



A new algorithm based on Jacobian free Newton - Krylov method and Bessel functions for viscous Burger equation

Mehran Nikarya ^{a,*}

^a Department of Electrical Engineering and Information Technology Iranian Research Organization for Science and Technology (IROST), Tehran, Iran

ARTICLE INFO

Article history:

Received: 2024-10-05

Accepted: 2025-06-17

Keywords:

Nonlinear viscous Burger equation
Bessel function of the first kind
Spectral collocation method
Jacobian free Newton method

ABSTRACT

The Burgers' equation is a fundamental nonlinear partial differential equation (PDE) that describes various phenomena in fluid dynamics, nonlinear acoustics, Optics and Photonics, Wave Propagation, and traffic flow. Actually, viscous Burger equation is a simplified version of famous Navier-Stokes equation for incompressible fluid, without pressure gradient and external forces. In this paper a new method will be introduced to solve this nonlinear partial differential equation based on Bessel functions and the Jacobian free Newton method combined with generalized minimum residual (JFNGMRes) method. In this work the nonlinear Burger equation is converted to a nonlinear system of algebraic equations without any linearization and discretization methods. Finally, by using JFNGMRes, the solution of this nonlinear system will be achieved. To show the reliability and applicability of the proposed method, we solve some examples of the 1D and the 2D nonlinear Burger equation and then we compare our results with some famous methods in other papers and exact solutions.

1. Introduction

Solving nonlinear partial differential equations (PDE) is a challenge among engineers, numerical and mathematics researchers. On the other hand, many arisen problems in sciences, engineering, finance, ecology, chemistry, etc. are defined as nonlinear PDEs; thus, many mathematicians, numerical analysts and computer scientists have tried to solve these problems by different methods and algorithms. For example Adomian decomposition method [1, 2, 3], homotopy analysis method [4], VIM [5, 6, 7], mesh free methods [8, 9, 10], combination of spectral methods with other methods like linearization and time discretization [11, 12, 13, 14, 15], spectral domain decomposition [16, 17] Taylor series [18], differential quadrature [19], finite difference [20, 21, 22, 23], finite element [24, 25, 43, 26], multi grid methods [27, 28], etc.

In solving nonlinear ordinary differential equations (ODEs) and integral-differential equations, spectral methods are one of the

high accuracy numerical tools in computational and numerical algorithms [29, 30, 31]. But spectral methods have not been used directly and without discretization and linearization to solve nonlinear PDEs, widely. Spectral methods convert the solving procedure of a nonlinear differential equation to solving of a nonlinear system of algebraic equations. In fact, the major difficulty of spectral methods is solving this system of nonlinear equations. One of the best methods to solve a nonlinear system of equations is the Jacobian free Newton-Krylov sub-space methods (JFNGMRes) [32,33].

Now, in this paper, we introduce a new method to solve nonlinear Burger equation by using the spectral collocation method based on the first kind Bessel functions and JFNGMRes method with adaptive preconditioner for the first time.

The rest of this paper is organized as follows: In Section 2, the viscous Burger equation is illustrated. The basic definitions and properties of the first kind of Bessel functions, function approximation based on Bessel function and spectral methods with to solve nonlinear PDE are presented in Section 3. In Section 4 the JFNGMRes with

* Corresponding author.

E-mail address: mehran.nikarya@gmail.com

adaptive preconditioner is described. Then to show the advantages, applicability and reliability of the proposed method we solve some examples of Burger's equation and compare our results with others in Section 5. Finally, the paper concludes in Section 6.

1.1. Introducing Burger's Equation

For the past century, the Burger equation has been subject to a huge number of studies in mathematics and physical sciences. In particular, the Burger equation has been applied to the study of traffic flow, acoustic transmission, shock waves in a viscous medium, gas dynamics, waves in fluid filled viscous elastic tubes and supersonic flow around airfoils [34]. The Viscous Burger equation is a simplified version of famous Navier-Stokes equation for incompressible fluid, without pressure and external forces, the 2D Burger equation is defined as follows [35, 36]:

$$\begin{aligned} u_t + uu_x + vu_y &= \nu(u_{xx} + u_{yy}) \\ v_t + uv_x + vv_y &= \nu(v_{xx} + v_{yy}) \end{aligned} \quad (1)$$

with initial conditions:

$$u(x, y, 0) = f(x, y), v(x, y, 0) = g(x, y), (x, y) \in \Lambda$$

and Dirichlet boundary conditions. Where $0 < \nu \leq 1$ is the coefficient of kinematic viscosity, f and g are known functions u and v are the velocity components in x and y directions, respectively. Also, u_t and v_t are unsteady terms, $uu_x + vu_y$ and $uv_x + vv_y$ are convective terms and $\nu(u_{xx} + u_{yy})$ and $\nu(v_{xx} + v_{yy})$ are viscous terms.

Also the one dimensional Burgers equation is defined as follows:

$$u_t - \nu u_{xx} + uu_x = 0, \quad (x, t) \in \Lambda \times (0, T], \quad (2)$$

with initial condition

$$u(x, 0) = f(x), \quad 0 \leq x \leq 1, \quad (3)$$

and boundary condition

$$u(0, t) = g_1(t), \quad u(1, t) = g_2(t), \quad 0 \leq t \leq T \quad (4)$$

Solving and finding the solution of the Burger equation has been studied for the last half century and still it is an active area of research to develop some better numerical algorithms and methods to approximate its solution. Many researchers solved this problem by using several numerical methods: finite difference methods [20, 21, 22, 37, 38, 39, 40], moving mesh in discontinues Galerkin method [41], finite elements methods [24, 25, 42, 43] an many other methods [3, 12, 13, 18, 19, 44, 45, 46, 47, 48].

2. Function Approximation and Spectral Methods Based on Bessel Functions

Definition 1: Let $\Lambda = \{x | a < x < b\}$, then nonnegative function $\omega(x)$ is a weight function if:

- $\omega(x) \geq 0$ is measurable on finite or infinite interval Λ .
- $\mu_k = \int_{\Lambda} x^k \omega(x) dx < \infty$ for $k = 0, 1, \dots$.
- $\int_{\Lambda} \omega(x) dx > 0$.

Let ω is a certain weight function, therefore:

$$L_{\omega}^2(\Lambda) = \{v | v \text{ is measurable and } \|v\|_{\omega} < \infty\}$$

Where

$$\|v\|_{\omega} = (v, v)_{\omega}^{\frac{1}{2}} = \left(\int_{\Lambda} |v(x)|^2 \omega(x) dx \right)^{\frac{1}{2}}$$

In particular, in Hilbert space $L_{\omega}^2(\Lambda)$, $(u, v)_{\omega} = \int_{\Lambda} u(x)v(x)\omega(x)dx$ is inner product. If $\omega = 1$ can be omitted, and $\partial_x^k v = \frac{d^k v}{dx^k}$. Now for nonnegative integer m , we define

$$H_{\omega}^m(\Lambda) = \{v | \partial_x^k v \in L_{\omega}^2(\Lambda), 0 \leq k \leq m\}$$

with following norm, and semi-norm

$$\begin{aligned} (u, v)_{m, \omega} &= \sum_{k=0}^m (\partial_x^k u, \partial_x^k v)_{\omega}, \\ |v|_{m, \omega} &= \| \partial_x^m v \|_{\omega}, \quad \|v\|_{m, \omega} = (v, v)_{m, \omega}^{\frac{1}{2}}, \end{aligned}$$

For any real $r \geq 0$, we define the space $H_{\omega}^r(\Lambda)$ by the space interpolation as in Adams [49]. In this paper we use the Bessel functions of the first kind $J_n(x)$ as the basis functions of $L^2(\Lambda)$:

$$J_n(x) = \sum_{r=0}^{\infty} \frac{(-1)^r}{r! \Gamma(n+r+1)} \left(\frac{x}{2}\right)^{2r+n}, \quad (5)$$

where $\Gamma(\lambda)$ is the gamma function. Series (5) is convergent for all $-\infty < x < \infty$ and is a solution of the below Sturm-Liouville equation [50, 51]:

$$\begin{aligned} x^2 y''(x) + xy'(x) + (x^2 - n^2)y(x) &= 0, \\ \text{or } x \in (-\infty, \infty), \quad (n \in \mathbb{R}) \end{aligned} \quad (6)$$

It is clear that, $J_n(x)$ are linear independent if n be integer and also $\int_{\Lambda} J_n(x) < \infty$ and $\|J_n(x)\| < \infty$ [50], therefor $J_n(x), n \in \mathbb{N}$ is a basis of $L^2(\Lambda)$. Lemma 1: one of the best relations of J'_n is:

$$J'_n(x) = \frac{1}{2}J_{n-1}(x) - \frac{1}{2}J_{n+1}(x) \quad (7)$$

Proof: by deriving (5) and using expansions of $J_{n-1}(x)$ and $J_{n+1}(x)$, the result is desirable.

