

Article

# Numerical Solutions of Coupled Burgers' Equations

Hijaz Ahmad <sup>1</sup> , Tufail A. Khan <sup>1</sup> and Clemente Cesarano <sup>2,\*</sup>

<sup>1</sup> Department of Basic Sciences, University of Engineering and Technology, Peshawar 25000, Pakistan; hijaz555@gmail.com (H.A.); tufailmarwat@uetpeshawar.edu.pk (T.A.K.)

<sup>2</sup> Section of Mathematics, International Telematic University Uninettuno, 39, 00186 Roma, Italy

\* Correspondence: c.cesarano@uninettunouniversity.net

Received: 9 October 2019; Accepted: 21 October 2019; Published: 23 October 2019



**Abstract:** In this article, two new modified variational iteration algorithms are investigated for the numerical solution of coupled Burgers' equations. These modifications are made with the help of auxiliary parameters to speed up the convergence rate of the series solutions. Three numerical test problems are given to judge the behavior of the modified algorithms, and error norms are used to evaluate the accuracy of the method. Numerical simulations are carried out for different values of parameters. The results are also compared with the existing methods in the literature.

**Keywords:** modified variational iteration algorithm-I; modified variational iteration algorithm-II; coupled Burgers equation; MVIA-II

## 1. Introduction

In recent years, coupled partial differential equations have been employed in various fields of engineering and applied sciences. Coupled Burgers' equations are coupled partial differential equations (PDEs), and describe the approximation theory of flow through a shock wave traveling in a viscous fluid.

In this paper, the following type of coupled Burger equations will be investigated:

$$\begin{cases} \frac{\partial u}{\partial t} - \alpha_1 \frac{\partial^2 u}{\partial x^2} + \beta_{1,1} \frac{\partial u}{\partial x} + \beta_{1,2} \frac{\partial uv}{\partial x} = 0, & t \in (0, T], \quad x \in [a, b], \\ \frac{\partial v}{\partial t} - \alpha_2 \frac{\partial^2 v}{\partial x^2} + \beta_{2,2} \frac{\partial v}{\partial x} + \beta_{2,1} \frac{\partial uv}{\partial x} = 0, & t \in (0, T], \quad x \in [a, b], \end{cases} \quad (1)$$

where  $\alpha_1$  and  $\alpha_2$  are positive and non-zero viscosity parameters,  $\beta_{1,1}$ ,  $\beta_{1,2}$ ,  $\beta_{2,2}$  and  $\beta_{2,1}$  are constants depend on the Stokes velocity. The initial and boundary conditions for coupled Burgers' Equation (1) are the following:

$$\begin{cases} u(x, 0) = \varnothing_1(x), & x \in [a, b], \\ v(x, 0) = \varnothing_2(x), & x \in [a, b], \end{cases} \quad (2)$$

$$\begin{cases} u(a, t) = \varnothing_3(t), & u(b, t) = \varnothing_2(t) \quad x \in [a, b], \\ v(a, t) = \varnothing_3(t), & v(b, t) = \varnothing_4(t) \quad x \in [a, b], \end{cases} \quad (3)$$

Many authors have solved different types of coupled Burgers' equations by different techniques such as the Fourier pseudospectral method [1] applied by Rashid and Ismail for one-dimensional coupled Burgers equations. A mesh-free interpolation method was employed by Islam et al. [2], where the radial basis function (RBF) collocation method has been formulated for the numerical treatment of coupled Burger equations and other nonlinear PDEs. Khater et al. [3] used the Chebyshev spectral collocation method, which has been developed using Chebyshev polynomials, and the Runge-Kutta 4th order method (RK4) method. Kya [4] used the decomposition method for the solution of a viscous coupled Burger equation, and obtained solutions in the form of a convergent power series. Mittal and

Arora [5] proposed a scheme known as the Lai cubic B-spline collocation scheme for the solution of coupled viscous Burger equations, where the authors used a Crank–Nicolson scheme and cubic B-spline functions for time integration and space integration, respectively. Lai and Ma [6] proposed a new lattice Boltzmann model for the solution of coupled Burger equations, after selecting the distribution functions, Chapman–Enskog expansion was employed for the solution of coupled Burger equations. The Chebyshev–Legendre pseudospectral method [7] has been utilized by Rashid et al. for coupled viscous Burgers’ (VB) equations, where the leapfrog scheme and Chebyshev–Legendre Pseudo-Spectral method (CLPS) method were used for the time direction and space direction, respectively. Kumar and Pandit [8] used a composite numerical scheme for the numerical evaluation of coupled Burger equations, where the scheme was developed based on Haar wavelets and finite difference. Mohammadi and Mokhtari [9] used a reproducing Kernel method for an analytical solution in the form series of systems of Burger equations. At last, in 2019, Bak et al. [10] developed a new approach named a semi-Lagrangian approach for the numerical solution of coupled Burger equations. We compare our results with those of [10], and show the applicability of our proposed method to handle such problems as those that arise in applied science and engineering.

This paper aims to solve three types of coupled Burgers’ equations by employing variational iteration algorithm-II and one of the examples to be solved by modified variational iteration algorithm-I. The organization of the rest of the paper is as follows; in Section 2, we elucidate the variational iteration algorithm-II, in the next Section 3, the semi-numerical method is applied to three test problems, and a comparison is made with some other methods; lastly, some conclusions are drawn in the last Section 4.

## 2. Modified Variational Iteration Algorithm-II

Consider the nonlinear diffusion equation:

$$L[w(x, t)] + N[w(x, t)] = c(x, t), \tag{4}$$

where the terms  $L[w(x, t)]$  and  $N[w(x, t)]$  are linear and nonlinear terms in that order, while  $c(x, t)$  is the source term. For a given  $w_0(x, t)$ , the approximate solution  $w_{n+1}(x, t)$  of Equation (4) can be obtained as below:

$$w_{n+1}(x, t) = w_n(x, t) + \int_0^t \lambda(s) [L\{w_n(x, s)\} + N\{\widetilde{w}_n(x, s)\} - c(x, s)] ds, \tag{5}$$

In Equation (5),  $\lambda$  is a parameter, which is known as the Lagrange multiplier [11]. This Lagrange multiplier is obtained by variational theory, where  $\widetilde{w}_n$  is a term being restricted, which in turn gives  $\delta \widetilde{w}_n = 0$  and gives the following Lagrange multipliers:

$$\lambda = -1 \text{ for } m = 1, \quad \lambda = s - t \text{ for } m = 2,$$

and in general, the Lagrange multiplier for  $m \geq 1$ :

$$\lambda = \frac{(-1)^m (s - t)^{m-1}}{(m - 1)!}. \tag{6}$$

Putting the value of the Lagrange multiplier from Equation (6) in the correctional function shown in Equation (5), we get the following iterative formula:

$$w_{n+1}(x, t) = w_n(x, t) + \int_0^t \frac{(-1)^m (s - t)^{m-1}}{(m - 1)!} [L\{w_n(x, s)\} + N\{w_n(x, s)\} - c(x, s)] ds, \tag{7}$$

This is known as variational iteration algorithm-I [12–14], which is an advance improvement of the common Lagrange multiplier method [11]. These days, this technique [15–23] has been set up for

offering a solution for a more extensive scope of problems, developing in several fields of pure and applied sciences. PDEs extensively arise in various physical applications such as propagation and the scattering of waves, magnetohydrodynamic flow through pipes, computational fluid dynamics, magnetic resonance imaging, the phenomena of turbulence and supersonic flow, the flow of a shock wave traveling in a viscous fluid, acoustic transmission, traffic and aerofoil flow theory, and the proposed technique has the ability to investigate these types problems effectively.

Inserting an auxiliary parameter in the variational iteration method and presenting a new modification of it, as below:

$$w_{n+1}(x, t) = w_n(x, t) + h \int_0^t \frac{(-1)^m (s-t)^{m-1}}{(m-1)!} [L\{w_n(x, s)\} + N\{w_n(x, s)\} - c(x, s)] ds, \tag{8}$$

we call it modified variational iteration algorithm-I. We will apply this method for a test problem for verifying its accuracy and compactness. According to He et al. [21], we can construct a more concise iteration formulation, which can be written as:

$$w_{n+1}(x, t) = w_0(x, t) + \int_0^t \frac{(-1)^m (s-t)^{m-1}}{(m-1)!} [N\{w_n(x, s)\} - c(x, s)] ds, \tag{9}$$

Equation (9) is called variational iteration algorithm-II; we further modify it by coupling an auxiliary term, and named it modified variational iteration algorithm-II, which can be written as:

$$w_{n+1}(x, t) = w_0(x, t) + h \int_0^t \frac{(-1)^m (s-t)^{m-1}}{(m-1)!} [N\{w_n(x, s)\} - c(x, s)] ds, \tag{10}$$

The iterative algorithm does not contain unsure constants aside from an auxiliary parameter  $h$ , which is utilized to control the convergence of the obtained solution ideally by limiting the norm 2 of the residual error over the space of the given problem. The ideal decision of this  $h$  improves the precision and proficiency of the algorithm. In the wake of presenting  $h$ , Equation (10) takes the structure:

$$\begin{aligned} w_0(\zeta) &\text{ is a proper initial approximation,} \\ w_1(\zeta, h) &= w_0(\zeta) + h \int_0^\zeta \lambda(s, \zeta) [N\{w_0(s)\} - c(s)] ds, \\ w_{n+1}(\zeta, h) &= w_0(\zeta, h) + h \int_0^\zeta \lambda(s, \zeta) [N\{w_n(s, h)\} - c(s, h)] ds, \\ &n = 1, 2, 3, \dots \end{aligned} \tag{11}$$

Now, we may start the procedures with the selective function  $w_0(x, t)$  and use the above iteration structures to get the approximate solutions. The iteration structure shown in Equation (11) will give several approximations of  $w(x, t)$ , and the exact solutions are obtained at the limit of the resulting successive approximations, i.e.:

$$w(x, t) = \lim_{n \rightarrow \infty} w_n(x, t). \tag{12}$$

This algorithm is named MVIA-II. We utilize this modified algorithm for the solution of coupled nonlinear partial differential equations.

