with results from [3-5] at $t = 2, 6, 10$.

	$t=2$		$t=6$		$t=10$	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
h=0.005, $\Delta t = 0.01, \nu = 0.01$						
Present	0.37885	0.81626	0.32600	0.52579	0.54702	1.28125
[6]	0.52308	1.21698	0.49023	0.72249	0.64007	1.28124
[7]	0.79043	1.70309	0.57672	0.76105	0.80026	1.80329
[10], (QBCA1)	0.37932	0.81680	0.32602	0.52579	0.54701	1.28125
[10], (QBCA2)	0.37951	0.82212	0.32427	0.52579	0.54354	1.28125
[11], (SBCM1)	0.38489	0.82934	-	-	0.54826	1.28127
[11], (SBCM2)	0.39078	0.82734	-	-	0.54612	1.28127
Present, [0,1.3]	0.37885	0.81626	0.27626	0.46514	0.25396	0.32449
[11], (SBCM1), [0,1.3]	0.38489	0.82934	-	-	0.25586	0.32723
[11], (SBCM2), [0,1.3]	0.39078	0.82734	-	-	0.25259	0.32337
h=0.005, $\Delta t = 0.01, \nu = 0.005$						
Present	0.22595	0.57915	0.16459	0.32977	0.13961	0.22884
[10], (QBCA1)	0.22651	0.57998	0.16460	0.32987	0.13959	0.22885
[10], (QBCA2)	0.22697	0.58660	0.16428	0.33654	0.13792	0.23506
h=0.005, $\Delta t = 0.001, \nu = 0.005$						
Present	0.22597	0.57919	0.16459	0.32977	0.13961	0.22884
[6]	0.25786	0.72264	0.22569	0.43082	0.18735	0.30006
[11], (SBCM1)	0.22890	0.58623	-	-	0.14042	0.23019
[11], (SBCM2)	0.23397	0.58424	-	-	0.13747	0.22626
h=0.005, $\Delta t = 0.01, \nu = 0.001$						
Present	0.06729	0.25905	0.04939	0.14783	0.04069	0.10259
[6]	0.06703	0.27967	0.06046	0.17176	0.05010	0.12129
[7]	0.18355	0.81862	0.08142	0.21348	0.05512	0.13943
[10], (QBCA1)	0.06811	0.26094	0.04942	0.14810	0.04067	0.10264
[10], (QBCA2)	0.06953	0.27283	0.04917	0.15656	0.04000	0.10835
[11], (SBCM1)	0.06843	0.26233	-	-	0.04080	0.10295
[11], (SBCM2)	0.07220	0.25975	-	-	0.03871	0.09882
h=0.02, $\Delta t = 0.01, \nu = 0.01$						
Present	0.37179	0.80467	0.32890	0.52579	0.55839	1.28125
[7]	0.79043	1.70309	0.51672	0.76105	0.80026	1.80239
[10], (QBCA1)	0.37911	0.81254	0.32941	0.52579	0.55848	1.28125
[10], (QBCA2)	0.39473	0.88383	0.31588	0.53910	0.52425	1.28125
[11], (SBCM1)	0.38474	0.82611	-	-	0.55985	1.28127
[11], (SBCM2)	0.41321	0.81502	-	-	0.55095	1.28127

4 Conclusions

In this paper, numerical solutions of the Modified Burgers equation based on the cubic B-spline finite element method have been calculated and presented. A test problem is worked out to examine the performance of the present algorithm. The performance and efficiency of the method are shown by calculating the error norms L_2 and L_∞. The obtained results show that the error norms are sufficiently small during all computer runs. The obtained results indicate that the present method is a particularly successful numerical scheme to solve the Modified Burgers equation. We have observed that the use of cubic B-splines with the linearization technique has an advantage over the use of higher degree B-splines in terms of computational labor and accuracy of the obtained results. As a conclusion, the method can be efficiently applied to this type of non-linear problems arising in physics and mathematics with success.

References

[1] H. Bateman, *Some recent researches on the motion of fluids,* Monthly Weather Rev. 43 (1915) 163-170.

[2] R.C. Mittal and R.K. Jain, *Numerical solutions of nonlinear Burgers equation with modified cubic B-splines collocation method*, Applied Mathematics and Computation 218 (2012) 7839-7855.

[3] E. Benton, G.W. Platzman, *A table of folutions of the one-dimensional Burgers equations*, Quart. Appl. Math. 30 (1972) 195-212.

[4] E.L. Miller, *Predictor-Corrector studies of Burger's model of turbulent flow* , M. S. Thesis, University of Delaware, Newark, Deleware, 1966.

[5] I. Dag, D. Irk and A. Sahin, *B-spline collocation methods for numerical solutions of the Burgers equation*, Matmematical Problems in Engineering 2005:5 (2005) 521-538.

[6] M. A. Ramadan and T. S. El-Danaf, *Numerical treatment for the modified burgers equation*, Matmematics and Computers in Simulation 70 (2005) 90-98.

[7] M. A. Ramadan , T. S. El-Danaf and F. E.I. Abd Alaal, *A numerical solution of the Burgers equation using septic B-splines*, Chaos, Solitons and Fractals 26 (2005) 795–804.

[8] R.L. Burden and J.D. Faires, *Numerical Analysis,* Brooks/cole Cenaage Learning, USA, 2011.

[9] G.D. Smith, *Numerical Solution of Partial Differential Equation: Finite Difference Method,* Learendom Press, Oxford, 1978.

[10] B. Saka and I. Dag, *A numerical study of the Burgers equation,* Journal of the Franklin Institute 345 (2008) 328–348.

[11] D. Irk, *Sextic B-spline collocation method for the modified Burgers equation*, Kybernetes, Vol. 38, No. 9, 2009, pp. 1599-1620.

[12] R.S. Temsah, *Numerical solutions for convection–diffusion equation using El-Gendi method,* Communications in Nonlinear Science and Numerical Simulation 14 (2009) 760–769.

[13] A. Griewank and T. S. El-Danaf, *Efficient accurate numerical treatment of the modified Burgers equation*, Applicable Analysis, vol. 88, No. 1, January 2009, 75-87.

[14] A. G. Bratsos, *An implicit numerical scheme for the modified Burgers equation*, in HERCMA 2009 (9º Hellenic-European Conference on Computer Mathematics and its Applications), 24-26 September 2009, Athens, Greece.

[15] A.G. Bratsos, A fourth-order numerical scheme for solving the modified Burgers equation, Computers and Mathematics with Applications 60 (2010) 1393 1400.

[16] A. G. Bratsos and L. A. Petrakis, *An explicit numerical scheme for the modified Burgers equation*, International Journal for Numerical Methods in Biomedical Engneering, 2011, 27: 232-237.

[17] T. Roshan and K. S. Bhamra, *Numerical solutions of the modified Burgers equation by Petrov-Galerkin method*, Applied Mathematics and Computation, 218 (2011) 3673-3679.

[18] R.C. Mittal and G. Arora, *Numerical solution of the coupled viscous Burgers equation*, Commun Nonlinear Sci Numer Simulat, 16 (2011) 1304-1313.

[19] P.M. Prenter, Splines and Variational Methods, John Wiley, New York, 1975.

[20] S. G. Rubin and R. A. Graves, *A Cubic spline approximation for problems in fluid mechanics,* Nasa TR R-436, Washington, DC, 1975.