Remark 1: the derivative operational matrix of the first kind Bessel functions can be obtained as follow:

let $J = [J_0(x), J_1(x), J_2(x), \dots, J_n(x)]^T$ therefor $J' \approx D J$, where D is the derivative operational matrix and is obtained by using (7) as follows:

$$D = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{2} & 0 & -\frac{1}{2} \\ a_0 & a_2 & a_3 & \cdots & \cdots & a_{n-1} & a_n \end{bmatrix} \quad (8)$$

where the $a_0, a_1, a_2, \dots, a_n$ will be obtained by an interpolation technique.

Now, let N be a positive integer, we define finite subspace

$$\mathcal{J}_N = \text{span}\{J_0, J_1, \dots, J_N\},$$

this is clear that $\mathcal{J}_N(\Lambda) \subseteq H^r(\Lambda) \subseteq L^2(\Lambda)$.

Now consider $L^2(\Lambda)$ -orthogonal projection $P_N: H^r(\Lambda) \rightarrow \mathcal{J}_N$ that, for any $v \in H^r(\Lambda)$:

$$(P_N v - v, \phi) = 0, \quad \forall \phi \in \mathcal{J}_N.$$

or equivalently,

$$P_N v(x) = \sum_{i=0}^N \hat{v}_i J_i(x).$$

Therefore, we can write:

$$(\sum_{i=0}^N \hat{v}_i J_i(x) - v, \phi) = 0, \quad \forall \phi \in \mathcal{J}_N.$$

so, for any $v \in H^r(\Lambda)$ and $r \geq 0$ we have [52]:

$$\|P_N v - v\| \leq c N^{-r} \|v\|_r, \quad (9)$$

Several papers have discussed about convergence of the spectral methods to solve the nonlinear Burger equation [53, 54, 55]. In a same way we can write about the convergence of the proposed method to solve the nonlinear Burger equation as follows.

Remark 2: assume that the $u \in L^2(0, T; H^r(\Lambda))$, $r > 1$, is the solution of the Burger equation (2), and the $u_N = P_N u \in \mathcal{J}_N(\Lambda) \times \mathcal{J}_N(0, T)$ is the approximation of u , then:

$$\|u - u_N\|_{H^r(\Lambda) \times L^2(0, T)} \leq C N^{-r},$$

$$\|u\|_{H^r(\Lambda) \times L^2(0, T)} = \int_0^T \|u\|_r^2 dt,$$

and C is a positive constant depending only on the norms of u in the spaces mentioned [52, 56, 57]. We know that, the $\mathcal{J}_N(\Lambda) \times \mathcal{J}_N(0, T)$ is a finite subspace of $H^r(\Lambda) \times L^2(0, T)$, according to the assumption $u \in H^r(\Lambda) \times L^2(0, T)$, the $\|u\| = \int_0^T \|u\|_r^2 dt < \infty$, therefore, there is a $N_0 \in \mathbb{N}$ that for any $N > N_0$, N^r be bigger

than $\|u\|_r$ or equivalently C . So, if N be large enough, we can say: if $N \rightarrow \infty$ then $\|u - u_N\|_{H^r(\Lambda) \times L^2(0, T)} \rightarrow 0$, hence, the error of this approximation by increasing N will be decreased, in numerical examples this principle will be shown. All of above relations can be extended for the 2D time dependent Burger equation.

3. Spectral Collocation Method

Spectral methods, in the context of numerical schemes for solving differential equations, generically belong to the family of weighted residual methods (WRMs) [58]. WRMs represent a particular group of approximation techniques, in which the residuals (or errors) are minimized in a certain way and thereby leading to specific methods including Galerkin, Petrov-Galerkin, collocation and tau formulations. consider the approximation of the following problem via spectral method:

$$\mathcal{L}u(x, t) + \mathcal{N}u(x, t) = f(x, t), \quad (x, t) \in \Lambda \times [0, T], \quad (10)$$

where $\mathcal{L}$ is the differential or integral operation, $\mathcal{N}$ is a lower-order linear and/or nonlinear operator involving only derivatives (if exist) and $f(x, t)$ is a function of variables x and t , with enough initial and boundary conditions.

The starting point of the spectral methods is to approximate the solution $u(x, t)$ by a finite summation:

$$u(x, t) \approx u_{M, N}(x, t) = \sum_{i=0}^N \sum_{j=0}^M a_{i, j} \phi_i(x) \phi_j(t), \quad (11)$$

$$x \in \Lambda, t \in [0, T]$$

where ϕ_n 's are the basis functions that we have chosen $J_n(x)$ as basis function and the expansion's coefficients must be determined. Substituting u with $u_{M, N}$ in (10) leads to the residual function:

$$\mathcal{R}_{M, N}(x, t) = \mathcal{L}u_{M, N}(x, t) + \mathcal{N}u_{M, N}(x, t) - f(x, t) \quad (12)$$

The notion of the WRM is to force the residual function to zero by requiring:

$$\langle \mathcal{R}_{M, N}, \psi_{1i} \rangle_{\omega_1} \langle \psi_{2j} \rangle_{\omega_2} = 0 \quad (13)$$

$$\equiv \int_0^T \int_{\Lambda} \mathcal{R}_{M, N}(x, t) \psi_{1i}(x) \psi_{2j}(t) \omega_1(x) \omega_2(t) dx dt = 0, \quad (14)$$

$$1 \leq i \leq N \quad 1 \leq j \leq M,$$

where $\{\psi_k\}$ are test functions, and ω is positive weight function. The choice of test functions results to a kind of the spectral methods. A method for forcing the residual function (12) to zero, is the collocation algorithm [51, 59]. In this method, by choosing the Lagrange basis polynomials constructed on the collocation points t_p and x_q as the test function, such that $\psi_{1i}(x_q) = \delta_{iq}$ and $\psi_{2j}(t_p) = \delta_{jp}$ and applying the Gauss quadrature rule can write:

$$\begin{aligned} & \ll \mathcal{R}_{M,N}, \psi_{1_i} >_{\omega_1}, \psi_{2_j} >_{\omega_2} = \\ & \sum_{p=1}^K \sum_{q=1}^L \mathcal{R}_{M,N}(x_q, t_p) \psi_{1_i}(x_q) \psi_{2_j}(t_p) \omega_1(x_q) W_q \omega_2(t_p) W_p = 0 \quad (15) \\ & 0 \leq i \leq N \quad 0 \leq j \leq M \end{aligned}$$

by setting $\omega_1 = \omega_2 = 1$, the Eq. (15) becomes:

$$\mathcal{R}_{M,N}(x_q, t_p) = 0, p = 0, 2, \dots, N, q = 0, 2, \dots, M \quad (16)$$

Now, we have $(N+1) \times (M+1)$ nonlinear equation and $(N+1) \times (M+1)$ unknown coefficients. For the 2D case the summation (11) becomes:

$$u_{M,N,L}(x, y, t) = \sum_{i=0}^N \sum_{j=0}^M \sum_{k=0}^L a_{i,j,k} \phi_i(x) \phi_j(y) \phi_k(t) \quad (17)$$

and the other relation can be extended to the 2D case.