### 3. Numerical Assessments

In this section, we discuss three test problems of different types of coupled Burgers' equations to check the accuracy of the proposed method. We assess the accuracy of the method by taking different values of parameters, and the obtained results are very encouraging and significant, while the error norms are calculated and compared with the error of the other methods available in the literature. For numerical computation, Maple 16 and MATLAB R2015a are used on a Dell core i3, 1.90-GHz PC with 4 GB of RAM.

### 3.1. Test Problem 1

Consider the coupled Burger’s Equation (1) with  $\alpha_1 = \alpha_2 = 1, \beta_{1,1} = \beta_{2,2} = -2$  and  $x \in [-10, 10]$ . The equation becomes:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - 2\frac{\partial u}{\partial x} + \beta_{1,2}\frac{\partial uv}{\partial x} = 0, & t > 0, \quad x \in [-10, 10], \\ \frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial x^2} - 2\frac{\partial v}{\partial x} + \beta_{2,1}\frac{\partial uv}{\partial x} = 0, & t > 0, \quad x \in [-10, 10], \end{cases} \tag{13}$$

The exact solution was given by [10]:

$$\begin{cases} u(x, t) = k(1 - \tanh(B(x - 2Bt))), \\ v(x, t) = k\left(\frac{2\beta_{2,1}-1}{2\beta_{1,2}-1} - \tanh(B(x - 2Bt))\right), \end{cases} \tag{14}$$

where  $\beta_{1,2}$  and  $\beta_{2,1}$  are nonzero arbitrary constants,  $k = 0.05$  and  $B = k\left(\frac{4\beta_{2,1}-1}{2(2\beta_{1,2}-1)}\right)$ . In this example, we investigate the proficiency and accuracy of the proposed algorithm by changing the values of arbitrary constants  $\beta_{1,2}$  and  $\beta_{2,1}$ , whose different magnitudes from each other produce the changed performance of  $u$  and  $v$ . The approximate numerical solutions are calculated with different parameters  $(\beta_{1,2}, \beta_{2,1}) = (0.1, 0.3)$  by varying the  $t$  from  $t = 0.5$  to  $10.0$ .

Constructing the correction function for the Equation (13) as:

$$\begin{aligned} u_{n+1}(x, t, h_1) = & u_n(x, t, h_1) \\ & + h_1 \int_0^t \lambda_1(\zeta) \left\{ \frac{\partial u_n(x, \zeta, h_1)}{\partial \zeta} - \frac{\partial^2 u_n(x, \zeta, h_1)}{\partial x^2} - 2\frac{\partial u_n(x, \zeta, h_1)}{\partial x} \right. \\ & \left. + \beta_{1,2} \frac{\partial u_n(x, \zeta, h_1)v_n(x, \zeta, h_1)}{\partial x} \right\} d\zeta. \end{aligned} \tag{15}$$

$$\begin{aligned} v_{n+1}(x, t, h) = & v_n(x, t, h) \\ & + h \int_0^t \lambda_2(\zeta) \left\{ \frac{\partial v_n(x, \zeta, h)}{\partial \zeta} - \frac{\partial^2 v_n(x, \zeta, h)}{\partial x^2} - 2\frac{\partial v_n(x, \zeta, h)}{\partial x} \right. \\ & \left. + \beta_{2,1} \frac{\partial u_n(x, \zeta, h)v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \end{aligned} \tag{16}$$

The values of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$  may be obtained most positively by the variational principle [24,25]. We obtain the estimation of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$ , which is  $\lambda_1(\zeta) = \lambda_2(\zeta) = -1$ .

First, we use the modified variational iteration algorithm-I; then, we will use the modified algorithm-II for the solution of this test problem. Utilizing the estimation of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$  in Equations (15) and (16) results in the underneath iterative structure:

$$u_{n+1}(x, t, h) = u_n(x, t, h_1) - h \int_0^t \left\{ \frac{\partial u_n(x, \zeta, h)}{\partial \zeta} - \frac{\partial^2 u_n(x, \zeta, h)}{\partial x^2} - 2\frac{\partial u_n(x, \zeta, h)}{\partial x} + \beta_{1,2} \frac{\partial u_n(x, \zeta, h)v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \tag{17}$$

$$v_{n+1}(x, t, h) = v_n(x, t, h) - h \int_0^t \left\{ \frac{\partial v_n(x, \zeta, h)}{\partial \zeta} - \frac{\partial^2 v_n(x, \zeta, h)}{\partial x^2} - 2\frac{\partial v_n(x, \zeta, h)}{\partial x} + \beta_{2,1} \frac{\partial u_n(x, \zeta, h)v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \tag{18}$$

Introducing with a proper initial guess:

$$\begin{aligned} u(x, 0) &= k(1 - \tanh(B(x))), \\ v(x, 0) &= k\left(\frac{2\beta_{2,1}-1}{2\beta_{1,2}-1} - \tanh(B(x))\right), \end{aligned}$$

one can get the beneath different approximations by utilizing the recurrence relations shown in Equations (17) and (18) and  $\beta_{1,2} = 0.1, \beta_{2,1} = 0.3$ :

$$\begin{aligned}
 u_1(x, t, h) &= (11 * h * t * \exp((11 * x)/450) * (299 * \exp((11 * x)/450) + 475)) / (36,450,000 \\
 &\quad * (\exp((11 * x)/450) + 1)^3 - \tanh((11 * x)/900) / 45 + 1/20, \\
 v_1(x, t, h) &= (11 * h * t * \exp((11 * x)/450) * (47 * \exp((11 * x)/450) + 127)) / (12,150,000 \\
 &\quad * (\exp((11 * x)/450) + 1)^3 - \tanh((11 * x)/900) / 45 + 1/40, \\
 u_2(x, t, h) &= (11 * h * t * \exp((11 * x)/450) * (299 * \exp((11 * x)/450) + 475)) / (36,450,000 \\
 &\quad * (\exp((11 * x)/450) + 1)^3 - h * ((11 * t * \exp((11 * x)/225) * (589,808,450 \\
 &\quad * h^2 * f2 + 611,112,954,375 * h * t + 1,082,072,925,000,000 * h \\
 &\quad - 1,082,072,925,000,000)) / (17,936,133,750,000,000,000 * (\exp((11 * x)/450) + 1)^7) \\
 &\quad - (11 * t * \exp((11 * x)/450) * (147,130,425,000,000 * \exp((11 * x)/90) \\
 &\quad - 233,735,625,000,000 * h + 1,817,725,050,000,000 * \exp((11 * x)/150) \\
 &\quad + 1,990,935,450,000,000 * \exp((11 * x)/225) + 822,257,325,000,000 * \exp((22 \\
 &\quad * x)/225) - 147,130,425,000,000 * h * \exp((11 * x)/90) - 1,817,725,050,000,000 * h \\
 &\quad * \exp((11 * x)/150) - 1,990,935,450,000,000 * h * \exp((11 * x)/225) \\
 &\quad - 822,257,325,000,000 * h * \exp((22 * x)/225) - 372,851,758,125 * h * t \\
 &\quad + 643,660,226 * h^2 * f2 * \exp((11 * x)/150) + 82,850,878 * h^2 * f2 * \exp((11 \\
 &\quad * x)/225) + 226,552,898 * h^2 * f2 * \exp((22 * x)/225) + 146,715,958,125 * h * t \\
 &\quad * \exp((11 * x)/90) + 1,368,940,196,250 * h * t * \exp((11 * x)/150) \\
 &\quad + 440,146,203,750 * h * t * \exp((11 * x)/225) + 837,248,754,375 * h * t * \exp((22 \\
 &\quad * x)/225) + 233,735,625,000,000)) / (17,936,133,750,000,000,000 * (\exp((11 * x)/450) \\
 &\quad + 1)^7)) - \tanh((11 * x)/900) / 45 + 1/20, \\
 v_2(x, t, h) &= (11 * h * t * \exp((11 * x)/450) * (47 * \exp((11 * x)/450) + 127)) / (12,150,000 \\
 &\quad * (\exp((11 * x)/450) + 1)^3 - h * ((11 * t * \exp((11 * x)/225) * (53,630,830 * h^2 \\
 &\quad * f2 + 22,989,470,625 * h * t + 91,033,875,000,000 * h \\
 &\quad - 91,033,875,000,000)) / (1,992,903,750,000,000,000 * (\exp((11 * x)/450) + 1)^7) \\
 &\quad - (11 * t * \exp((11 * x)/450) * (7,709,175,000,000 * \exp((11 * x)/90) \\
 &\quad - 20,831,175,000,000 * h + 129,579,750,000,000 * \exp((11 * x)/150) \\
 &\quad + 155,823,750,000,000 * \exp((11 * x)/225) + 51,667,875,000,000 * \exp((22 \\
 &\quad * x)/225) - 7,709,175,000,000 * h * \exp((11 * x)/90) - 129,579,750,000,000 * h \\
 &\quad * \exp((11 * x)/150) - 155,823,750,000,000 * h * \exp((11 * x)/225) \\
 &\quad - 51,667,875,000,000 * h * \exp((22 * x)/225) - 27,835,396,875 * h * t + 47,729,902 \\
 &\quad * h^2 * f2 * \exp((11 * x)/150) + 42,038,546 * h^2 * f2 * \exp((11 * x)/225) \\
 &\quad + 8,746,606 * h^2 * f2 * \exp((22 * x)/225) + 3,778,396,875 * h * t * \exp((11 \\
 &\quad * x)/90) + 115,439,073,750 * h * t * \exp((11 * x)/150) + 77,016,926,250 * h * t \\
 &\quad * \exp((11 * x)/225) + 47,046,470,625 * h * t * \exp((22 * x)/225) \\
 &\quad + 20,831,175,000,000)) / (1,992,903,750,000,000,000 * (\exp((11 * x)/450) + 1)^7)) \\
 &\quad - \tanh((11 * x)/900) / 45 + 1/40, \\
 &\quad \vdots \\
 &\quad \vdots
 \end{aligned}$$

