A Numerical Solution of Burger's Equation Based on Milne Method

Supranee Chonladed and Kanognudge Wuttanachamsri

Abstract—Burger's equation is a nonlinear parabolic partial differential equation used in several fields such as fluid dynamics and traffic flow. In this research, we find the numerical solution of the one-dimensional Burger's equation by using the multi-step Milne method and the central finite difference approach. A linearization scheme with a weighted technique is employed to handle the nonlinear term. Due to the multistep approach, the second-order Runge-Kutta and Modified-Newton Raphson schemes are applied to determine the second initial condition. The numerical results are compared with the exact solution, where the L_2 and L_∞ norms of the errors are used to verify the accuracy of the techniques. The numerical solutions are in good agreement with the exact solution. Among various different variables in the governing equation and the initial condition, the numerical visualizations are provided for the different values of the parameters.

Index Terms—Burger's equation, Milne method, Finite difference method, Runge-Kutta method, Modified-Newton Raphson method.

I. INTRODUCTION

B URGER'S equation is a fundamental partial differential equation. It was firstly given by Harry Bateman in 1915 [1]. The Burger's equation becomes one of the leading equations in the field of fluid mechanics which was suitable for the analysis of various important areas such as modeling of gas dynamics, heat conduction, and traffic flow [2].

Burger's equation is solved by numerous numerical methods such as the finite difference and the finite element methods [3]-[16]. For example, A. Chaiyasit et al. [3] provided the numerical solution of the one-dimensional Burger's equation by using the second-order Runge-Kutta method, the central finite difference scheme, and the Newton-Raphson approach. Z.Y. Ali [4] used a new iterative method to find the solution of the Burger's equation. N.A. Mohamed [5], provided a new numerical scheme based on the finite difference method for solving the nonlinear one-dimensional Burgers' equation and N.A. Mohamed [6] introduced new fully implicit schemes for solving the one-dimensional and two-dimensional unsteady Burger's equation. P.G. Zang and J.P. Wang [7] proposed a compact predictor-corrector finite difference scheme to solve the Burger's equation. S.M. Zulkifli et al. [8] used inviscid Burger's equation to model traffic flow and find the solution of one-way traffic flow by using the method of linear system. K. Ali et al. [9] found the new exact solution of the Burger's equation by using the Hopf-Cole transform and the Fourier transform. Y. Ucar et al. [10] obtained the numerical solution of the modified Burger's equation

Manuscript received September 28, 2020; revised April 14, 2021.

K. Wuttanachamsri is a faculty at the Department of Mathematics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang Bangkok, 10520 THAILAND e-mail: kanognudge.wu@kmitl.ac.th

S. Chonladed is an M.S. student at the Department of Mathematics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang Bangkok, 10520 THAILAND e-mail: 62605015@kmitl.ac.th by using the finite difference methods. G. Celikten et al. [11] found the numerical solution of the modified Burger's equation by using the explicit exponential finite difference schemes based on four different linearization techniques. S. Sungnul et al. [12] found the numerical solutions of the modified Burger's equation by using the Forward Time Centered Space (FTCS) implicit scheme. M.A. Sheikh et al. [13] compared the numerical solutions of the Burger's equation by using Lax-Friedrich and Lax-Wendroff schemes. D. Deng and J. Xie [14] used Crank-Nicolson method combined with Richardson extrapolation scheme and a fourthorder compact finite difference method for solving the onedimensional Burgers equation. D. Deng and T. Pan [15] applied the fourth-order methods of lines (MOL) based on the Hopf-Cole transformation to solve the one-dimensional Burger's equation. P.W. Li [16] used a generalized finite difference approach and the Newton's method to solve the two-dimensional unsteady Burger's equation.

In this research, the Milne method combining with the finite difference scheme is used to solve the one-dimensional Burger's equations. Because of the multi-step approach, Runge-Kutta and Modified-Newton Raphson methods are applied to the Burger's equation to determine another initial condition. The structure of this paper is as follows. In Section 2, we provide Burger's equation, boundary conditions, and an initial condition. In Section 3, the solution procedure is described. The exact solution is given in Section 4. The numerical results of the Burger's equation are presented in Section 5. Conclusions are given in Section 6.