4. Newton - Krylov Aagorithm

Solving a nonlinear differential equation by spectral method directly (without linearization or discretization) leads to solving a large nonlinear system of equations $F(x) = 0$, where $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a function $F(x) = (f_1(x), f_2(x), f_3(x), \dots, f_n(x))^T$ and $x \in \mathbb{R}^n$ is a vector. So, speed and accuracy of solving this nonlinear system is very important. Many works have been done to improve solving the nonlinear systems [32, 33, 60, 61, 62]. One of the best methods to solve a nonlinear system is the classical Newton iterative method:

$$F(x_{n+1}) = F(x_n) + (x_n - x_{n+1})F'(x_n) \quad (18)$$

presuppose x_{n+1} be root of $F(x)$, $F(x_{n+1}) = 0$:

$$F(x_n) + (x_{n+1} - x_n)F'(x_n) = 0 \quad (19)$$

$$\Rightarrow x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)} \quad (20)$$

$F'(x) = J(x)$ is the $n \times n$ Jacobian matrix and is defined as follows:

$$J_{ij} = \left(\frac{\partial f_i}{\partial x_j} \right) \quad (21)$$

therefore:

$$x_{n+1} = x_n - J(x_n)^{-1} F(x_n). \quad (22)$$

In fact, in each iteration, a linear system must be solved:

$$\begin{cases} x_{n+1} = x_n + \delta x_n \\ J(x_n) \delta x_n = F(x_n). \end{cases} \quad (23)$$

The obtained nonlinear systems of spectral methods are usually large and ill-condition and for large and complicated nonlinear systems, calculation and reordering $J(x_n)$ and solving obtained linear system in each iteration could be most time consuming. One good idea is to use the finite difference technique to approximate the Jacobian-vector product:

$$J(x_n) v_n \simeq \frac{f(x_n + \varepsilon v_n) - f(x_n)}{\varepsilon}, \quad (24)$$

where ε is a very small value. Jacobian-vector product, can be useful to approximate the Jacobian matrix. Also, for large dimensions, iterative methods such as GMRes or BiCGSTAB are preferred over direct solvers [32, 33]. In this paper, a Jacobian-free Newton GMRes (JFNGMRes) with an adaptively preconditioner have been used to solve large nonlinear system of equations. In this method, a preconditioner be updated in both Newton iterations and GMRes iterations. Many researchers have used Newton-Krylov methods and Jacobian free Newton-krylov methods to solve several nonlinear problems [32, 33, 63, 64, 65].

4.1. Algorithm of Preconditioned JFNGMRES

In this section we describe the Jacobian free Newton's method alongside generalized minimum residual method with adaptive preconditioner to solve the nonlinear system that is obtained from solving Burger equation via proposed:

1. Set $k = 0$ (k is the iteration counter of the Newton method), choose an initial guess x_0 .
2. Select a nonsingular Matrix $M_0 = \eta I$ as the preconditioner, where $\eta \in \mathbb{R}$ and $I = \text{diag}(1, \dots, 1) \in \mathbb{R}^{n \times n}$.
3. Begin of Newton's iterations: Repeat until $\|F(x_k)\|_2 < \text{err}$.
4. Use GMRes method with Jacobian free formula to solve linear system $J(x_k) \delta_k = F(x_k)$.

$$4.1. r_0 = F(x_k) - \frac{F(x_k + \varepsilon x_k) - F(x_k)}{\varepsilon}, \beta = \|r_0\|_2, v_1 = \frac{r_0}{\beta} \text{ and } M_{in}^0 = M_k$$

4.2. for $j = 1$ to N do

$$4.2.1. v_j = M_k \cdot v_j.$$

$$4.2.2. \Delta F_{in}^j = F(x_k + \varepsilon v_j) - F(x_k)$$

$$4.2.3. \Delta x_{in}^j = \varepsilon v_j.$$

$$4.2.4. v_{j+1} = \frac{\Delta F_{in}^j}{\varepsilon}.$$

4.2.5. Update the preconditioner matrix:

$$M_{in}^{j+1} = M_{in}^j + \frac{(\Delta x_{in}^j - M_{in}^j \cdot \Delta F_{in}^j) \Delta x_{in}^{jT} \cdot M_{in}^j}{\Delta x_{in}^{jT} \cdot M_{in}^j \cdot \Delta F_{in}^j}.$$

4.2.6. for $i = 1$ to j do

$$h_{ij} = \langle v_{j+1}, v_i \rangle.$$

$$v_{j+1} = v_{j+1} - h_{ij} \cdot v_i.$$

end for

4.2.7. $h_{j+1,j} = \|v_{j+1}\|_2$.

4.2.8. if $h_{j+1,j} = 0$ then goto step 4.c else

$$v_{j+1} = \frac{v_{j+1}}{h_{j+1,j}}.$$

end for (4.2.)

4.3. Now the Hessenberg matrix $\bar{H}_{N+1 \times N} = h_{ij}$ is constructed, square Hessenberg matrix $H_{N \times N}$ is obtained by elimination of the last row of $\bar{H}$.

4.4. $V_{N \times N} = [v_1, v_2, \dots, v_N]$.

4.5. Now the least-square linear problem $y_k = \min \| \rho e_1 - Hy \|_2$ must be solved, where $e_1 = [1, 0, 0, \dots, 0]^T$.

4.5.1. Using N Givens rotations $G = g_1, g_2, \dots, g_N$, matrix H becomes an upper trigonal matrix.

4.5.2. Now set $R = GH$ and $W = \beta G e_1$.

4.5.3. By using simple backward method, the $Ry_k = W$ can be solved.

4.5.4. $\delta_k = Vy_k$

4.6. Set $M_k = M_{in}^{j+1}$

5. $x_{k+1} = x_k - \delta_k$

6. Compute $F(x_{k+1})$ and $\Delta F_k = F(x_{k+1}) - F(x_k)$

7. Update the preconditioner matrix M_k

$$M_{k+1} = M_k + \frac{(\Delta x_k - M_k \Delta F_k)(M_k^T \Delta x_k)^T}{\Delta x_k^T M_k \Delta F_k}$$

8. $k = k + 1$

end repeat (3).

In [32, 33] inner and outer preconditioners M_{in}^l and M_k and how to help it to Newton-Krylov method have been discussed.

5. Solving Some Examples of the Nonlinear Burger Equation

Now, in this section we want to apply the proposed method to solve some examples of Burger's equation. To show the reliability and applicability of the proposed method, we compare our results with others. In all of the following examples, we use the roots of Chebyshev polynomial $T_n(x)$ as collocation points and vector $[0, 0, \dots, 0]^T$ as initial guess of the iterative JFNGMRes method. Also, we use the RMS error in this article:

$$RMS = \sqrt{\frac{\sum_{i=1}^K (u_i^{\text{exact}} - u_i^{\text{approx}})^2}{K}},$$

5.1. Example 1:

Consider the equation (2) with conditions:

$$u(x, 0) = \frac{2v\pi \sin(\pi x)}{2 + \cos(\pi x)}, \quad (25)$$

$$0 \leq x \leq 1, \quad u(0, t) = u(1, t) = 0, \quad t \geq 0.$$

The exact solution of this equation is given by Wood [66] as follows:

$$u(x, t) = \frac{2ve^{-\pi^2 vt} \sin \pi x}{2 + e^{-\pi^2 vt} \cos(\pi x)}, \quad (26)$$

This example solved by Bernstein polynomial and DQ method for space discretization and fourth order Runge-Kutta algorithm [19] and Asaithambi solved this problem by using automatic differentiation [18]. Now for solving this problem by proposed method, we approximate the solution by using Bessel functions as follows:

$$u_{M,N} = u(x, 0) + \sum_{i=1}^{N+1} \sum_{j=1}^{M+1} a_{ij} J_i(x) J_j(t)$$

In Table 1 the general information about solving example 1 by Bessel function collocation (BFC) method and JFNGMRes is presented $v = 1, 0.1, 0.01, 0.001$ and 0.00001 , in this table IT is the number of Newton method iterations. To show the power of the proposed method, in Table 2 we have compared results of the proposed method with results of the BPDQM [19] for $v = 0.001$, and the convergence rate of proposed method to solve this example is shown in Table 3. Also, the graphs of approximated solutions for several v values are presented in Fig. 1.

