A B-SPLINE-SSPRK54 METHOD FOR ADVECTION-DIFFUSION PROCESSES

SHKO ALI TAHIR, MURAT SARI

ABSTRACT. In this work, we have developed here a striking numerical method for solving the Burgers equation. The numerical scheme is based on collocation of the modified cubic B-splines basis functions in space variable. The obtained results have been computed without using any linearization and transformation processes. The produced diagonal system has been solved by the optimal strong stability preserving time stepping Runge-Kutta for five stage and order four scheme(SSPRK54). The present approach has been seen to be appropriate for the advection dominant cases. The effectiveness of this method has been verified by considering some test problems. The numerical solutions are in good agreement with the exact solutions and available literature. The present method has been seen to be relatively easy and economical for researchers. And also, the proposed scheme needs relatively less storage space and computational time.

1. INTRODUCTION

Consider the Burgers equation

$$\frac{\partial u}{\partial t}(x,t) - \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x} - f(x,t) = 0, \quad (x,t) \in \Omega = [a,b] \times [t_0, T], \qquad (1.1)$$

with the initial and boundary conditions are given by

$$u(x,t_0) = u_0(x), \qquad (1.2)$$

$$u(a,t) = g_1(t), \quad u(b,t) = g_2(t), \qquad (1.3)$$

where $Re = \dfrac{1}{\lambda}$ is the Reynolds number of the viscous fluid flow problem, g_1, g_2 and u_0 are known functions. Here, $f(x,t)$ represents the source term.

The structure of the Burgers equation can be seen as the one dimensional Navier Stokes equation with no stress term, and is a useful model for various kinds of physical phenomena such as turbulence, fluid mechanics, traffic flow and shock waves. It was first introduced by Bateman[2] in 1915. Since then, the study of this model has been considered by many researchers. It is well known that the exact solution

of the Burgers equation can only be computed for restricted values of the Reynolds number.

In the last half century, many authors have used many numerical techniques for solving the Burgers equation for some reasons: the Burgers equation takes into account both nonlinear advection and dissipation terms for simulating the physical behavior of wave motion and its shock wave behavior when the Reynolds number Re is large. The development of numerical methods for seeking accurate and efficient solutions of the Burgers equation with large values of Re, still remains as a challenging task. Various numerical methods have been studied, for instance see [[1, 7, 3, 10, 13, 15]]

Schoenberg [17] found mathematical relations to splines in the context of piecewise polynomial approximations. Types of piecewise polynomial spline and B-spline functions are utilized with other numerical techniques for getting the numerical solutions of the Burgers equation playing an important role in their computationally. This means that any B-spline basis function of order n is non zerop over at most n adjacent intervals and zero otherwise [4, 8, 18].

In this paper, we have proposed a numerical scheme to find the approximate solution of the Burgers equation to achieve this, we accept a modified cubic B-spline approximation in space. It produces a system of first order ODEs and obtains always a diagonal matrix. We do not meet the question of the linearizion and transformation processes. Therefore, we can solve the resulting system of the ODEs by the optimal five stage and order four strong stability preserving time stepping Runge-Kutta (SSPRK54) scheme [19]. The SSP is a more suitable approach in high order time discretization methods preserve the strong stability properties in any norm of the spatial discretization with first-order Euler time stepping. In order to have stability when using explicit numerical schemes, we are required to apply the CFL condition [14]. Thus, the optimal SSPRK54 scheme is made more efficient by the CFL. The optimality of this scheme is guaranteed by using an approach based on global optimization. Therefore, the proposed method needs less storage space and low cost. In addition this is why we interested in the SSPRK54 scheme.

The remainder of this paper is organized as follows. In section 2, we give a brief introduction of the numerical technique using new cubic B-spline interpolation to solve the model equation. In section 3, we present two problems, and compare the obtained results by the proposed scheme with exact solutions and those already available in the literature. In section 4, we present some final remarks.