we stop the procedure at the third-order approximation. In order to find an appropriate value of  $h$  for the approximate solutions, a residual function can be defined as:

$$\begin{aligned}
 r_{3a}(x, t, h) &= \frac{\partial u_3(x, t, h)}{\partial t} - \frac{\partial^2 u_3(x, t, h)}{\partial x^2} - 2 \frac{\partial u_3(x, t, h)}{\partial x} + 0.1 \frac{\partial u_3(x, t, h) v_3(x, t, h)}{\partial x}, \\
 r_{3b}(x, t, h) &= \frac{\partial u_3(x, t, h)}{\partial t} - \frac{\partial^2 u_3(x, t, h)}{\partial x^2} - 2 \frac{\partial u_3(x, t, h)}{\partial x} + 0.3 \frac{\partial u_3(x, t, h) v_3(x, t, h)}{\partial x},
 \end{aligned}
 \tag{19}$$

The square of the  $r_{3a}(x, t, h)$  and  $r_{3b}(x, t, h)$  functions with respect to the auxiliary term  $h$  in the domain  $(x, t) \in [0, 1] \times [0, 1]$  is:

$$\begin{aligned}
 e_{3a}(h) &= \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=-10}^{10} (r_3(i, j, h))^2 \right]^{\frac{1}{2}} \\
 e_{3b}(h) &= \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=-10}^{10} (r_3(i, j, h))^2 \right]^{\frac{1}{2}}
 \end{aligned}
 \tag{20}$$

For finding an optimum value of the auxiliary term  $h$ , the lowest point of the error of norm 2 of Function (19) is selected. The lowest point of  $e_{3a}(h)$ , as  $h = 0.994178085892076$  and  $e_{3b}(h)$  as  $h = 0.996482304232335$  is gained by means of Maple mathematical software. By replacing the value of the auxiliary term in  $u_3(x, t, h)$ , the absolute error of the fourth-order approximation of the proposed algorithm decreases remarkably.

Now, we employ the modified variational iteration algorithm-II for the solution of test problem 1. Utilizing this estimation of  $\lambda(\zeta)$  in Equation (13) results in the underneath iterative structure:

$$\begin{aligned}
 u_{n+1}(x, t, h_1) &= u_0(x, t, h_1) \\
 &\quad - h_1 \int_0^t \left\{ -\frac{\partial^2 u_n(x, \zeta, h_1)}{\partial x^2} - 2 \frac{\partial u_n(x, \zeta, h_1)}{\partial x} + \beta_{1,2} \frac{\partial u_n(x, \zeta, h_1) v_n(x, \zeta, h_1)}{\partial x} \right\} d\zeta.
 \end{aligned}
 \tag{21}$$

$$\begin{aligned}
 v_{n+1}(x, t, h) &= v_0(x, t, h) \\
 &\quad - h \int_0^t \left\{ -\frac{\partial^2 v_n(x, \zeta, h)}{\partial x^2} - 2 \frac{\partial v_n(x, \zeta, h)}{\partial x} + \beta_{2,1} \frac{\partial u_n(x, \zeta, h) v_n(x, \zeta, h)}{\partial x} \right\} d\zeta.
 \end{aligned}
 \tag{22}$$

Introducing with a proper initial guess:

$$u(x, 0) = k(1 - \tanh(B(x))),$$

$$v(x, 0) = k \left( \frac{2^{22}_{2,1} - 1}{2^{22}_{1,2} - 1} - \tanh(B(x)) \right),$$

one can get the beneath different approximations by utilizing the recurrence relations shown in Equations (21) and (22) for  $\beta_{1,2} = 0.1$  and  $\beta_{2,1} = 0.3$ :

$$\begin{aligned}
 u_1(x, t, h) &= (11 * h * t * \exp((11 * x)/450) * (299 * \exp((11 * x)/450) + 475)) / (36,450,000 \\
 &\quad * (\exp((11 * x)/450) + 1)^3 - \tanh((11 * x)/900) / 45 + 1/20,
 \end{aligned}$$

$$\begin{aligned}
 v_1(x, t, h) &= (11 * h * t * \exp((11 * x)/450) * (47 * \exp((11 * x)/450) + 127)) / (12,150,000 \\
 &\quad * (\exp((11 * x)/450) + 1)^3 - \tanh((11 * x)/900) / 45 + 1/40,
 \end{aligned}$$

$$\begin{aligned}
 u_2(x, t, h) &= 1/20 - h * ((11 * t * \exp((11 * x)/225) * (589,808,450 * h^2 * t^2 + 61,111,2954,375 * h \\
 &\quad * t - 1,082,072,925,000,000)) / (17,936,133,750,000,000,000 * (\exp((11 * x)/450) \\
 &\quad + 1)^7) - (11 * t * \exp((11 * x)/450) * (147,130,425,000,000 * \exp((11 * x)/90) \\
 &\quad + 1,817,725,050,000,000 * \exp((11 * x)/150) + 1,990,935,450,000,000 * \exp((11 \\
 &\quad * x)/225) + 822,257,325,000,000 * \exp((22 * x)/225) - 372,851,758,125 * h * t \\
 &\quad + 643,660,226 * h^2 * t^2 * \exp((11 * x)/150) + 82,850,878 * h^2 * t^2 * \exp((11 \\
 &\quad * x)/225) + 226,552,898 * h^2 * t^2 * \exp((22 * x)/225) + 146,715,958,125 * h * t \\
 &\quad * \exp((11 * x)/90) + 1,368,940,196,250 * h * t * \exp((11 * x)/150) \\
 &\quad + 440,146,203,750 * h * t * \exp((11 * x)/225) + 837,248,754,375 * h * t * \exp((22 \\
 &\quad * x)/225) + 233,735,625,000,000)) / (17,936,133,750,000,000,000 * (\exp((11 * x)/450) \\
 &\quad + 1)^7) - \tanh((11 * x)/900) / 45,
 \end{aligned}$$

$$\begin{aligned}
 v_2(x, t, h) = h * & \left( (11 * t * \exp((11 * x) / 450) * (7,709,175,000,000 * \exp((11 * x) / 90) \right. \\
 & + 129,579,750,000,000 * \exp((11 * x) / 150) + 155,823,750,000,000 * \exp((11 \\
 & * x) / 225) + 51,667,875,000,000 * \exp((22 * x) / 225) - 27,835,396,875 * h * t \\
 & + 47,729,902 * h^2 * t^2 * \exp((11 * x) / 150) + 42,038,546 * h^2 * t^2 * \exp((11 \\
 & * x) / 225) + 8,746,606 * h^2 * t^2 * \exp((22 * x) / 225) + 3,778,396,875 * h * t \\
 & * \exp((11 * x) / 90) + 115,439,073,750 * h * t * \exp((11 * x) / 150) + 77,016,926,250 \\
 & * h * t * \exp((11 * x) / 225) + 47,046,470,625 * h * t * \exp((22 * x) / 225) \\
 & + 20,831,175,000,000) / (1,992,903,750,000,000,000 * (\exp((11 * x) / 450) + 1)7) \\
 & - (11 * t * \exp((11 * x) / 225) * (53,630,830 * h^2 * t^2 + 22,989,470,625 * h * t \\
 & - 91,033,875,000,000) / (1,992,903,750,000,000,000 * (\exp((11 * x) / 450) + 1)7)) \\
 & - \tanh((11 * x) / 900) / 45 + 1 / 40, \\
 & \vdots \\
 & \vdots
 \end{aligned}$$

we stop the procedure at the third-order approximation. In order to find an appropriate value of  $h$  for the approximate solutions, a residual function can be defined as:

$$\begin{aligned}
 r_{3a}(x, t, h) &= \frac{\partial u_3(x, t, h)}{\partial t} - \frac{\partial^2 u_3(x, t, h)}{\partial x^2} - 2 \frac{\partial u_3(x, t, h)}{\partial x} + 0.1 \frac{\partial u_3(x, t, h) v_3(x, t, h)}{\partial x} \\
 r_{3b}(x, t, h) &= \frac{\partial u_3(x, t, h)}{\partial t} - \frac{\partial^2 u_3(x, t, h)}{\partial x^2} - 2 \frac{\partial u_3(x, t, h)}{\partial x} + 0.3 \frac{\partial u_3(x, t, h) v_3(x, t, h)}{\partial x}
 \end{aligned} \tag{23}$$

The square of the  $r_{3a}(x, t, h)$  and  $r_{3b}(x, t, h)$  functions with respect to the auxiliary term  $h$  in the domain  $(x, t) \in [0, 1] \times [0, 1]$  is:

$$\begin{aligned}
 e_{3a}(h) &= \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=-10}^{10} (r_3(i, j, h))^2 \right]^{\frac{1}{2}} \\
 e_{3b}(h) &= \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=-10}^{10} (r_3(i, j, h))^2 \right]^{\frac{1}{2}}
 \end{aligned} \tag{24}$$

For finding an optimum value of the auxiliary term  $h$ , the lowest point of the error of norm 2 of Function (24) is selected. The lowest point of  $e_{3a}(h)$  as  $h = 1.00000000005237$ , and  $e_{3b}(h)$  as  $h = 0.999999981901321$ , are gained by means of Maple mathematical software. By replacing the value of the auxiliary term in  $u_4(x, t, h)$ , the absolute error of the fourth-order approximation of the proposed algorithm decreases remarkably. A comparison is presented in the following tables with the Fourier pseudospectral method [1], meshfree interpolation method [2], Chebyshev spectral collocation method [3], decomposition method [4], cubic B-spline collocation scheme [5], new lattice Boltzmann model [6], and semi-Lagrangian approach [10].