II. BURGER'S EQUATION

In this work, we consider the one-dimensional timedependent Burger's equation

$$\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} = \mu \frac{\partial^2 v}{\partial x^2}, \qquad \alpha < x < \beta \tag{1}$$

subjected to the boundary conditions

$$v(\alpha, t) = f_1(t), \quad 0 \le t \le T$$
(2)

$$v(\beta, t) = f_2(t), \quad 0 \le t \le T$$
(3)

and the initial condition

$$v(x,0) = g(x), \quad \alpha \le x \le \beta,$$
 (4)

where the coefficient μ is the kinematic viscosity and T is the final time. The functions f_1 , f_2 , and g are prescribed conditions depending on each specific problem. The parameters α and β are the endpoints of the domain.

III. NUMERICAL METHOD

The numerical methods used to find the solution of the Burger's equation are provided in this section. We first employ the second-order finite difference method to the spatial derivatives as shown in subsection A.

A. The Second-Order Finite Difference Method

Consider the Burger's equation

$$\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} = \mu \frac{\partial^2 v}{\partial x^2}, \quad \alpha < x < \beta.$$
(5)

We apply the central finite difference method to the derivative terms $\frac{\partial v}{\partial x}$ and $\frac{\partial^2 v}{\partial x^2}$ in Eq. (5). The closed interval $[\alpha, \beta]$ is divided into L subintervals. That is $\alpha = x_0 \leq x_1 \leq ... \leq x_L = \beta$. Then,

$$\frac{\partial v}{\partial x} \approx \frac{V_{m+1}(t) - V_{m-1}(t)}{2h} \tag{6}$$

$$\frac{\partial^2 v}{\partial x^2} \approx \frac{V_{m+1}(t) - 2V_m(t) + V_{m-1}(t)}{h^2},\tag{7}$$

where $V_m(t)$ is the approximation of v at the point x_m , m = 1, 2, ..., L - 1 and $h = x_m - x_{m-1}$. Substituting Eqs. (6) and (7) into Eq. (5), we obtain

$$\begin{bmatrix} \frac{\partial v}{\partial t} \end{bmatrix}_m + V_m(t) \begin{bmatrix} \frac{V_{m+1}(t) - V_{m-1}(t)}{2h} \end{bmatrix}$$
$$= \mu \begin{bmatrix} \frac{V_{m+1}(t) - 2V_m(t) + V_{m-1}(t)}{h^2} \end{bmatrix},$$
(8)

or

$$\begin{bmatrix} \frac{\partial v}{\partial t} \end{bmatrix}_{m} = \frac{\mu}{h^{2}} \left[V_{m+1}(t) - 2V_{m}(t) + V_{m-1}(t) \right] - \frac{V_{m}(t)}{2h} \left[V_{m+1}(t) - V_{m-1}(t) \right],$$
(9)

where $\left[\frac{\partial v}{\partial t}\right]_m$ is $\frac{\partial v}{\partial t}$ at the point (x_m, t) and L is the number of grids.

B. Milne Method

The Milne method is a multi-step method for solving the initial-value problem of the equation $\frac{dy}{dt} = f(t,y)$ defined by

$$y_{n+2} - y_n = k\left(\frac{1}{3}f_{n+2} + \frac{4}{3}f_{n+1} + \frac{1}{3}f_n\right), \qquad (10)$$

where the positive integer n is the n^{th} time step, and $k = t_n - t_{n-1}$. It is an implicit method and is characterized by two polynomials

$$p_1(x) = x^2 - 1$$

$$p_2(x) = \frac{1}{3}x^2 + \frac{4}{3}x + \frac{1}{3}x^2$$

The roots of p_1 are +1 and -1, which are simple roots. Furthermore, $p'_1(x) = 2x$ and $p'_1(1) = 2 = p_2(1)$. Thus, the conditions of consistency and stability are achieved [17]. Applying the multi-step Milne method to the time-dependent expression in the Burger's equation, Eq. (9), we have

$$V_m^{n+2} - V_m^n = \frac{k}{3} \left\{ \frac{\mu}{h^2} \left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2} \right] - \frac{V_m^{n+2}}{2h} \right] \\ \left[V_{m+1}^{n+2} - V_{m-1}^{n+2} \right] + 4 \left(\frac{\mu}{h^2} \left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1} \right] - \frac{V_m^{n+1}}{2h} \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right) + \frac{\mu}{h^2} \\ \left[V_{m+1}^n - 2V_m^n + V_{m-1}^n \right] - \frac{V_m^n}{2h} \left[V_{m+1}^n - V_{m-1}^n \right] \right\}.$$