2. Description of the method

Consider the mesh points $a = x_0 < x_1 < \ldots < x_m = b$ with uniform length $h = x_j - x_{j-1}$ for $j = 1, 2, \ldots, N$. Our numerical scheme for solving (1.1) is to find an approximation $S(x, t)$ to the exact solution $u(x, t)$ which can be expressed in term of the collocation method with cubic B-splines as trial functions:

$$S(x) = \sum_{j=-1}^{N+1} \delta_j(t) B_j(x), \ \forall x \in [a, b], \tag{2.1}$$

where $\delta_i(t)$ are time dependent quantities to be determined and $B_j(x)$ are cubic B-spline functions. The cubic B-spline $B_j(x)$ with required properties at the knots are given by [16]

$$B_j(x) = \frac{1}{h^3} \begin{cases} 0 & x < x_{j-2} \text{ or } x \geq x_{j-2}, \\ (x - x_{j-2})^3 & x_{j-2} \leq x < x_{j-1}, \\ h^3 + 3h^2(x - x_{j-1}) + 3h(x - x_{j-1})^2 - 3(x - x_{j-1})^3 & x_{j-1} \leq x < x_j, \\ h^3 + 3h^2(x_{j+1} - x) + 3h(x_{j+1} - x)^2 - 3(x_{j+1} - x)^3 & x_j \leq x < x_{j+1}, \\ (x_{j+2} - x)^3 & x_{j+1} \leq x < x_{j+2} \\ 0 & otherwise, \end{cases} \tag{2.2}$$

where the set of splines $\{B_{-1}, B_0, \ldots, B_N, B_{N+1}\}$ construct a basis over the domain $[a, b]$. By using the spline function (2.1) and cubic splines (2.2), the values of B-spline $B_j(x)$ and their derivatives can be calculated at the nodes x_j in term of the time parameters δ_j by

$$\begin{array}{rcl} S(x_j) & = & \delta_{j-1} + 4\delta_j + \delta_{j-1} \\ S'(x_j) & = & (3/h)(\delta_{j+1} - \delta_{j-1}) \\ S''(x_j) & = & (6/h^2)(\delta_{j-1} - 2\delta_j + \delta_{j+1}), \end{array} \tag{2.3}$$

where $S_j = S(x_j)$. In order to obtain a tridiagonal matrix system of differential equations, we have defined new cubic B-spline basis functions to solve the Burgers equation as follows:

$$\begin{array}{rcll} \mathcal{B}_0(x) & = & B_0(x) & j = 0 \\ \mathcal{B}_1(x) & = & B_1(x) - B_{-1}(x) & j = 1 \\ \mathcal{B}_j(x) & = & B_j(x) & j = 2, 3, \ldots, N-2 \\ \mathcal{B}_{N-1} & = & B_{N-1}(x) - B_{N+1}(x) & j = N - 1 \\ \mathcal{B}_N & = & B_N(x) & j = N. \end{array} \tag{2.4}$$

Now assume the approximation solution is given by

$$S(x) = \sum_{j=0}^{N} \delta_j(t)\mathcal{B}_j(x) \ \forall x \in [a, b]. \tag{2.5}$$

Here, we apply the proposed method by using approximate solution (2.5) with the modified set of cubic B-splines given by (2.4) at the knots. The rest of our numerical scheme for solving the Burgers equation can be expressed as follows:

At boundaries (1.3) for $x = x_0$ and $x = x_N$ the approximation solution (2.5) becomes

$$\begin{array}{rclcl} S(x_0, t) & = & \delta_0(t)\mathcal{B}_0(x_0) + \delta_1(t)\mathcal{B}_1(x_0) & = & g_1(t) \\ S(x_N, t) & = & \delta_{N-1}(t)\mathcal{B}_{N-1}(x_N) + \delta_N(t)\mathcal{B}_N(x_N) & = & g_2(t). \end{array} \tag{2.6}$$

Substitution of the approximate solution (2.5) in (1.1) leads to

$$\sum_{j=0}^{N} \delta_j'(t)\mathcal{B}_j(x) = -\Big(\sum_{j=0}^{N} \delta_j(t)\mathcal{B}_j(x)\Big)\Big(\sum_{j=0}^{N} \delta_j(t)\mathcal{B}_j'(x)\Big) + \lambda\Big(\sum_{j=0}^{N} \delta_j(t)\mathcal{B}''_j(x)\Big) + f(x,t), \tag{2.7}$$

where $\delta'(t)$ is the first derivative with respect to t. The cubic B-splines basis $\mathcal{B}'_j(x)$ and $\mathcal{B}''_j(x)$ denote the first and the second differentiation with respect to x. Let us discretize the domain $[a, b]$ into grid points and let us take $x = x_j$ for $j = 0, \ldots, N$ in equation (2.7). We thus obtain