Table 1. The General Information of the Proposed Method to Solve Example 1.

v	M	N	RMS	IT
0.00001	6	7	7.1e-10	1
0.0001	8	9	3.2e-9	2
0.001	12	11	8.6e-8	2
0.01	12	16	1.5e-7	3
0.1	13	17	3.4e-7	3
1	13	27	2.6e-6	4

Table 2. Comparison of RMS Error of the Proposed Method and BPDQM[19] for Example 1 with $N=0.001$.

t	BFC-JFN	BPDQM[19]
0.5	6.60e-10	8.15e-5
1	3.77e-10	5.23e-5
2	3.32e-9	4.11e-5
3	2.85e-9	2.36e-5

Table 3. Convergence Rate of the Proposed Method to Solve Example 1 for $N=0.0001, 0.001, 0.01$ and 0.001 .

$v = 0.0001$			$v = 0.001$			$v = 0.01$			$v = 0.1$		
M	N	RMS	M	N	RMS	M	N	RMS	M	N	RMS
4	4	3.12e-3	5	5	9.47e-3	6	8	6.46e-3	7	10	3.21e-3
5	6	1.25e-5	8	7	1.05e-5	8	10	9.96e-4	9	12	9.16e-4
6	7	6.75e-7	10	9	8.21e-6	10	13	8.27e-5	11	14	5.31e-5
8	9	3.21e-9	12	11	2.69e-8	12	16	3.41e-7	13	17	3.48e-7

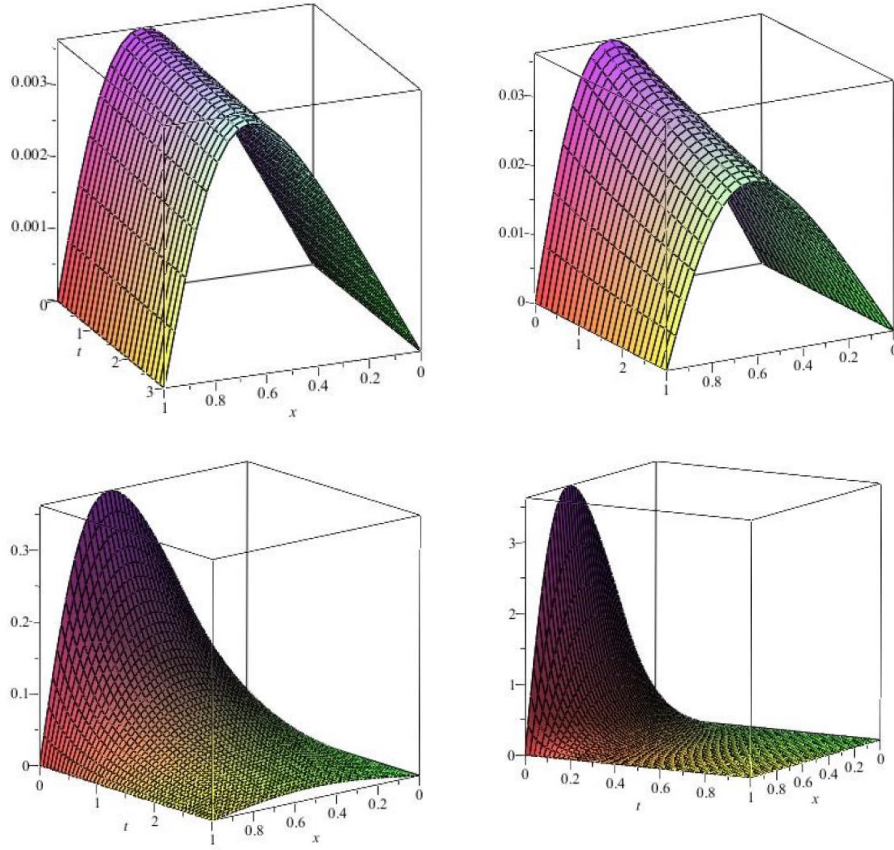


Figure 1. The graphs of obtained solutions of example 1 via the proposed method for several v .

5.2. Example 2:

Consider the one-dimensional Burger's equation (2) with initial condition:

$$u(x, 0) = 0.1 + 2v \tanh(x) \quad (27)$$

and boundary conditions:

$$u(0, t) = 0.1 + 2v \tanh(-0.1t), \quad (28)$$

$$u(1, t) = 0.1 + 2v \tanh(1 - 0.1t). \quad (29)$$

A semi-analytic solution of this example is obtained by Soliman [48] as follows:

$$u(x, t) = 0.1 + 2v \tanh(x - 0.1t). \quad (30)$$

Table 4. The General Information of the Proposed Method to Solve Example 2.

v	M	N	RMS	IT
0.00001	10	7	$1.2e - 10$	1
0.0001	11	8	$1.2e - 8$	2
0.001	13	9	$8.6e - 7$	3
0.01	14	9	$3.8e - 6$	4

Table 5. Comparison of the Proposed Method, BPDQ[19] and METF[48] for Example 2 with $N=0.001$.

x	t	JFN-BFC	METF[48]	BPDQ[19]
0.25	0.5	0.1003940	0.1003947	0.10004
	1	0.1002970	0.1002977	0.10031
	1.5	0.1001982	0.1001993	0.10023
	2	0.1000991	0.1000999	0.10018
	3	0.0999008	0.0999000	----
0.5	0.5	0.1008433	0.1008437	0.10084
	1	0.1007585	0.1007598	0.10076
	1.5	0.1006700	0.1006727	0.10067
	2	0.1005801	0.1005826	0.10059
	3	0.1003921	0.1003947	----
0.75	0.5	0.1012070	0.1012087	0.10121
	1	0.1011401	0.1011433	0.10114
	1.5	0.1010703	0.1010740	0.10107
	2	0.1010000	0.1010010	0.10100
	3	0.1008404	0.1008437	----

This example has been solved by some researchers such as, Soliman [48] by using modified extended tanh-function method and Mittal [19] by using Berestain polynomial DQ and Runge-Kutta[19]. Now, according to the conditions of problem we use the proposed method with following expansion:

$$u_{M,N} = u(x, 0) + \sum_{i=0}^N \sum_{j=1}^{M+1} a_{ij} J_i(x) J_j(t)$$

In Table 4 the general information about solving this example by the proposed method are shown for several ν . Table 5 compares results of the proposed method with results of BPDQM [19] and METF [48] for $\nu = 0.001$. Also, to show the convergence rate of proposed method to solve this example for several ν is shown in Table 6.

Table 6. Convergence Rate of the Proposed Method to Solve Example 2 for $N=0.00001, 0.0001$ and 0.001 .

$\nu = 0.0001$			$\nu = 0.001$			$\nu = 0.01$		
M	N	RMS	M	N	RMS	M	N	RMS
5	4	1.32e-4	5	5	1.27e-4	7	5	3.90e-3
6	5	2.65e-6	7	6	3.50e-5	9	7	2.36e-4
8	6	8.71e-8	9	7	2.81e-6	11	8	5.21e-5
10	7	8.63e-10	11	8	1.22e-8	13	9	8.63 -7

5.3. Example 3:

Consider the Burger's equation (2) with conditions:

$$u(x, 0) = \sin(\pi x), \quad 0 \leq x \leq 1, \quad u(0, t) u(1, t) = 0, \quad t \geq 0. \quad (31)$$

The exact solution of this problem is obtained by Hopf-Cole transformation:

$$u(x, t) = \frac{2\pi\nu \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2\nu t} n \sin(n\pi x)}{A_0 + \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2\nu t} n \cos(n\pi x)}$$

where:

$$A_0 = \int_0^1 e^{2\pi\nu^{-1}(1-\cos(\pi x))} dx$$

$$A_n = 2 \int_0^1 e^{2\pi\nu^{-1}(1-\cos(\pi x))} \cos(n\pi x) dx$$

Table 7. The General Information of Solving Example 3 VIA the Proposed Method.