In Tables 1 and 2, the  $L_2$  and  $L_\infty$  error norms are given for our proposed algorithm MVIA-II, MVIA-I, and finite difference method (FDM), as well as the lattice Boltzmann model (LBM) for  $u$  and  $v$ , respectively, while Table 3 shows the comparison of errors of different methods for showing the accuracy of our proposed method. We can see that MVIA-II produces more accurate results than both FDM and LBM. The  $L_2$  and  $L_\infty$  error norms for different values of  $t$  are reported in Tables 2 and 3, which show that the proposed method gives more accurate results than the previous methods that have been used before. Figures 1 and 2 display the behavior of exact and numerical solutions for  $u$  and  $v$  by MVIA-II. Figure 3 presents the absolute error graphs for different values of  $t$  for  $u$  and  $v$ , while Figure 4 is devoted to a plot solution at different time levels computed by MVIA-II. It can be seen that the method is uniformly convergent and gives accurate results. It is clear from the figures and tables that the proposed method can handle the coupled Burgers' equations very accurately.

**Table 1.** Comparison of numerical results of  $u$  for different values of  $t$  and  $x$  for Test Problem 1.

$t$	$\beta_{1,2}$	$\beta_{2,1}$	$L_2$					$L_\infty$				
			CBCS [5]	LBM [6]	CBSLM [10]	MVIA-II	MVIA-I	CBCS [5]	LBM [6]	CBSLM [10]	MVIA-II	MVIA-I
0.5	0.1	0.3	$6.7307 \times 10^{-4}$	$6.7736 \times 10^{-4}$	$5.7744 \times 10^{-4}$	$7.0326 \times 10^{-4}$	$2.2440 \times 10^{-4}$	$6.5947 \times 10^{-4}$	$6.6014 \times 10^{-4}$	$5.6194 \times 10^{-4}$	$6.7509 \times 10^{-4}$	$2.1840 \times 10^{-4}$
1.0	0.1	0.3	$1.3173 \times 10^{-3}$	$1.3317 \times 10^{-3}$	$0.2346 \times 10^{-3}$	$1.4073 \times 10^{-3}$	$4.4886 \times 10^{-4}$	$1.3020 \times 10^{-3}$	$1.3045 \times 10^{-3}$	$1.2083 \times 10^{-3}$	$1.3500 \times 10^{-3}$	$4.3681 \times 10^{-4}$
5.0	0.1	0.3	$5.9329 \times 10^{-3}$	$6.1433 \times 10^{-3}$	$6.0584 \times 10^{-3}$	$7.0707 \times 10^{-3}$	$2.2464 \times 10^{-3}$	$6.1046 \times 10^{-3}$	$6.1515 \times 10^{-3}$	$6.0644 \times 10^{-3}$	$6.7445 \times 10^{-3}$	$2.1844 \times 10^{-3}$
10	0.1	0.3	$1.0760 \times 10^{-2}$	$1.1439 \times 10^{-2}$	$1.1365 \times 10^{-2}$	$1.4221 \times 10^{-2}$	$4.4981 \times 10^{-3}$	$1.1541 \times 10^{-2}$	$1.1713 \times 10^{-2}$	$1.1633 \times 10^{-2}$	$1.3468 \times 10^{-2}$	$4.3694 \times 10^{-3}$

**Table 2.** Comparison of numerical results for  $v$  for different values of  $t$  and  $x$  for Test Problem 1.

$t$	$\beta_{1,2}$	$\beta_{2,1}$	$L_2$					$L_\infty$				
			CBCS [5]	LBM [6]	CBSLM [10]	MVIA-II	MVIA-I	CBCS [5]	LBM [6]	CBSLM [10]	MVIA-II	MVIA-I
0.5	0.1	0.3	$5.0960 \times 10^{-4}$	$5.0505 \times 10^{-4}$	$4.3123 \times 10^{-4}$	$5.1860 \times 10^{-4}$	$2.5955 \times 10^{-4}$	$9.1286 \times 10^{-4}$	$5.6806 \times 10^{-4}$	$4.8567 \times 10^{-4}$	$6.2517 \times 10^{-4}$	$2.5169 \times 10^{-4}$
1.0	0.1	0.3	$9.9228 \times 10^{-4}$	$9.8477 \times 10^{-4}$	$9.1399 \times 10^{-4}$	$1.0389 \times 10^{-3}$	$5.1920 \times 10^{-4}$	$1.8244 \times 10^{-3}$	$1.0942 \times 10^{-3}$	$1.0169 \times 10^{-3}$	$1.2509 \times 10^{-3}$	$5.0343 \times 10^{-4}$
5.0	0.1	0.3	$4.3800 \times 10^{-3}$	$4.3984 \times 10^{-3}$	$4.3400 \times 10^{-3}$	$5.2635 \times 10^{-3}$	$2.5996 \times 10^{-3}$	$9.0708 \times 10^{-3}$	$4.6640 \times 10^{-3}$	$4.6046 \times 10^{-3}$	$6.2794 \times 10^{-3}$	$2.5190 \times 10^{-3}$
10	0.1	0.3	$7.8592 \times 10^{-3}$	$8.0429 \times 10^{-3}$	$7.9933 \times 10^{-3}$	$7.8956 \times 10^{-2}$	$5.2084 \times 10^{-3}$	$1.0815 \times 10^{-2}$	$8.2714 \times 10^{-3}$	$8.2233 \times 10^{-3}$	$7.9438 \times 10^{-2}$	$5.0426 \times 10^{-3}$

**Table 3.** Comparison of absolute errors for different values of  $t$  and  $x$  for Test Problem 1.

$u/v$	$t$	$\beta_{1,2}$	$\beta_{2,1}$	[9]	[3]	[5]	[1]	[2]	MVIA-II	MVIA-I
$u$	0.5	0.1	0.3	$4:251 \times 10^{-5}$	$4:38 \times 10^{-5}$	$4:167 \times 10^{-5}$	$9:619 \times 10^{-4}$	$4:084 \times 10^{-5}$	$2.2743 \times 10^{-4}$	$1.1511 \times 10^{-5}$
	1.0	0.1	0.3	$8:150 \times 10^{-5}$	$8:66 \times 10^{-5}$	$8:258 \times 10^{-5}$	$1:153 \times 10^{-3}$	$8:157 \times 10^{-5}$	$4.5485 \times 10^{-4}$	$2.3024 \times 10^{-5}$
$v$	0.5	0.1	0.3	$4:051 \times 10^{-5}$	$4:99 \times 10^{-5}$	$1:480 \times 10^{-5}$	$3:332 \times 10^{-4}$	$3:713 \times 10^{-5}$	$2.1639 \times 10^{-4}$	$6.9734 \times 10^{-6}$
	1.0	0.1	0.3	$7:158 \times 10^{-5}$	$9:92 \times 10^{-5}$	$4:770 \times 10^{-5}$	$1:162 \times 10^{-3}$	$7:358 \times 10^{-5}$	$4.3275 \times 10^{-4}$	$1.3949 \times 10^{-5}$

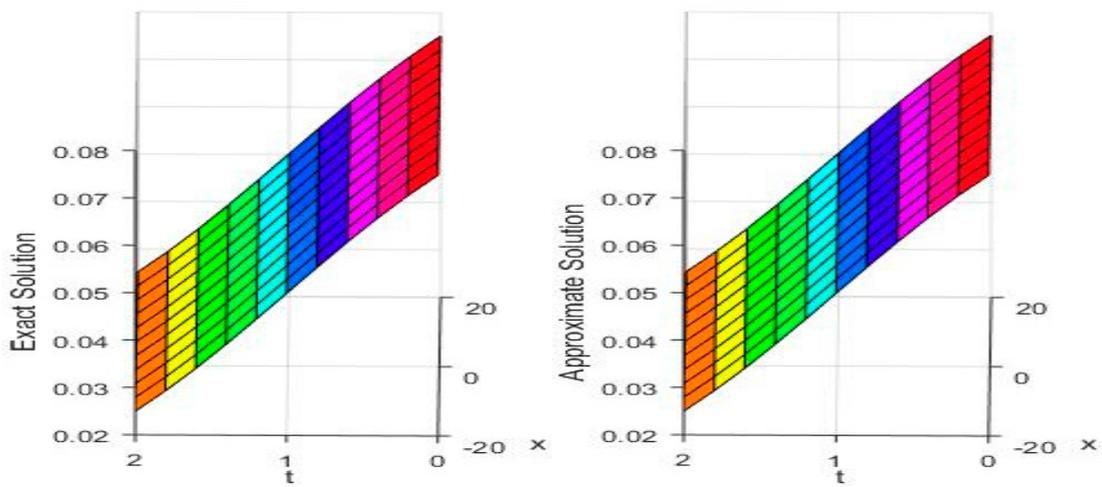


Figure 1. Behavior of exact (Left) and approximate solutions (Right) of  $u$  for Test Problem 1.