$$\sum_{j=0}^{N} \delta_j'(t)\mathcal{B}_j(x_j) = -\Big(\sum_{j=0}^{N} \delta_j(t)\mathcal{B}_j(x_j)\Big)\Big(\sum_{j=0}^{N} \delta_j(t)\mathcal{B}_j'(x_j)\Big) + \lambda\Big(\sum_{j=0}^{N} \delta_j(t)\mathcal{B}''_j(x_i)\Big) + f(x_j, t). \tag{2.8}$$

By using the approximation values of $S(x_j)$, $S'(x_j)$ and $S''(x_j)$ given by equations (2.3) at the knots in equation (2.8), we get the following difference equations with the variables $\delta(t)$,

$$
\begin{cases}
4\delta'_0 & = & g'_0(t) & j = 0, \\[2mm]
\delta'_{j-1} + 4\delta'_j + \delta'_{j+1} & = & \dfrac{-3}{h}(\delta_{j-1} + 4\delta_j + \delta_{j+1})(\delta_{j+1} - \delta_{j-1}) & \\[2mm]
& + & \dfrac{6\lambda}{h^2}(\delta_{j-1} - 2\delta_j + \delta{j} + 1) & j = 1, 2, \ldots, N-1, \\[2mm]
4\delta'_N & = & g'_1(t) & j = N.
\end{cases}
\tag{2.9}
$$

Now, using equations (2.9), we obtain the following system with $N+1$ equations and $N+1$ unknowns.

$$
A\,\delta' = \Phi,
\tag{2.10}
$$

where

$$
A = \begin{pmatrix}
4 & 1 & 0 & \cdots & 0 \\
1 & 4 & 1 & & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & & 1 & 4 & 1 \\
0 & \cdots & 0 & 1 & 4
\end{pmatrix},
$$

$\delta' = \left[\delta'_0, \delta'_1, \ldots, \delta'_{N-1}, \delta'_N\right]^T$, $\Phi = \left[\Phi_0, \Phi_1, \ldots, \Phi_{N-1}, \Phi_N\right]^T$, corresponding to the knots are evaluated as:

$\Phi_0 = g'_0(t) \quad for \quad j = 0;$

$\Phi_j = -(\delta_{j-1} + 4\delta_j + \delta_{j+1})\dfrac{3}{h}(\delta_{j+1} - \delta_{j-1}) + \dfrac{6\lambda}{h^2}(\delta_{j-1} - 2\delta_j + \delta_{j+1}) \quad for \quad j = 1, \ldots, N-1,$

$\Phi_N = g'_1(t) \quad for \quad j = N.$

Now, we apply the SSP-RK54 method to solve the first order ordinary differential equation system (2.11). Once the parameter $\delta^0 = \delta(t_0)$ has been determined at a specified time level, we can compute the solution at the required time level by using iterations.

Initial vector of parameters δ^0 can be obtained by using the initial and boundary conditions at $t = 0$. We then have the following relations

$S(x_0, 0) = g_0(0) \quad for \quad j = 0;$

$S(x_j, 0) = u_0(x_j) \quad for \quad j = 1, \ldots, N-1,$

$S(x_N, 0) = g_1(0) \quad for \quad j = N.$

The above equations yield a tridiagonal matrix system by using the approximation solution (2.5) as given

$$
A\,\delta = b,
\tag{2.11}
$$

where

$$
A = \begin{pmatrix}
4 & 0 & 0 & \cdots & 0 \\
1 & 4 & 1 & & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & & 1 & 4 & 1 \\
0 & \cdots & 0 & 0 & 4
\end{pmatrix}
$$

, $\delta^0 = \left[\delta^0_0, \delta^0_1, \ldots, \delta^0_{N-1}, \delta^0_N\right]^T$, $b = \left[g_0(0), u_0(x_1), \ldots, u_0(x_{N-1}), g_1(0)\right]^T$.

We have used the Thomas algorithm to solve the tridiagonal system encountered in (2.11) and then used SSP-RK54 method to solve the ODE system.