ν	M	N	RMS	IT
1	20	12	5.21e - 7	1
0.1	29	11	3.36e - 6	3
0.01	22	18	8.67e - 5	4

According to the conditions of problem we use following expansion:

$$u_{M,N} = \sin(\pi x) + \sum_{i=1}^{N+1} \sum_{j=1}^{M+1} a_{ij} J_i(x) J_j(t),$$

and the condition $u(1, t)$ is satisfied in nonlinear system of equations. Now, the general information of solving example

3 by the proposed method is presented for $\nu = 1, 0.1, 0.01$ in Table 7. Table 8 compares results of the proposed method with results of BPDQM [19] and exact solution for $\nu = 1$. Also, the solutions of this example for $0.1, 0.01$ is presented in Table 9 and 10 respectively, these tables compared results of the proposed method with Haar wavelet quasilinearization [11], least-squares quadratic B-spline finite element method [43], explicit Fourth-order finite difference method [25] and exact-explicit finite difference methods [20]. The convergence rate of proposed method to solve this equation for $\nu = 0.01, 0.1$ and 1 is shown in Table 11. The graphs of obtained approximated solution are presented in Fig. 2, for $\nu = 1, 0.1, 0.01$.

Table 8. Comparison of Exact Solution, Results of BPDQ [19] and the Proposed Method to Solve Example 3 for $N=1$.

x	t	JFN-BFC	Exact	BPDQ [19]
0.25	0.4	0.01357	0.01357	0.01363
	0.6	0.00189	0.00189	0.00190
	0.8	0.00026	0.00026	0.00026
	1	0.00004	0.00004	0.00003
	3	0.00000	0.00000	0.00000
0.5	0.4	0.01923	0.01923	0.01933
	0.6	0.00267	0.00267	0.00269
	0.8	0.00037	0.00037	0.00037
	1	0.00005	0.00005	0.00005
	3	0.00000	0.00000	0.00000
0.75	0.4	0.01363	0.01363	0.01369
	0.6	0.00189	0.00189	0.00190
	0.8	0.00026	0.00026	0.00026
	1	0.00004	0.00004	0.00003
	3	0.00000	0.00000	0.00000

Table 9. Comparison of Exact Solution, Results of [11, 20, 25] and the Proposed Method to Solve Example 3 for $N=0.1$.

x	t	JFN-BFC	FD[20]	FE [25]	HWQ[11]	exact
0.25	0.4	0.30889	0.30891	0.31429	0.30887	0.30889
	0.6	0.24074	0.24075	0.24373	0.24070	0.24074
	0.8	0.19568	0.19568	0.19758	0.19566	0.19568
	1	0.16256	0.16257	0.16391	0.16255	0.16256
	3	0.02720	0.02720	0.02743	0.02721	0.02720
0.5	0.4	0.56963	0.56964	0.57636	0.56956	0.56963
	0.6	0.44721	0.44721	0.45169	0.44715	0.44721
	0.8	0.35924	0.35924	0.36245	0.35920	0.35924
	1	0.29192	0.29192	0.29437	0.29188	0.29192
	3	0.04021	0.04021	0.04057	0.04022	0.04021
0.75	0.4	0.62544	0.62542	0.62592	0.62540	0.62544
	0.6	0.48722	0.48721	0.49034	0.48716	0.48722
	0.8	0.37393	0.37392	0.37713	0.37389	0.37393
	1	0.28747	0.28748	0.29016	0.28743	0.28747
	3	0.02977	0.02977	0.01334	0.02978	0.02977

Table 10. Comparison of Exact Solution, Results of [11, 20, 43] and the Proposed Method to Solve Example 3 for $N=0.01$.

x	t	JFN-BFC	FE [43]	FD [20]	HWQ [11]	exact
0.25	0.4	0.34191	0.34819	0.34244	0.34184	0.34191
	0.6	0.26896	0.27536	0.26905	0.26891	0.26896
	0.8	0.22148	0.22752	0.22145	0.22143	0.22148
	1	0.18819	0.19375	0.18813	0.18815	0.18819
	3	0.07511	0.07754	0.07509	0.07510	0.07511
0.5	0.4	0.66071	0.66543	0.67152	0.66060	0.66071
	0.6	0.52942	0.53525	0.53406	0.52932	0.52942
	0.8	0.43914	0.44526	0.44143	0.43905	0.43914
	1	0.37442	0.38047	0.37568	0.37436	0.37442
	3	0.15018	0.15362	0.15020	0.15017	0.15018
0.75	0.4	0.91026	0.91201	0.94675	0.91026	0.91026
	0.6	0.76724	0.77132	0.78474	0.76719	0.76724
	0.8	0.64740	0.65254	0.65659	0.64745	0.64740
	1	0.55605	0.56157	0.56135	0.55608	0.55605
	3	0.22481	0.22874	0.22502	0.22504	0.22481

Table 11. Convergence Rate of the Proposed Method to Solve Example 3 for $N=1,0.1$ and 0.01 .

$\nu = 1$			$\nu = 0.1$			$\nu = 0.01$		
M	N	RMS	M	N	RMS	M	N	RMS
10	8	2.33e-3	20	7	6.38e-3	7	5	3.90e-2
14	9	6.84e-4	22	8	3.25e-4	9	7	2.36e-3
17	10	4.90e-5	26	10	2.95e-5	11	8	5.21e-4
20	12	5.21e-7	29	11	3.36e-6	22	18	8.67e-5

5.4. Example 4:

Consider the Burger's equation (2) with conditions:

$$u(x, 0) = 4x(1 - x), \quad 0 \leq x \leq 1, \quad u(0, t) = u(1, t) = 0, \quad t \geq 0 \quad (32)$$

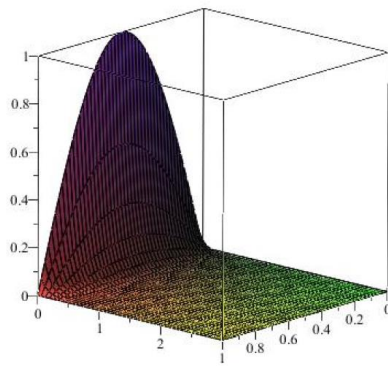
The exact solution of this equation by using Hopf-Cole transformations is as follow:

$$u(x, t) = \frac{2\pi\nu \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2\nu t} n \sin(n\pi x)}{A_0 + \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2\nu t} n \cos(n\pi x)}$$

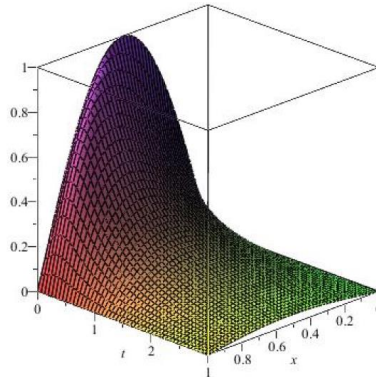
where:

$$A_0 = \int_0^1 e^{-(3\nu)^{-1}(3x^2-2x^3)} dx$$

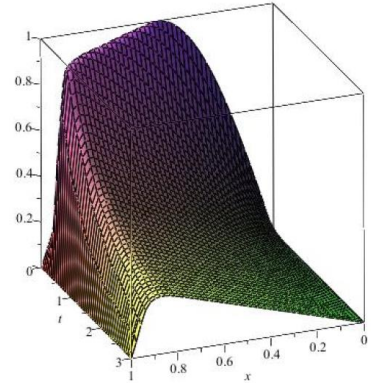
$$A_n = \int_0^1 e^{-(3\nu)^{-1}(3x^2-2x^3)} \cos(n\pi x) dx$$



a) $\nu = 1$



b) $\nu = 0.1$



c) $\nu = 0.01$

Figure 2. The Graphs of Obtained Solutions of Example 3 VIA the Proposed Method for Several N .