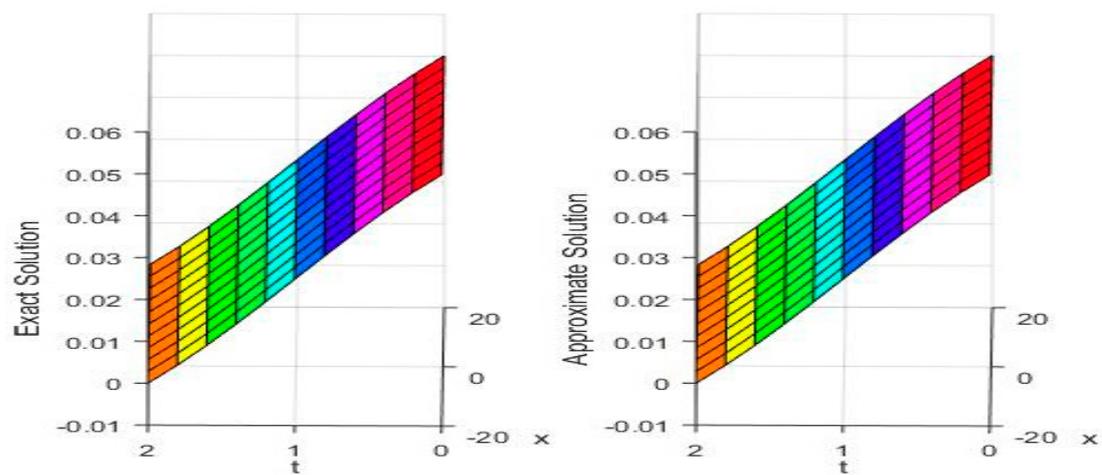


Figure 2. Behavior of exact (Left) and approximate solutions (Right) of  $v$  for Test Problem 1.

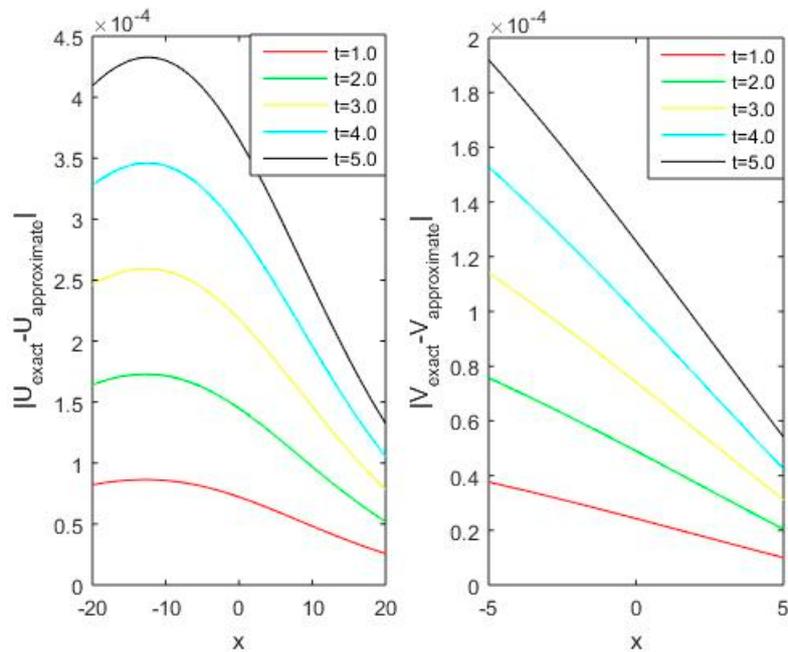
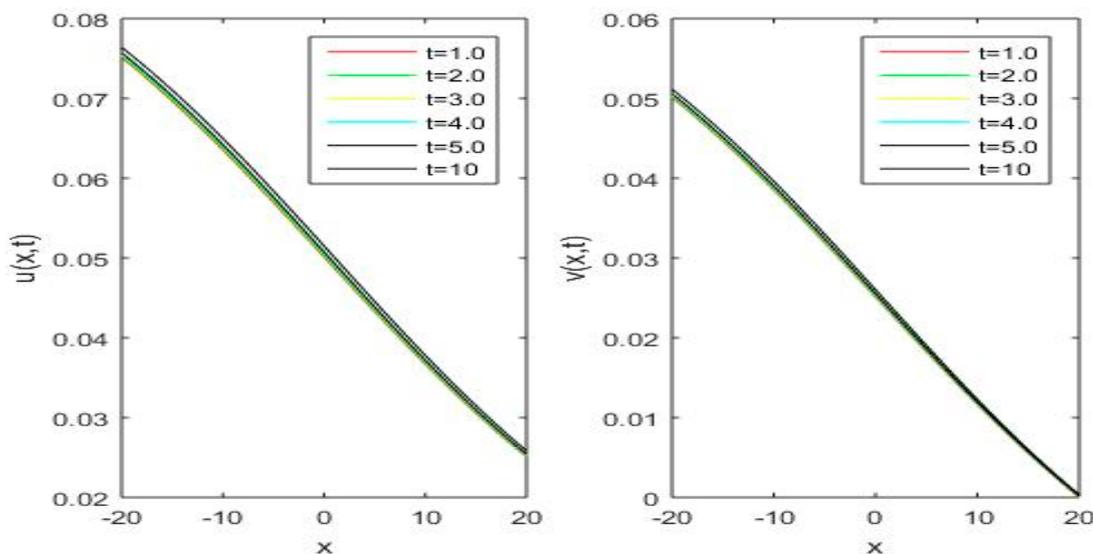


Figure 3. Comparison of absolute errors for different values of  $t$  for Test Problem 1.



**Figure 4.** Approximation solution of  $u$  (Left) and approximation solution of  $v$  (Right) for different values of  $t$ .

3.2. Test Problem 2

Consider the coupled Burgers’ Equation (1) with  $\alpha_1 = \alpha_2 = 1, \beta_{1,1} = \beta_{2,2} = -2, \beta_{1,2} = \beta_{2,1} = 2.5$  and  $x \in [-20, 20]$ . The equation becomes:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - 2\frac{\partial u}{\partial x} + 2.5\frac{\partial uv}{\partial x} = 0, & t > 0, \quad x \in [-20, 20], \\ \frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial x^2} - 2\frac{\partial v}{\partial x} + 2.5\frac{\partial uv}{\partial x} = 0, & t > 0, \quad x \in [-20, 20], \end{cases} \tag{25}$$

The exact solution was given by [10]:

$$u(x, t) = v(x, t) = k(1 - \tanh(\frac{3}{2}k(x - 3kt))). \tag{26}$$

where  $k$  is a non-zero arbitrary constant. We noticed that when the value of  $k$  increases, then the traveling waves become faster as time goes on, and the waveforms become increasingly steeper.

Constructing the correction function for Equation (25) as:

$$\begin{aligned} u_{n+1}(x, t, h) = & u_n(x, t, h) \\ & + h \int_0^t \lambda_1(\zeta) \left\{ \frac{\partial u_n(x, \zeta, h)}{\partial \zeta} - \frac{\partial^2 \widetilde{u_n(x, \zeta, h)}}{\partial x^2} - 2\frac{\partial \widetilde{u_n(x, \zeta, h)}}{\partial x} \right. \\ & \left. + 2.5\frac{\partial \widetilde{u_n(x, \zeta, h)v_n(x, \zeta, h)}}{\partial x} \right\} d\zeta. \end{aligned} \tag{27}$$

$$\begin{aligned} v_{n+1}(x, t, h) = & v_n(x, t, h) \\ & + h \int_0^t \lambda_2(\zeta) \left\{ \frac{\partial v_n(x, \zeta, h)}{\partial \zeta} - \frac{\partial^2 \widetilde{v_n(x, \zeta, h)}}{\partial x^2} - 2\frac{\partial \widetilde{v_n(x, \zeta, h)}}{\partial x} \right. \\ & \left. + 2.5\frac{\partial \widetilde{u_n(x, \zeta, h)v_n(x, \zeta, h)}}{\partial x} \right\} d\zeta. \end{aligned} \tag{28}$$

The values of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$  may be obtained most positively by the variational principle. We obtain the estimation of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$ , which is  $\lambda_1(\zeta) = \lambda_2(\zeta) = -1$ . Utilizing this estimation of  $\lambda(\zeta)$  in Equations (27) and (28) results in the underneath iterative structure:

$$u_{n+1}(x, t, h) = u_0(x, t, h) - h \int_0^t \left\{ -\frac{\partial^2 u_n(x, \zeta, h)}{\partial x^2} - 2 \frac{\partial u_n(x, \zeta, h)}{\partial x} + 2.5 \frac{\partial u_n(x, \zeta, h) v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \tag{29}$$

$$v_{n+1}(x, t, h) = v_0(x, t, h) - h \int_0^t \left\{ -\frac{\partial^2 v_n(x, \zeta, h)}{\partial x^2} - 2 \frac{\partial v_n(x, \zeta, h)}{\partial x} + 2.5 \frac{\partial u_n(x, \zeta, h) v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \tag{30}$$