3. Numerical Illustrations

In this section we discuss numerical testes to demonstrate the accuracy and efficiency of the proposed method. The numerical solution obtained by this method are tabulated and compared with some works in the literature for various values of x, t and the viscosity $0 < \lambda$. In order to measure the accuracy of the proposed method, we discretize the solution domain $[a, b]$ into uniformly sized length h by new equally spaced points $x'_i = a + i\frac{b-a}{k}$, for $i = 0, \ldots, k$. For our discretization with the forward Euler, the linear stability yields the restriction and given by: $CFL = \lambda \Delta t / h^2$. We adopt the relative error as defined by

$$e_\infty(k) = \frac{\max\limits_{0 \leq n \leq N} \left(\max\limits_{0 \leq i \leq k} |u(x'_i, t_n) - s_{h,n}(x'_i)| \right)}{\max\limits_{0 \leq n \leq N} \left(\max\limits_{0 \leq i \leq k} |u(x'_i, t_n)| \right)}, \tag{3.1}$$

3.1. Problem 1. We consider the exact solutions of the Burgers equation as given in [7]

$$u(x, t) = \frac{x/t}{1 + \sqrt{t/t_0} e^{x^2/4\lambda t}}, \quad t \geq 1; \quad 0 \leq x \leq 1,$$

where $t_0 = e^{1/8\lambda}$. The initial condition is obtained from the exact solution when $t = 1$. Boundary conditions are $u(0, t) = u(1, t) = 0$. The numerical solution represents shock like solution of the Burgers equation with those reported in the paper [7]. Numerical solutions of this problem the illustrate the propagation of shock for $\lambda = 5E - 03, 5E - 04, 5E - 05$ and $\lambda = 5E - 06$ at different values of the space and time steps. Figures 1a and 1b present the shock for the viscosity λ as $5E - 04$ and $5E - 03$, respectively. From these figures, we have seen the initial shocks for $\lambda = 5E - 04$ which are steep behaviour and, these sharp continues during time progression. In the same figures,the agreement between the numerical and the exact solution appears satisfactorily. So that, the behaviour of the current scheme are in very good agreement for various viscosities. And, they keep the correct physical characteristics of the problem. Figures 2a and 2b present that the numerical solutions become more steep with the smaller viscosities $\lambda = 5E - 05$ and $\lambda = 5E - 06$, respectively. Here, we have noted that the steepness remains almost unchanged as time progresses. The same figures present the effectiveness of the present scheme with the small viscosities. In Table 1, we give e_∞ errors at various values of x, t at for various spatial and time increments and, compare with some works given in the literature. We have observed that the approximate solutions are seen to be very close to the exact solutions. And also, the comparisons showed that the present scheme offers better results than the numerical schemes given in [7]. The physical behavior of the numerical solutions of the present scheme at various λ values are depicted in Figure 3.

Table 1. Comparison of the present results with the literature at $\lambda = 5E - 04$, $\Delta t = 1E - 02$ and $h = 5E - 03$.

x	t	Ref [7]	Present Method	Exact
0.1	1.7	0.058830	0.058821	0.058820
0.3	1.7	0.176480	0.176472	0.176470
0.5	2.5	0.200010	0.200004	0.200000
0.7	3.25	0.215390	0.215380	0.215380
0.9	3.25	0.123580	0.124354	0.124350

3.2. Problem 2. The exact solution for the current problem was determined in terms of the infinite series as given in [5]