Table 13. Comparison of Exact Solution, Results of [11, 43, 25] and the Proposed Method to Solve Example 4 for $N=0.1$.

x	t	JFN-BFC	FE [43]	FD [25]	HWQ [11]	exact
0.25	0.4	0.31752	0.32091	0.32679	0.30887	0.31752
	0.6	0.24614	0.24910	0.25117	0.24609	0.24614
	0.8	0.19956	0.20211	0.20270	0.19952	0.19956
	1	0.16560	0.16782	0.16780	0.16557	0.16560
	3	0.02773	0.02828	0.02804	0.02775	0.02775
0.5	0.4	0.58453	0.58788	0.59661	0.56979	0.58454
	0.6	0.45798	0.46174	0.46581	0.45790	0.45798
	0.8	0.36740	0.37111	0.37293	0.36734	0.36740
	1	0.29834	0.30183	0.30253	0.29829	0.29834
	3	0.04103	0.04185	0.04155	0.04105	0.04106
0.75	0.4	0.64562	0.65054	0.64680	0.62567	0.64562
	0.6	0.50267	0.50825	0.50852	0.48715	0.50268
	0.8	0.38533	0.39068	0.39117	0.38525	0.38534
	1	0.29586	0.30057	0.30066	0.29578	0.29586
	3	0.03041	0.03106	0.03081	0.03043	0.03044

Table 12. The General Information of Solving Example 4 VIA the Proposed Method.

ν	M	N	RMS	IT
1	20	12	$3.11e-7$	2
0.1	26	11	$4.31e-5$	3
0.01	22	18	$7.55e-5$	4

According to the conditions of problem we use following expansion:

$$u_{M,N} = 4x(1 - x) + \sum_{i=1}^{N+1} \sum_{j=1}^{M+1} a_{ij} J_i(x) J_j(t),$$

and the condition $u(1, t)$ is satisfied in nonlinear system of equations. In Table 12 the general information of solving example 4 by the proposed method for $\nu = 1, 0.1, 0.01$ is presented. Solutions of this example for $\nu = 0.1, 0.01$ are presented in Table 13 and Table 14 respectively, these tables compare obtained results of the proposed method with Haar wavelet quasilinearization [11], least-squares quadratic B-spline finite element method [43], explicit Fourth-order finite difference method [25] and a parameter-uniform implicit finite difference methods [22].

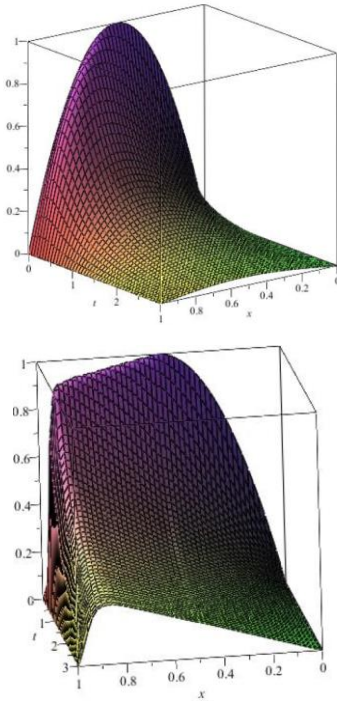
The convergence rate of proposed method to solve this equation for $\nu = 0.01, 0.1$ is shown in Table 15. The obtained graphs of this example by the proposed method have been shown in Figure 3 for $\nu = 0.1, 0.01$ respectively.

Table 14. Comparison of Between Exact Solution, Results of [11, 43, 22] and the Proposed Method to Solve Example 4 for $N=0.01$

x	t	JFN-BFC	FE [43]	FD [22]	HWQ [11]	exact
0.25	0.4	0.36225	0.36911	0.36273	0.36217	0.36226
	0.6	0.28204	0.28905	0.28212	0.28197	0.28204
	0.8	0.23046	0.23703	---	0.23040	0.23045
	1	0.19460	0.20069	0.19467	0.19465	0.19469
	3	0.07611	0.07865	0.07613	0.07613	0.07613
0.5	0.4	0.68368	0.68818	0.69186	0.68357	0.68368
	0.6	0.54833	0.55425	0.55125	0.54822	0.54832
	0.8	0.45376	0.46011	---	0.45363	0.45371
	1	0.38562	0.39206	0.38627	0.38561	0.38568
	3	0.15216	0.15576	0.15218	0.15217	0.15218
0.75	0.4	0.92050	0.92194	0.94940	0.92050	0.92050
	0.6	0.78298	0.78676	0.79399	0.78293	0.78299
	0.8	0.66274	0.66777	---	0.66264	0.66272
	1	0.56930	0.57491	0.57170	0.56924	0.56932
	3	0.22778	0.23183	0.22778	0.22774	0.22774

Table 15. Convergence Rate of the Proposed Method to Solve Example 4 for $N=1, 0.1$ and 0.01 .

$v = 1$			$v = 0.1$			$v = 0.01$		
M	N	RMS	M	N	RMS	M	N	RMS
12	8	6.95e-4	17	7	8.63e-2	18	11	3.19e-2
15	9	4.31e-5	20	8	1.81e-3	21	14	7.29e-3
18	10	2.01e-6	23	10	4.53e-4	25	16	3.74e-4
20	12	3.11e-7	26	12	3.36e-5	28	18	8.67e-5

**Figure 3.** The Graphs of Obtained Solutions of Example 4 VIA the Proposed Method for Several N .

5.5. Example 5:

Consider the 2-D system of nonlinear PDEs (1) with initial conditions:

$$u(x, y, 0) = x + y, \quad v(x, y, 0) = x - y, \quad (34)$$

and exact solutions:

$$u(x, y, t) = \frac{x+y-2xt}{1-2t^2}$$

$$v(x, y, t) = \frac{x-y-2yt}{1-2t^2}.$$

This example has been solved by discrete Adomian decomposition method with finite difference [35] and using modified Bi-cubic B-spline finite elements [36]. Proposed method has solved this example by $M = N = L = 6$ and two iterations of JFNGMRes. The obtained solutions of this example by the proposed method are compared with exact solution and results of [35, 36] in Table 16 and Table 17 for $t = 0.1$ and $t = 0.4$, respectively. In this example $D = [0.5, 0.5] \times [0.5, 0.5]$ and we set $v = 0.1$.

6. Conclusions

In this paper, a new algorithm was proposed to solve the 1D and the 2D viscous Burger equations. This method uses the spectral collocation method, the Bessel functions of the first kind as basis function and roots of the shifted Chebyshev polynomials as collocation point to convert nonlinear the nonlinear Burger equation to a nonlinear system of algebraic equations. Then, this nonlinear system is solved by using a Jacobian free Newton method and GMRes method with an adaptive preconditioner updated in each iteration. The obtained nonlinear system from spectral methods usually is large and ill-condition, therefore, iterative methods such as GMRes are preferred over direct solvers. Now we use an adaptive preconditioner to enhance the convergent rate of JFNGMRes.

As indicated in the presented examples, the solutions of the nonlinear systems are obtained in 3 to 5 Newton iterations, also in all examples the initial guess of JFNGMRes is simple vector $[0, 0, \dots, 0]^T$, that show the speed and power of the proposed method. Also, according to tables and comparison of proposed method and others methods show efficiently, applicability and reliability of collocation method based on the Bessel functions of the first kind. Some advantages of the presented method include high convergence rate of collocation method based on Bessel functions of the first kind to solve 1D and the 2D viscous Burger equations, few iterations for Newton method, simple initial guess for JFNGMRes method and no need for discretization and linearization and saving memory and processing.

Table 16. Comparison of Exact Solution, Results of [35, 36] and the Proposed Method to Solve Example 5 for $T=0.1$.