Introducing with a proper initial guess:

$$u_0(x, t) = v_0(x, t) = k(1 - \tanh\left(\frac{3}{2}kx\right)).$$

one can get beneath different iterations by utilizing the recurrence relations shown in Equations (29) and (30), for  $k = 0.1$ :

$$\begin{aligned} u_1(x, t, h) &= (9 * k^3 * t) / (2 * \cosh((3 * k * x) / 2)^2) - k * (\tanh((3 * k * x) / 2) - 1), \\ v_1(x, t, h) &= (9 * k^3 * t) / (2 * \cosh((3 * k * x) / 2)^2) - k * (\tanh((3 * k * x) / 2) - 1), \\ u_2(x, t, h) &= (9 * k^3 * t * (2 * \cosh((3 * k * x) / 2)^3 + 27 * k^4 * t^2 * \sinh((3 * k * x) / 2) + 9 * k^2 * t * \cosh((3 * k * x) / 2)^2 * \sinh((3 * k * x) / 2))) / (4 * \cosh((3 * k * x) / 2)^5) \\ &\quad - k * (\tanh((3 * k * x) / 2) - 1), \\ v_2(x, t, h) &= (9 * k^3 * t * (2 * \cosh((3 * k * x) / 2)^3 + 27 * k^4 * t^2 * \sinh((3 * k * x) / 2) + 9 * k^2 * t * \cosh((3 * k * x) / 2)^2 * \sinh((3 * k * x) / 2))) / (4 * \cosh((3 * k * x) / 2)^5) \\ &\quad - k * (\tanh((3 * k * x) / 2) - 1), \\ &\vdots \\ &\vdots \end{aligned}$$

we stop the procedure at  $u_4(x, t, h)$ . In order to find an appropriate value of  $h$  for the approximate solutions, a residual function can be defined as:

$$r_4(x, t, h) = \frac{\partial u_4(x, t, h)}{\partial t} - \frac{\partial^2 u_4(x, t, h)}{\partial x^2} - 2 \frac{\partial u_4(x, t, h)}{\partial x} + 2.5 \frac{\partial u_4(x, t, h) v_4(x, t, h)}{\partial x} \tag{31}$$

The square of the  $r_4(x, t, h)$  function with respect to the auxiliary term  $h$  in the domain  $(x, t) \in [0, 1] \times [0, 1]$  is:

$$e_4(h) = \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=0}^{10} \left( r_4\left(\frac{i}{10}, \frac{j}{10}, h\right) \right)^2 \right]^{\frac{1}{2}} \tag{32}$$

For finding an optimum value of the auxiliary term  $h$ , the lowest point of the error of norm 2 of Function (31) is selected. The lowest point of  $e_4(h)$ , as  $h = 1.00000000005237$ , is gained by means of Maple mathematical software. By replacing the value of the auxiliary term in  $u_4(x, t, h)$ , the absolute error of the fourth-order approximation of the proposed algorithm decreases remarkably.

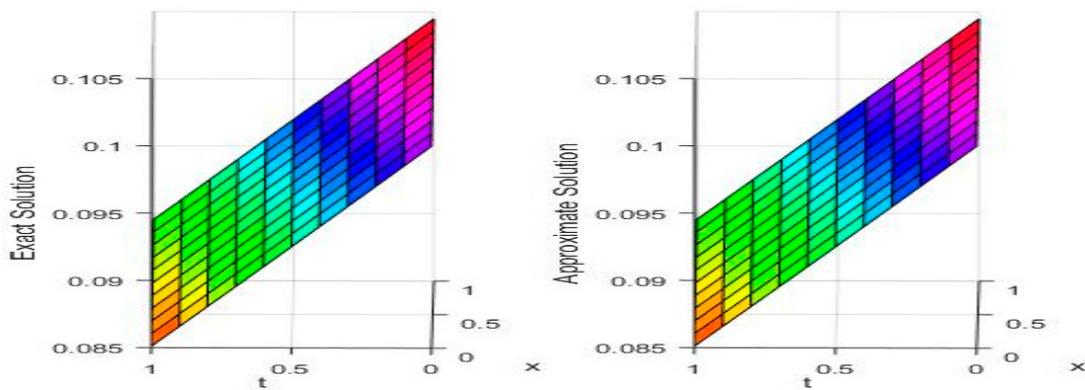
In Tables 4 and 5,  $L_2$  and  $L_\infty$  error norms are given for our proposed algorithm MVIA-II, and other methods available in the literature. We can see that MVIA-II produces more accurate results than FDM, LBM, and CBSLM. The  $L_2$  and  $L_\infty$  error norms for different values of  $t$  are reported in Table 5, which shows that the proposed method gives more accurate results than previous methods used. Figure 5 displays the behavior of the exact and numerical solutions, while Figure 6 is devoted to a plot absolute error graph for different values of  $t$ . The method is uniformly convergent and gives accurate results. It is clear from the figures and tables that the proposed method can handle the coupled Burgers' equations very accurately.

**Table 4.** Comparison of error norms different values of  $t$  and  $x$  for Test Problem 2.

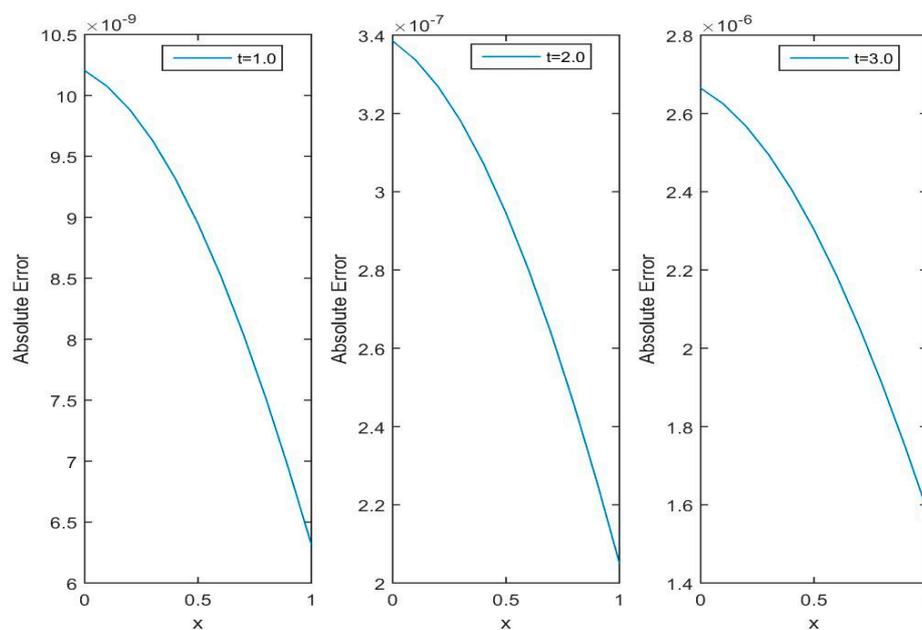
$t$	$L_2$			$L_\infty$		
	FDM [6]	LBM [6]	MVIA-II	FDM [6]	LBM [6]	MVIA-II
1.0	$1.5724 \times 10^{-6}$	$1.4829 \times 10^{-6}$	$7.8577 \times 10^{-10}$	$6.2856 \times 10^{-7}$	$5.7788 \times 10^{-7}$	$6.1316 \times 10^{-13}$
2.0	$2.9383 \times 10^{-6}$	$2.7955 \times 10^{-6}$	$2.6625 \times 10^{-10}$	$1.1138 \times 10^{-6}$	$1.0754 \times 10^{-6}$	$1.5700 \times 10^{-9}$
3.0	$4.1676 \times 10^{-6}$	$3.9298 \times 10^{-6}$	$8.7801 \times 10^{-9}$	$1.5879 \times 10^{-6}$	$1.4861 \times 10^{-6}$	$5.1132 \times 10^{-8}$
4.0	$5.2504 \times 10^{-6}$	$4.9434 \times 10^{-6}$	$2.9783 \times 10^{-7}$	$1.9868 \times 10^{-6}$	$1.8800 \times 10^{-6}$	$1.6959 \times 10^{-6}$
5.0	$6.1878 \times 10^{-6}$	$5.8615 \times 10^{-6}$	$3.4271 \times 10^{-5}$	$2.3468 \times 10^{-6}$	$2.2034 \times 10^{-6}$	$1.8436 \times 10^{-4}$

**Table 5.** Comparison of error norms different values of  $t$  and  $x$  for Test Problem 2.

u/v	$t$	$L_2$			$L_\infty$		
		LBM [6]	CBSLM [10]	MVIA-II	LBM [6]	CBSLM [10]	MVIA-II
u	1	$2.992 \times 10^{23}$	$3.0876 \times 10^{-7}$	$8.96888 \times 10^{-9}$	$2.6797 \times 10^{24}$	$5.324 \times 10^{-7}$	$5.1556 \times 10^{-8}$
	2	NaN	$6.2561 \times 10^{-7}$	$3.03807 \times 10^{-7}$	NaN	$1.056 \times 10^{-6}$	$1.7139 \times 10^{-6}$
v	1	$2.992 \times 10^{23}$	$8.2321 \times 10^{-8}$	$8.96888 \times 10^{-9}$	$2.6797 \times 10^{24}$	$1.479 \times 10^{-7}$	$5.1556 \times 10^{-8}$
	2	NaN	$1.6134 \times 10^{-7}$	$3.03807 \times 10^{-7}$	NaN	$2.639 \times 10^{-7}$	$1.7139 \times 10^{-6}$