$$u(x, t) = 2\pi v \frac{\sum_{j=1}^{\infty} j a_j sin(j\pi x) e^{-j^2 \pi^2 v t}}{a_0 + 2 \sum_{j=1}^{\infty} a_j cos(j\pi x) e^{-j^2 \pi^2 v t}},$$
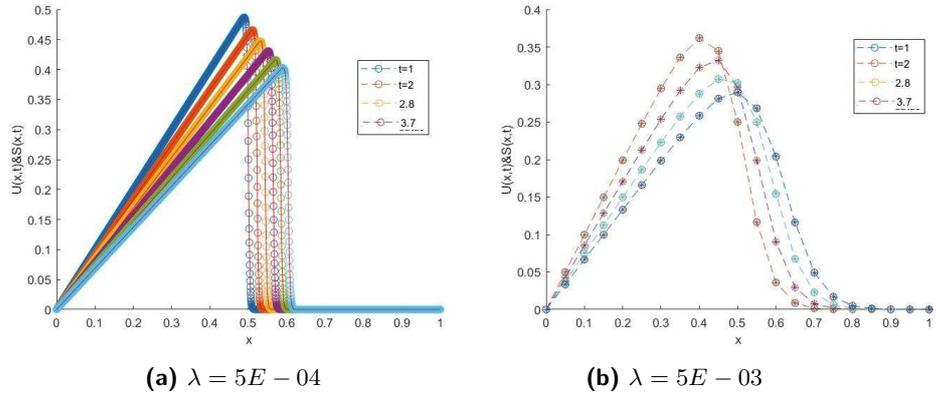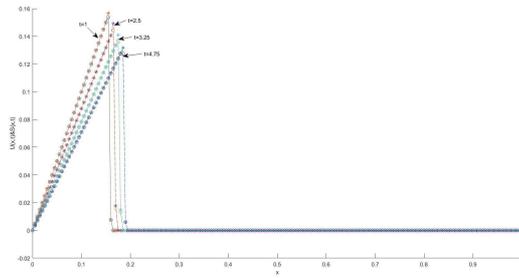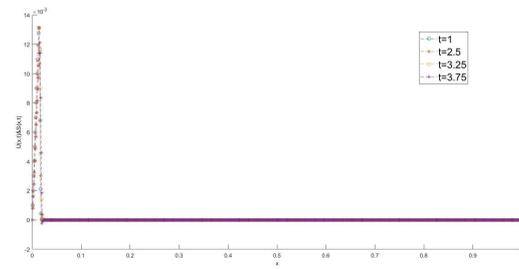
(a) $\lambda = 5E - 04$ **(b)** $\lambda = 5E - 03$

Figure 1. Numerical and exact solutions of problem 1 at different times produced for the various parameters.

(a) CFL=$1.4E - 03$, $\lambda = 5E - 05$, $\Delta t = 1.1E - 04$

(b) CFL=$1.4E - 02$, $\lambda = 5E - 06$, $\Delta t = 1.1E - 02$

Figure 2. Numerical and exact solutions of problem 1 at $t = 1.5$.

where

$$a_j = \int_0^1 e^{-(2\pi v)^{-1}(1-cos(\pi v))} cos(j\pi x) dx \quad \text{for all} \quad j \geq 1.$$
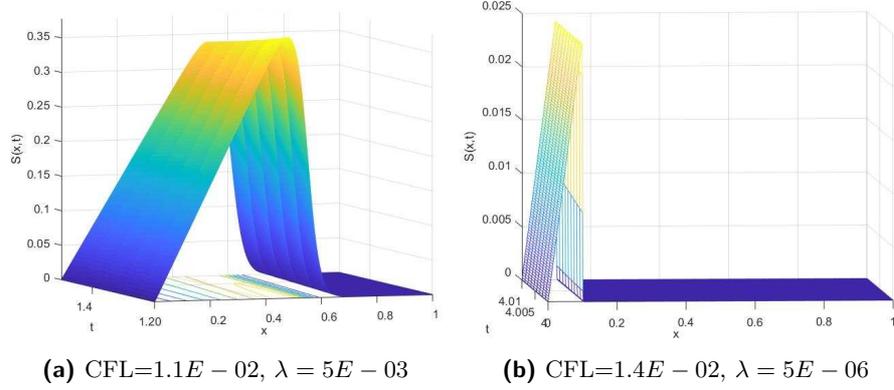
(a) CFL=$1.1E - 02$, $\lambda = 5E - 03$ **(b)** CFL=$1.4E - 02$, $\lambda = 5E - 06$

Figure 3. Physical behavior of the computed solution for problem 1 at various λ values.