0.05	BFC-JFNGMRes		Exact solution		ADM& FD [35]		FE [36]	
(x, y)	$u(x, y, t)$	$v(x, y, t)$	$u(x, y, t)$	$v(x, y, t)$	$u(x, y, t)$	$v(x, y, t)$	$u(x, y, t)$	$v(x, y, t)$
(0.1, 0.1)	0.183674	-0.0204081	0.183673	-0.0204081	0.18368	-0.02041	0.183675	-0.0204010
(0.3, 0.1)	0.346935	0.1836735	0.346938	0.1836735	0.34694	0.18368	0.346931	0.1836730
(0.2, 0.2)	0.367349	-0.0408130	0.367346	-0.0408163	0.36735	-0.04082	0.367340	-0.0408102
(0.4, 0.2)	0.530608	0.1632680	0.530612	0.1632653	0.53062	0.16327	0.530613	0.1632653
(0.1, 0.3)	0.387755	-0.2653031	0.387755	-0.2653061	0.38776	-0.26531	0.387750	-0.2653003
(0.3, 0.3)	0.551034	-0.0612217	0.551020	-0.0612244	0.55103	-0.06123	0.551022	-0.0612212
(0.2, 0.4)	0.571422	-0.2856703	0.571428	-0.2857142	0.57144	-0.28572	0.571428	-0.2857124
(0.3, 0.4)	0.653070	-0.1836424	0.653061	-0.1836734	0.65307	-0.18368	0.653061	-0.1836711
(0.5, 0.5)	0.918326	-0.1020412	0.918367	-0.1020408	0.91838	-0.10205	0.918361	-0.1020405

Table 17. Comparison of Exact Solution, Results of [35, 36] and the Proposed Method to Solve Example 5 for T=0.4.

	BFC-JFNGMRes		Exact solution		ADM & FD [35]		FE [36]	
(x, y)	$u(x, y, t)$	$v(x, y, t)$	$u(x, y, t)$	$v(x, y, t)$	$u(x, y, t)$	$v(x, y, t)$	$u(x, y, t)$	$v(x, y, t)$
(0.1,0.1)	0.176476	-0.117645	0.1764705	-0.11764705	0.17657	-0.11729	0.176470	-0.1176411
(0.3,0.1)	0.235293	0.176461	0.2352941	0.17647058	0.23585	0.17657	0.235291	0.1764703
(0.2,0.2)	0.352900	-0.235220	0.3529411	-0.23529411	0.35314	-0.23458	0.352940	-0.2352935
(0.4,0.2)	0.411757	0.058819	0.4117647	0.05882352	0.41242	0.05928	0.411761	0.0588233
(0.1,0.3)	0.470573	-0.647042	0.4705882	-0.64705882	0.47044	-0.64574	0.470581	-0.6470530
(0.3,0.3)	0.529422	-0.352924	0.5294117	-0.35294117	0.52972	-0.35188	0.529412	-0.3529405
(0.2,0.4)	0.647025	-0.764714	0.6470588	-0.76470588	0.64701	-0.76303	0.647051	-0.7647033
(0.3,0.4)	0.676486	-0.617654	0.6764705	-0.61764705	0.67665	-0.61610	0.676472	-0.6176410
(0.5,0.5)	0.882307	-0.588268	0.8823529	-0.58823529	0.88286	-0.58646	0.882352	-0.5882311

References

- [1] Sohail, A., Maqbool, K. and Hayat, T. Painlevé property and approximate solutions using Adomian decomposition for a nonlinear KDV-like wave equation, *Appl. Math. Comput.*, 229, 359–366 (2014)
- [2] Zhang, B. and Lu, J. Exact solutions of homogeneous partial differential equation by a new Adomian decomposition method, *Procedia Environ. Sci.*, 11, 440–446 (2011)
- [3] Abbasbandy, S. and Darvishi, M. A numerical solution of Burgers equation by time discretization of Adomian decomposition method, *Appl. Math. Comput.*, 170, 95–102 (2005)
- [4] Dehghan, M., and Salehi, R. The use of Homotopy analysis method to solve the time-dependent nonlinear eikonal partial differential equation, *Zeit. Natur. A* 66, 259–271 (2011)
- [5] Fatoorehchi, H. and Abolghasemi, H. The variational iteration method for theoretical investigation of falling film absorbers, *Acad. Sci. Lett.* 38, 67–70 (2015)
- [6] Batiha, B., Noorani, M. and Hashim, I. Application of variational iteration method to the generalized Burger-Huxley equation, *Chaos, Solitons Fractals*, 36, 660–663 (2008)
- [7] Moghimi, M. and Hejazi, F.S. Variational iteration method for solving generalized Burger - Fisher and Burger equations, *Chaos, Solitons Fractals*, 33, 1756–1761 (2007)
- [8] Dehghan, M., Abbaszadeh, M. and Mohebbi, A. The numerical solution of nonlinear high dimensional generalized
- [9] Benjamin-Bona-Mahony- Burgers equation via the meshless method of radial basis functions, *Comput. Math. Appl.*, 68, 212–237 (2014)
- [10] Rad, J.A., Parand, K. and Abbasbandy, S. Local weak form meshless techniques based on the radial point interpolation (RPI) method and local boundary integral equation (LBIE) method to evaluate European and American options, *Commun. Nonlinear Sci. Numer. Simul.*, 22, 1178–1200 (2015)
- [11] Nikarya, M. An Analysis on Boundary Layer Flows over a Vertical Slender Cylinder Via Spectral Method, *Int. J. Indus. Engin. Manag. Sci.*, 1, 35–43 (2022)
- [12] Jiwari, R. A Haar wavelet quasilinearization approach for numerical simulation of Burgers equation, *Comput. Phys. Commun.*, 183, 2413–2423 (2012)
- [13] Khater, A., Tamsah, R. and Hassan, M. A Chebyshev spectral collocation method for solving Burger type equations, *J. Comput. Appl. Math.*, 222, 333–350 (2008)
- [14] Aksan, E. and Ozdes, A. A numerical solution of Burgers equation, *Appl. Math. Comput.*, 156, 395–402 (2004)
- [15] Khader, M.M. Approximate solutions for the problem of liquid film flow over an unsteady stretching sheet with thermal radiation and magnetic field, *Appl. Math. Mech.* (2018)
- [16] Wang, J., Liu, X. and Zhou, Y. A high-order accurate wavelet method for solving Schrödinger equations with general nonlinearity, *Appl. Math. Mech.*, 39, 1-16 (2018)
- [17] Pfeiffer, H.P., Kidder, L.E., Scheel, M.A. and Teukolsky, S.A. A multi-domain spectral method for solving elliptic equations, *Comput. Phys. Commun.*, 152, 253-273 (2003)
- [18] Li, B. and Chen, S. Direct spectral domain decomposition method for 2d incompressible Navier-Stokes equations, *Appl. Math. Mech.*, 36, 1073-1090 (2015)
- [19] Asaithambi, A. Numerical solution of the Burgers equation by automatic differentiation, *Appl. Math. Comput.*, 216, 2700-2708 (2010)
- [20] Mittal, R. and Rajni, R. A study of one-dimensional nonlinear diffusion equations by Bernstein polynomial based differential quadrature method, *J. Math. Chem.*, 1, 1-23 (2016)
- [21] Kutluay, S., Bahadır, A. and Özdes, A. Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods, *J. Comput. Appl. Math.*, 103, 251-261 (1999)
- [22] Hassanien, I., Salama, A. and Hosham, H. Fourth-order finite difference method for solving Burgers equation, *Appl. Math. Comput.*, 170, 781-800 (2005)
- [23] Kadalbajoo, M.K., Sharma, K. and Awasthi, A. A parameter-uniform implicit difference scheme for solving time-dependent Burgers equations, *Appl. Math. Comput.*, 170, 1365-1393 (2005)
- [24] Rauf, A., Abbas, Z., Shehzad, S. A., Alsaedi, A. and Hayat T., Numerical simulation of chemically reactive Powell-Eyring liquid flow with double diffusive Cattaneo-Christov heat and mass flux theories, *Appl. Math. Mech.*, 39, 467-476 (2018)
- [25] Aksan, E. Quadratic b-spline finite element method for numerical solution of the Burgers equation, *Appl. Math. Comput.*, 174 884-896 ((2006)
- [26] Ozis, T., Aksan, E. and Ozdes, A. A finite element approach for solution of Burgers equation, *Appl. Math. Comput.*, 139, 417-428 (2003)
- [27] Su, H., Yan, H. and Jin, B. Finite element method for coupled diffusion-deformation theory in polymeric gel based on slip-link model, *Appl. Math. Mech.*, 39, 581-596(2018)
- [28] Alcouffe, R. E., Brandt, A., Dendy, J. E. and Painter, J. W. The multi-grid method for the diffusion equation with strongly discontinuous coefficients, *SIAM J. Sci. Stat. Comput.*, 2, 430-454 (1981)
- [29] Fish, J. and Belsky, V. Multi-grid method for periodic heterogeneous media part 2: Multiscale modeling and quality