**Figure 5.** Behavior of exact (Left) and approximate solution (Right) for Test Problem 2.



**Figure 6.** Comparison of exact and approximate solution for Test Problem 2.

3.3. Test Problem 3

Consider the coupled Burger Equation (1) with  $\alpha_1 = \alpha_2 = 1, \beta_{1,1} = \beta_{2,2} = -2, \beta_{1,2} = \beta_{2,1} = 1$  and  $x \in [-\pi, \pi]$ . The equation becomes:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - 2u \frac{\partial u}{\partial x} + \frac{\partial uv}{\partial x} = 0, & t > 0, \quad x \in [-\pi, \pi], \\ \frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial x^2} - 2v \frac{\partial v}{\partial x} + \frac{\partial uv}{\partial x} = 0, & t > 0, \quad x \in [-\pi, \pi], \end{cases} \tag{33}$$

The exact solution was given by [10]:

$$u(x, t) = v(x, t) = \exp(-t)\sin(x) \tag{34}$$

where  $\beta_{1,2}$  and  $\beta_{2,1}$  are non-zero arbitrary constants. Constructing the correction function for Equation (33) as:

$$\begin{aligned} u_{n+1}(x, t, h_1) = & u_0(x, t, h_1) \\ & + h_1 \int_0^t \lambda_1(\zeta) \left\{ \frac{\partial u_n(x, \zeta, h_1)}{\partial \zeta} - \frac{\partial^2 u_n(x, \zeta, h_1)}{\partial x^2} - 2 \frac{\partial u_n(x, \zeta, h_1)}{\partial x} \right. \\ & \left. + \frac{\partial u_n(x, \zeta, h_1) v_n(x, \zeta, h_1)}{\partial x} \right\} d\zeta. \end{aligned} \tag{35}$$

$$\begin{aligned} v_{n+1}(x, t, h) = & v_0(x, t, h) \\ & + h \int_0^t \lambda_2(\zeta) \left\{ \frac{\partial v_n(x, \zeta, h)}{\partial \zeta} - \frac{\partial^2 v_n(x, \zeta, h)}{\partial x^2} - 2 \frac{\partial v_n(x, \zeta, h)}{\partial x} \right. \\ & \left. + \frac{\partial u_n(x, \zeta, h) v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \end{aligned} \tag{36}$$

The values of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$  may be obtained most positively by the variational principle. We obtain the estimation of  $\lambda_1(\zeta)$  and  $\lambda_2(\zeta)$  which is  $\lambda_1(\zeta) = \lambda_2(\zeta) = -1$ . Utilizing this estimation of  $\lambda(\zeta)$  in Equations (35) and (36) results in the underneath iterative structure:

$$\begin{aligned} u_{n+1}(x, t, h_1) = & u_0(x, t, h_1) \\ & - h_1 \int_0^t \left\{ -\frac{\partial^2 u_n(x, \zeta, h_1)}{\partial x^2} - 2 \frac{\partial u_n(x, \zeta, h_1)}{\partial x} + \frac{\partial u_n(x, \zeta, h_1) v_n(x, \zeta, h_1)}{\partial x} \right\} d\zeta. \end{aligned} \tag{37}$$

$$v_{n+1}(x, t, h) = v_0(x, t, h) - h \int_0^t \left\{ -\frac{\partial^2 v_n(x, \zeta, h)}{\partial x^2} - 2 \frac{\partial v_n(x, \zeta, h)}{\partial x} + \frac{\partial u_n(x, \zeta, h) v_n(x, \zeta, h)}{\partial x} \right\} d\zeta. \tag{38}$$

Introducing with a proper initial guess:

$$u(x, 0) = v(x, 0) = \sin(x),$$

one can get the beneath different approximations by utilizing the recurrence relations shown in Equations (37) and (38):

$$\begin{aligned} u_1(x, t, h) &= -\sin(x) * (h * t - 1), \\ v_1(x, t, h) &= -\sin(x) * (h * t - 1), \\ u_2(x, t, h) &= (\sin(x) * (h^2 * t^2 - 2 * h * t + 2)) / 2, \\ v_2(x, t, h) &= (\sin(x) * (h^2 * t^2 - 2 * h * t + 2)) / 2, \\ u_3(x, t, h) &= -(\sin(x) * (h^3 * t^3 - 3 * h^2 * t^2 + 6 * h * t - 6)) / 6, \\ v_3(x, t, h) &= -(\sin(x) * (h^3 * t^3 - 3 * h^2 * t^2 + 6 * h * t - 6)) / 6, \\ &\vdots \\ &\vdots \end{aligned}$$

we stop the procedure at the 11th-order approximation. In order to find an appropriate value of  $h$  for the approximate solutions, a residual function can be defined as:

$$\begin{aligned}
 r_{11a}(x, t, h) &= \frac{\partial u_{11}(x, t, h)}{\partial t} - \frac{\partial^2 u_{11}(x, t, h)}{\partial x^2} - 2 \frac{\partial u_{11}(x, t, h)}{\partial x} + \frac{\partial u_{11}(x, t, h) v_{11}(x, t, h)}{\partial x} \\
 r_{11b}(x, t, h) &= \frac{\partial u_{11}(x, t, h)}{\partial t} - \frac{\partial^2 u_{11}(x, t, h)}{\partial x^2} - 2 \frac{\partial u_{11}(x, t, h)}{\partial x} + \frac{\partial u_{11}(x, t, h) v_{11}(x, t, h)}{\partial x}
 \end{aligned}
 \tag{39}$$

The square of the  $r_{11a}(x, t, h)$  and  $r_{11b}(x, t, h)$  functions with respect to the auxiliary term  $h$  in the domain  $x \in [-\pi, \pi]$  is:

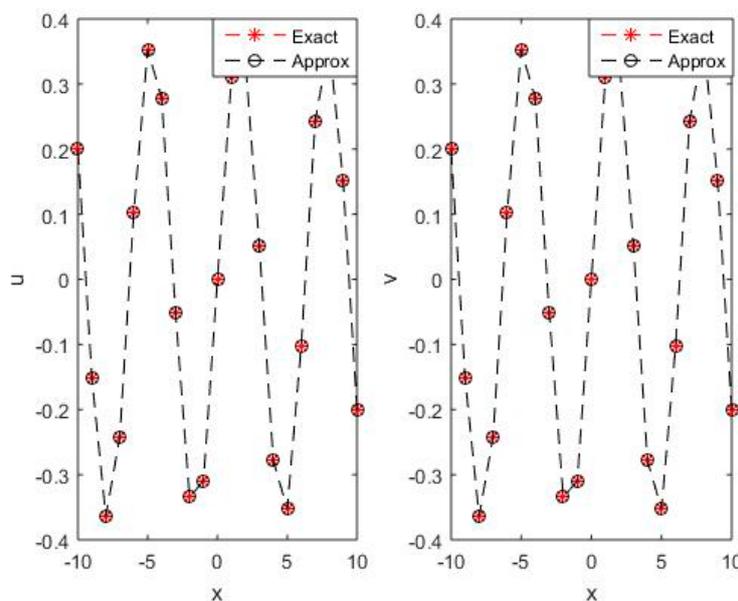
$$\begin{aligned}
 e_{11a}(h) &= \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=0}^{10} (r_{11}(\frac{i}{10}, \frac{j}{10}, h))^2 \right]^{\frac{1}{2}} \\
 e_{11b}(h) &= \left[ \frac{1}{(11)^2} \sum_{i=0}^{10} \sum_{j=-\pi}^{\pi} (r_{11}(\frac{i}{10}, \frac{j}{10}, h))^2 \right]^{\frac{1}{2}}
 \end{aligned}
 \tag{40}$$

For finding an optimum value of the auxiliary term  $h$ , the lowest point of the error of norm 2 of Function (40) is selected. The lowest point of  $e_{11}(h)$ , as  $h = 1.00000000005237$ , is gained by means of Maple mathematical software. By replacing the value of the auxiliary term in  $u_{11}(x, t, h)$ , the absolute error of the 11th-order approximation of the proposed algorithm decreases remarkably.