The initial and boundary conditions for this problem are

$$u(0,t) = u(1,t) = 0 \text{ and } u(x,0) = sin(\pi x) \quad (x,t) \in \Omega = [0,1] \times [0,T].$$

Table 2. Comparison of the present results with the literature at $\lambda = 1$, $\Delta t = 1E - 04$ and $h = 1E - 01$.

x	Ref [6]	Ref [7]	Present Method	Exact
0.1	0.10831	0.10898	0.108180	0.10954
0.2	0.20724	0.208623	0.207097	0.20979
0.3	0.28799	0.29013	0.287882	0.29190
0.4	0.34273	0.34564	0.342730	0.37158
0.5	0.36531	0.36895	0.365519	0.35905
0.6	0.35223	0.356339	0.352663	0.30991
0.7	0.30400	0.30748	0.3039613	0.30991
0.8	0.22358	0.22606	0.2231817	0.12069
0.9	0.11860	0.11988	0.118137	0.12069

Table 3. Comparison of the present results with the literature at $\lambda = 1$, $\Delta t = 1E - 04$ and $h = 5E - 02$.

x	Ref [6]	Ref [7]	Present Method	Exact
0.1	0.10920	0.109374	0.109143	0.10954
0.2	0.20912	0.20946	0.209001	0.20979
0.3	0.29088	0.291404	0.290719	0.29190
0.4	0.34658	0.347287	0.346398	0.34792
0.5	0.36997	0.37083	0.369793	0.37158
0.6	0.35740	0.35826	0.357162	0.35905
0.7	0.30847	0.30917	0.308146	0.30991
0.8	0.22676	0.227255	0.2264421	0.22782
0.9	0.12012	0.120384	0.119929	0.12069
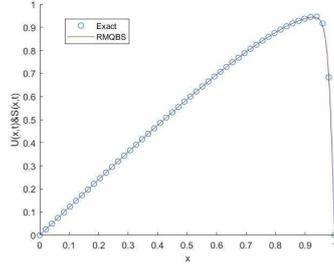
(a) $\lambda = 1E - 02$, $\Delta t = 1E - 03$, and CFL=$4E - 03$
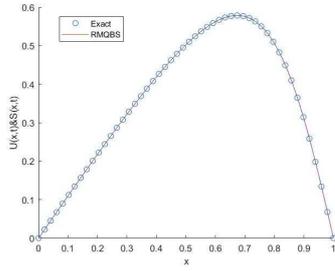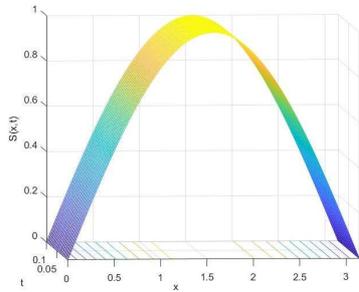

(b) $\lambda = 1E - 01$, $\Delta t = 1E - 03$ and CFL=$4E - 02$
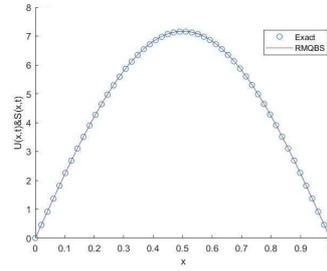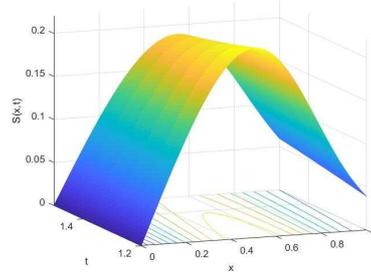

(c) $\lambda = 1$, $\Delta t = 1E - 03$ and CFL=$4E - 01$

Figure 4. Numerical and exact solutions of problem 2 at $t = 5E - 01$ produced for the various parameters.

(a) CFL=$1.1E - 02$, $\lambda = 1E - 03$

(b) CFL=$1.4E - 02$, $\lambda = 1E - 04$

Figure 5. Physical behavior of the computed solution for problem 2 at various λ and CFL values.

The presented results together with exact solutions are documented in Tables 2 and 3. It is shown that the agreement between the numerical and exact solutions appear satisfactorily. In the same tables, we compare between the current method and some previous works [6, 7]. The proposed scheme exhibits more accurate results quite than the rival methods. The numerical solutions are visualized at various values of the parameters λ, Δt and CFL in Figures 4a, 4b and 4c. The initial shock which is very steep with $\lambda = 1E - 02$. Here, it can be concluded that the numerical results are found to be in

very good agreement with the exact solution. The exact values are not practical to make comparison for the small values of λ because of slow convergence of the Fourier series result. Figure 5 presents the physical behavior of the numerical solutions of the present scheme at various λ values for difference CFL conditions. Here, we have been observed the physical behavior of this problem cannot keep their characteristic for $\lambda < 1E - 03$.