- control in multidimensional case, *Comput. Methods Appl. Mech. Eng.*, 126, 17-38. (1995)
- [30] Parand, K. and Nikarya, M. Application of Bessel functions for solving differential and integro-differential equations of the fractional order, *Appl. Math. Model.*, 38, 4137-4147 (2014)
- [31] Rasanan, A.H.H., Nikarya, M. Moayeri, MM. and Bahramnezhad, A. A comparison between pre-Newton and post-Newton approaches for solving a physical singular second-order boundary problem in the semi-infinite interval, *Comput. Math. Comput. Modeling with Appl.*, 91, 1239-1254 (2023)
- [32] Tajvidi, T., Razzaghi, M. and Dehghan, M. Modified rational Legendre approach to laminar viscous flow over a semi-infinite flat plate, *Chao. Solit. Frac.*, 35, 59-66 (2008)
- [33] Soulaïmani, A., Salah, N. B. and Saad, Y. Enhanced GMRES acceleration techniques for some CFD problems, *I. J. Comput. Fluid Dynam.*, 16, 1-20 (2002)
- [34] Chen, Y. and Shen, C. A Jacobian-free Newton-GMRES(m) method with adaptive preconditioner and its application for power flow calculations, *IEEE Trans. Power Syst.*, 21, 1096-1103 (2006)
- [35] C. Fletcher, A comparison of finite element and finite difference solutions of the one- and two-dimensional Burgers equations, *J. Comput. Phys.*, 51, 159-188 (1983)
- [36] Zhu, H., Shu, H. and Ding, M. Numerical solutions of two-dimensional Burgers equations by discrete Adomian decomposition method, *Comput. Math. Appl.*, 60, 840-848 (2010)
- [37] Mittal, R.C. and Tripathi, A. Numerical solutions of two-dimensional Burgers equations using modified bi-cubic b-spline finite elements, *Eng. Comput.*, 32, 1275-1306 (2015)
- [38] Zheng, Q., Zhao, X. and Liu, Y. A novel finite difference scheme for Burgers equation on unbounded domains, *Appl. Numer. Math.*, 111, 1-16 (2017)
- [39] Fernando, A. M. and Hu, F. Q. DGM-FD: A finite difference scheme based on the discontinuous Galerkin method applied to wave propagation, *J. Comput. Phys.*, 230, 4871-4898 (2011)
- [40] Liao, W. A fourth-order finite-difference method for solving the system of two-dimensional Burgers equations, *Int. J. Numer. Methods Fluids*, 64, 565-590 (2010)
- [41] Liao, W. and Zhu, J. Efficient and accurate finite difference schemes for solving one-dimensional Burgers equation, *Int. J. Comput. Math.*, 88, 2575-2590 (2011)
- [42] Uzunca, M., Karasozen, B. and Kucukseyhan, T. Moving mesh discontinuous Galerkin methods for PDEs with traveling waves, *Appl. Math. Comput.*, 292, 9-18 (2017)
- [43] Caldwell, J. Wanless, P. and Cook, A. A finite element approach to Burgers equation, *Appl. Math. Modell.*, 5, 189-193 (1981)
- [44] Kutluay, S., Esen, A. and Dag, I. Numerical solutions of the Burgers equation by the least-squares quadratic B-spline finite element method, *J. Comput. Appl. Math.*, 167, 21-33 (2004)
- [45] Ghayebi, B., Hosseini, S. M. and Blomker, D. Numerical solution of the Burgers equation with Neumann boundary noise, *J. Comput. Appl. Math.*, 311, 148-164 (2017)
- [46] Basto, M., Semiao, V. and Calheiros, F. L. Numerical study of modified Adomian's method applied to Burgers equation, *J. Comput. Appl. Math.*, 206, 927-949 (2017)
- [47] Al-Fhaid, A. S. A quasilinear and Chebyshev collocation method for solving nonlinear 1-D Burgers-type equations, *J. Comput. Theor. Nanosci.*, 13, 3112-3121 (2016)
- [48] Saka, B. Algorithms for numerical solution of the modified equal width wave equation using collocation method, *Math. and Comput. Model.*, 45, 1096-1117 (2007)
- [49] Soliman, A. The modified extended tanh-function method for solving Burgers-type equations, *Physica A*, 361, 394-404 (2006)
- [50] Adams, R. A. Sobolev Spaces, Academic Press, New York (1975)
- [51] Watson, G. A treatise on the theory of Bessel Functions, 2nd edition, Cambridge University Press, Cambridge (England) (1967)
- [52] Nikarya, M. and Sarabadan, S., An Improvement on Collocation Algorithm to Solve Initial Value Problems, *Int. J. Indus. Math.* 14, 279-289 (2022)
- [53] Guo, B. Y. Spectral Methods and Their Applications, World Scientific, Singapore, River Edge (1998)
- [54] Tadmor, E. Convergence of spectral methods for nonlinear conservation laws, *SIAM J. Numer. Anal.*, 26, 30-44 (1989)
- [55] Wu, S. and Liu, X. Convergence of spectral method in time for Burgers equation, *Acta Mathematicae Applicatae Sinica*, 13, 314-320 (1997)
- [56] Weinan, E. Convergence of spectral methods for Burgers equation, *SIAM J. Numer. Anal.*, 29, 1520-1541 (1997)
- [57] Atkinson, K., Hansen, O. and Chien, D. A spectral method for parabolic differential equations, *Numer. Algorithms*, 63, 213-237 (2013)
- [58] Guo, B. Jacobi approximations in certain Hilbert spaces and their applications to singular differential equations, *J. Math. Anal. Appl.*, 243, 373-408 (2000)
- [59] Shen, J., Tang, L. and Wang, L. Spectral Methods: Algorithms, Analysis and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
- [60] Parand, K. and Nikarya, M. A numerical method to solve the 1D and the 2D reaction diffusion equation based on Bessel functions and Jacobian free Newton-Krylov subspace methods, *Eur. Phys. J. Plus*, 132, 496-514 (2017)
- [61] Shroff, G. M. and Keller, H. B. Stabilization of unstable procedures: The recursive projection method, *SIAM J. Numer. Anal.*, 30, 1099-1120 (1993)
- [62] Cordero, A., Hueso, J. L., Martinez, E. and Torregrosa, J. R. Accelerated methods of order for systems of nonlinear equations, *J. Comput. Appl. Math.*, 233, 2696-2702 (2010)
- [63] Nedzhibov, G. H. A family of multi-point iterative methods for solving systems of nonlinear equations, *J. Comput. Appl. Math.*, 222, 244-250 (2008)
- [64] Knoll, D. and Keyes, D. Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *J. Comput. Phys.*, 193, 357-397 (2004)
- [65] Asgharzadeh, H. and Borazjani, I. A Newton-Krylov method with an approximate analytical Jacobian for implicit solution of Navier-Stokes equations on staggered overset-curvilinear grids with immersed boundaries, *J. Comput. Phys.*, 331, 227-256 (2017)
- [66] Hajizadeh, A., Kazeminejad, H. and Talebi, S. A new numerical method for solution of boiling flow using combination of SIMPLE and Jacobian free Newton-Krylov algorithms, *Prog. Nucl. Energy*, 95, 48-60 (2017)
- [67] Wood, W. L. An exact solution for Burgers equation, *Commun. Numer. Meth. Eng.*, 22, 797-798 (2006).