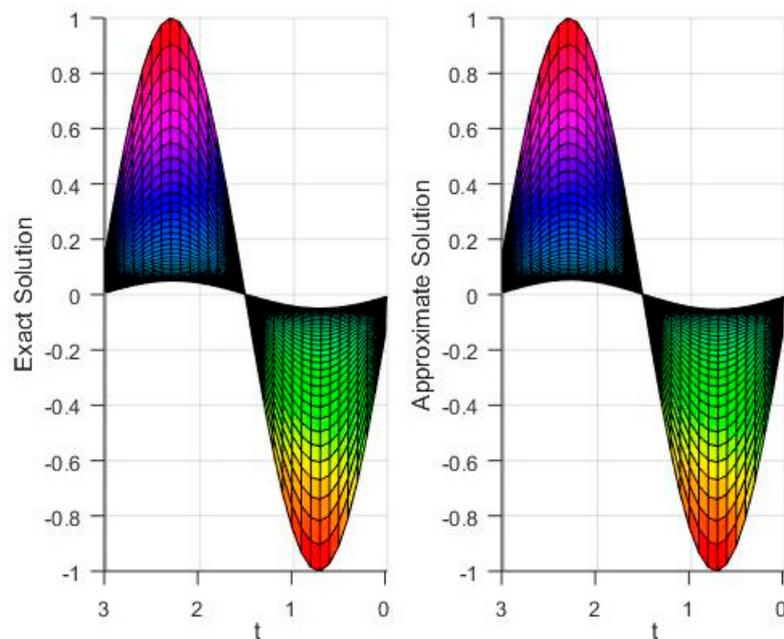
In Table 6,  $L_2$  and  $L_\infty$  error norms are given for our proposed algorithm MVIA-II and other numerical methods available in the literature. We can see that MVIA-II produces more accurate results than the others. Figure 7 displays the comparison of exact and numerical solutions of  $u$  and  $v$  for  $t = 1$ . Figure 8 is devoted to showing the behavior of exact and approximate solutions for  $u$  and  $v$ . It can be seen that the method is uniformly convergent and gives accurate results. It is clear from the figures and tables that the proposed method can handle the coupled Burgers' equations very accurately.

**Table 6.** Comparison of  $L_2$  and  $L_\infty$  error norms for different values of  $t$  for Test Problem 3.

$t$	$L_2$			$L_\infty$		
	[9]	[5]	MVIA-II	[9]	[5]	MVIA-II
0.1	$2.5 \times 10^{-6}$	$2.05 \times 10^{-6}$	$5.33 \times 10^{-11}$	$2.5 \times 10^{-6}$	$1.86 \times 10^{-6}$	$4.81 \times 10^{-11}$
0.5	$8.8 \times 10^{-6}$	$1.02 \times 10^{-5}$	$2.46 \times 10^{-10}$	$4.8 \times 10^{-6}$	$6.22 \times 10^{-6}$	$1.49 \times 10^{-11}$
1.0	$1.1 \times 10^{-5}$	$2.04 \times 10^{-5}$	$6.22 \times 10^{-8}$	$3.7 \times 10^{-5}$	$7.56 \times 10^{-6}$	$2.29 \times 10^{-8}$



**Figure 7.** Comparison of the exact and approximate solution for  $u$  and  $v$  for  $t = 1$  for Test Problem 3.



**Figure 8.** Comparison of the behavior of the exact and approximate solution for Test Problem 3.

#### 4. Conclusions

In this paper, we have proposed a technique to acquire numerical simulations of coupled Burgers' equations by modifying the variational iteration algorithm-II. The technique effectively gives extremely precise solutions using various values of parameters. The applicability, performance, and capability of the algorithm have been investigated over the presented test examples, either by ascertaining the maximum, relative, and absolute error norms for various time levels, or by contrasting the idea of numerical results with the nature of the solutions accessible in the literature. We conclude that the proposed algorithm provides an accurate numerical/analytical solution and can handle nonlinear coupled PDEs in a good and reliable manner in various cases.

The obtained results are acceptable and capable with the results that are accessible in the literature. The modified algorithm can be utilized without any need for discretization, linearization, or complex and lengthy calculations; rather, it is a simple solution process. The proposed algorithm is reasonable, has the easiest implementation, and is reliable as well. It can handle a large class of similar nonlinear coupled equations, which often arises in science and engineering.

**Author Contributions:** Conceptualization, H.A.; methodology, H.A.; software, H.A.; validation, H.A., T.A.K. and C.C.; formal analysis, H.A.; investigation, H.A.; resources, H.A.; data curation, H.A.; writing—original draft preparation, H.A.; writing—review and editing, H.A., T.A.K. and C.C.; visualization, H.A.; supervision, T.A.K. and C.C.; project administration, T.A.K. and C.C.; funding acquisition, C.C.

**Acknowledgments:** The first author thanks the Siraj-ul-Islam, Dean of faculty of allied Sciences, University of engineering & technology Peshawar, for his indispensable guidance, sympathetic encouragement, valuable suggestions and generous help during the course of study. All the authors thankfully acknowledge the fruitful comments and significant remarks of the Editor-in-Chief and unknown referees, which helped them improve the quality and readability of the paper and led to a significant improvement of the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Rashid, A.; Ismail, A.I.B.M. A Fourier Pseudospectral Method for Solving Coupled Viscous Burgers Equations. *Comput. Methods Appl. Math.* **2009**, *9*, 412–420. [[CrossRef](#)]
2. Islam, S.-U.; Haq, S.; Uddin, M. A meshfree interpolation method for the numerical solution of the coupled nonlinear partial differential equations. *Eng. Anal. Bound. Elem.* **2009**, *33*, 399–409. [[CrossRef](#)]

3. Khater, A.; Tamsah, R.; Hassan, M. A Chebyshev spectral collocation method for solving Burgers'-type equations. *J. Comput. Appl. Math.* **2008**, *222*, 333–350. [[CrossRef](#)]
4. Kaya, D. An explicit solution of coupled viscous Burgers' equation by the decomposition method. *Int. J. Math. Math. Sci.* **2001**, *27*, 675–680. [[CrossRef](#)]
5. Mittal, R.; Arora, G. Numerical solution of the coupled viscous Burgers' equation. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 1304–1313. [[CrossRef](#)]
6. Lai, H.; Ma, C. A new lattice Boltzmann model for solving the coupled viscous Burgers' equation. *Phys. A: Stat. Mech. Appl.* **2014**, *395*, 445–457. [[CrossRef](#)]
7. Rashid, A.; Abbas, M.; Ismail, A.I.M.; Majid, A.A. Numerical solution of the coupled viscous Burgers equations by Chebyshev–Legendre Pseudo-Spectral method. *Appl. Math. Comput.* **2014**, *245*, 372–381. [[CrossRef](#)]
8. Kumar, M.; Pandit, S. A composite numerical scheme for the numerical simulation of coupled Burgers' equation. *Comput. Phys. Commun.* **2014**, *185*, 809–817. [[CrossRef](#)]
9. Mohammadi, M.; Mokhtari, R. A reproducing kernel method for solving a class of nonlinear systems of pdes. *Math. Model. Anal.* **2014**, *19*, 180–198. [[CrossRef](#)]
10. Bak, S.; Kim, P.; Kim, D. A semi-Lagrangian approach for numerical simulation of coupled Burgers' equations. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *69*, 31–44. [[CrossRef](#)]
11. Inokuti, M.; Sekine, H.; Mura, T. *General Use of the Lagrange Multiplier in Nonlinear Mathematical Physics*; Pergamon Press: Oxford, UK, 1978.
12. He, J.-H.; Wu, G.-C.; Austin, F. The variational iteration method which should be followed. *Nonlinear Sci. Lett. A* **2010**, *1*, 1–30.
13. Ahmad, H. Variational Iteration Algorithm-I with an Auxiliary Parameter for Solving Fokker-Planck Equation. *Earthline J. Math. Sci.* **2019**, *2*, 29–37. [[CrossRef](#)]
14. Ahmad, H.; Khan, T.A. Variational iteration algorithm-I with an auxiliary parameter for wave-like vibration equations. *J. Low Freq. Noise Vib. Act. Control* **2019**, *38*, 1113–1124. [[CrossRef](#)]
15. Anjum, N.; He, J.-H. Laplace transform: Making the variational iteration method easier. *Appl. Math. Lett.* **2019**, *92*, 134–138. [[CrossRef](#)]
16. Yu, D.-N.; He, J.-H.; Garcia, A.G. Homotopy perturbation method with an auxiliary parameter for nonlinear oscillators. *J. Low Freq. Noise Vib. Act. Control* **2018**, *38*, 1540–1554. [[CrossRef](#)]
17. He, J.-H. Variational approach to the Thomas–Fermi equation. *Appl. Math. Comput.* **2003**, *143*, 533–535. [[CrossRef](#)]
18. He, J.-H. Variational iteration method—Some recent results and new interpretations. *J. Comput. Appl. Math.* **2007**, *207*, 3–17. [[CrossRef](#)]
19. He, J.-H.; Wu, X.-H. Variational iteration method: New development and applications. *Comput. Math. Appl.* **2007**, *54*, 881–894. [[CrossRef](#)]
20. Nadeem, M.; Ahmad, H. Variational Iteration Method for Analytical Solution of the Lane-Emden Type Equation with Singular Initial and Boundary Conditions. *Earthline J. Math. Sci.* **2019**, 127–142. [[CrossRef](#)]
21. Ahmad, H. Variational iteration method with an auxiliary parameter for solving differential equations of the fifth order. *Nonlinear Sci. Lett. A* **2018**, *9*, 27–35.
22. Rafiq, M.; Ahmad, H.; Mohyud-Din, S.T. Variational iteration method with an auxiliary parameter for solving Volterra's population model. *Nonlinear Sci. Lett. A* **2017**, *8*, 389–396.
23. Nadeem, M.; Li, F.; Ahmad, H. Modified Laplace variational iteration method for solving fourth-order parabolic partial differential equation with variable coefficients. *Comput. Math. Appl.* **2019**, *78*, 2052–2062. [[CrossRef](#)]
24. He, J.-H. Lagrange crisis and generalized variational principle for 3D unsteady flow. *Int. J. Numer. Methods Heat Fluid Flow* **2019**. [[CrossRef](#)]
25. He, J.-H.; Sun, C. A variational principle for a thin film equation. *J. Math. Chem.* **2019**, *57*, 2075–2081. [[CrossRef](#)]