4. Conculusions and Recommendations

This article has explored the utility of a collocation scheme based on modified cubic B-spline basis functions in space with the SSPRK54 scheme in time for solving the Burgers equation. It has been observed that the currently modified techniques approximates the exact solution very well. The obtained results show that the proposed scheme is efficient and reliable for solving the Burgers equation for quite small values of the viscosity constant. The produced results have also been shown to be more accurate than some existing solutions in the literature. In a similar manner, a more detailed discussion can also carried out for more involved models.

References

[1] Ali A.H.A., Gardner L.R.T., Gardner G.A., *A collocation method for Burgers equation using cubic splines*, Computer Methods in Applied Mechanics and Engineering **100** (1992) 325–337.

[2] Bateman H., *Some recent researchers on the motion of fluids*, Monthly Weather Review **43** (1915) 163–170.

[3] Benton E., Platzman G.W., *A table of solutions of the one dimensional Burgers equations*, Quarterly of Applied Mathematics, **30** (1972) 195–212.

[4] Boor C., *A Practical Guide to Spline*, Mathematics of Computation, 27, (1978).

[5] Cole J.D., *On a quasi-linear parabolic equation occuring in aerodynamics*, Quarterly of Applied Mathematics **3** (1951) 225–236.

[6] Chakrone O., Diyer O., Sbibih D., *Numerical solution of Burgers equation based on cubic B-splines quasi-interpolants and matrix arguments,* Boletim da Sociedade Paranaense de Matemtica **33 2** (2015) 109–119.

[7] Dag I., Irk D., Sahin A., *B-Spline collocation methods for numerical solutions of the Burgers equation*, Mathematical Problems in Engineering **5** (2005) 521–538.

[8] Dag, I., Irk D., Sari M., *The extended cubic B-spline algorithm for a modified regularized long wave equation*, Chinese Physics B **22** (2013) 1–6.

[9] Jiang Z., Wang R., *An improved numerical solution of Burgers equation by cubic B-spline quasi-interpolation,* International Journal of Computing Science and Mathematics **7** (2010) 1013–800.

[10] Jain P.C., Shankar R., Singh T.V., *Numerical technique for solving convective-reaction-diffusion equation*, Mathematical and Computer Modelling **22** (1995) 113–125.

[11] Lyche T., Schumaker L., *A Multiresolution Tensor Spline Method for Fitting Functions on the Sphere*, SIAM Journal of Scientific Computing, **22** (2000) 724–746.

[12] Lyche T., *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces (Geometric Design Publications)*, Society for Industrial and Applied Mathematics, (1987).

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

[14] Morton K.W., Mayers D.F., *Numerical Solution of Partial Differential Equations*, Cambridge university press, (1994).

[15] Sari M., Tunc H., *An optimization technique in analyzing the Burgers equation*, Sigma Journal of Engineering and Natural Sciences, **35** (2017) 369–386.

[16] Prenter P.M., *Splines and vartational methods*, John wiley, New York, (1975).

[17] Schoenberg I. J., *Contribution to the problem of approximation of equidistant data by analytic functions*, Quarterly of Applied Mathematics **2** (1946) 45–99.

[18] Schumaker L., *Spline Functions and Basic Theory*, Cambridge Mathematical Library, (2007).

[19] Spiteri J.R., RuthM S.J., *A new class of optimal high-order strong stability preserving time stepping scheme*, SIAM Journal on Numerical Analysis **14** (2002) 469–491.

[20] Schultz M.H., *Spline analysis*, Prentice-Hall series in automatic computation (1973).

[21] Von Rosenberg D.U., *Methods for Solution of Partial Differential Equations*, American Elsevier Publishing Co. **113** (1969).

SHKO ALI TAHIR
YILDIZ TECHNICAL UNIVERSITY.
DEPARTMENT OF MATHEMATICS, FACULTY OF ARTS AND SCIENCE, 34220, ISTANBUL, TURKEY
  E-mail address: shko.ali.tahir@std.yildiz.edu.tr

MURAT SARI
YILDIZ TECHNICAL UNIVERSITY.
DEPARTMENT OF MATHEMATICS, FACULTY OF ARTS AND SCIENCE, 34220, ISTANBUL, TURKEY
  E-mail address: sarim@yildiz.edu.tr