From Eq. (11), the nonlinear term $V_m^{n+2} \left[V_{m+1}^{n+2} - V_{m-1}^{n+2} \right]$ is calculated by using the linearization method provided in [5] such that $V_m^{n+2}(V_{m+1}^{n+2} - V_{m-1}^{n+2}) \approx U_m^{n+2}(V_{m+1}^{n+2} - V_{m-1}^{n+2})$, where U_m^{n+2} is computed by using linear extrapolation depending on V_m^{n+1} and V_m^n , Therefore

$$V_m^{n+2} \cong U_m^{n+2} = \left(1 + \left(\frac{b_{n+2}}{b_{n+1}}\right)\right) V_m^{n+1} - \left(\frac{b_{n+2}}{b_{n+1}}\right) V_m^n \tag{12}$$

where $b_{n+2} = t_{n+2} - t_{n+1}$ and $b_{n+1} = t_{n+1} - t_n$. Substituting Eq. (12) into Eq. (11), we have

$$V_m^{n+2} - V_m^n = \frac{k}{3} \left\{ \frac{\mu}{h^2} \left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2} \right] - \frac{1}{2h} \\ \left[\left(1 + \left(\frac{b_{n+2}}{b_{n+1}} \right) \right) V_m^{n+1} - \left(\frac{b_{n+2}}{b_{n+1}} \right) V_m^n \right] \\ \left[V_{m+1}^{n+2} - V_{m-1}^{n+2} \right] + 4 \left(\frac{\mu}{h^2} \left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1} \right] - \frac{V_m^{n+1}}{2h} \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right) + \frac{\mu}{h^2} \\ \left[V_{m+1}^n - 2V_m^n + V_{m-1}^n \right] - \frac{V_m^n}{2h} \\ \left[V_{m+1}^n - V_{m-1}^n \right] \right\}.$$
(13)

In this work, we fix the time step for every time period to be Δt . Therefore, $b_{n+2} = b_{n+1} = \Delta t$. So Eq. (13) is rewritten as

$$\begin{split} V_m^{n+2} - V_m^n = & \frac{k}{3} \left\{ \frac{\mu}{h^2} \left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2} \right] - \frac{1}{2h} \\ & \left[2V_m^{n+1} - V_m^n \right] \left[V_{m+1}^{n+2} - V_{m-1}^{n+2} \right] + 4 \\ & \left(\frac{\mu}{h^2} \left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1} \right] \\ & - \frac{V_m^{n+1}}{2h} \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right) + \frac{\mu}{h^2} \left[V_{m+1}^n \\ & - 2V_m^n + V_{m-1}^n \right] - \frac{V_m^n}{2h} \left[V_{m+1}^n - V_{m-1}^n \right] \right\}. \end{split}$$
(14)

The Eq. (14) can be rewritten in the form

$$\gamma_m V_{m-1}^{n+2} + \delta_m V_m^{n+2} + \lambda_m V_{m+1}^{n+2} = g_m, \qquad (15)$$

where

$$\begin{split} \gamma_m &= -2k\mu - 2hkV_m^{n+1} + hkV_m^n \\ \delta_m &= 6h^2 + 4k\mu \\ \lambda_m &= -2k\mu + 2hkV_m^{n+1} - hkV_m^n \\ g_m &= 6h^2V_m^n + 8k\mu \left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1}\right] \\ &- 4hkV_m^{n+1} \left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right] + 2k\mu \left[V_{m+1}^n - 2V_m^n + V_{m-1}^n\right] - hkV_m^n \left[V_{m+1}^n - V_{m-1}^n\right]. \end{split}$$

Since V_m^{n+1} and V_m^n are used to calculate V_m^{n+2} and the initial condition can be employed to V_m^n . To find V_m^{n+1} , we apply the Runge-Kutta and Modified-Newton Raphson methods [3] to the Burger's equation as described in the next section.

C. Modified-Newton Raphson Method

The discretized Burger's equation with the central finite difference and the second-order Runge-Kutta method per-(11) formed in [3] is

Volume 51, Issue 2: June 2021

$$V_{m}^{n+1} - V_{m}^{n} - \frac{k}{2} \left\{ \frac{\mu}{h^{2}} \left[V_{m+1}^{n} - 2V_{m}^{n} + V_{m-1}^{n} \right] - \frac{V_{m}^{n}}{2h} \left[V_{m+1}^{n} - V_{m-1}^{n} \right] \right\} - \frac{k}{2} \left(\frac{\mu}{h^{2}} \left[V_{m+1}^{n+1} - 2V_{m}^{n+1} + V_{m-1}^{n+1} \right] - \left[\frac{V_{m}^{n+1}}{2h} + \frac{k}{2h} \left\{ \frac{\mu}{h^{2}} \left[V_{m+1}^{n+1} - 2V_{m}^{n+1} + V_{m-1}^{n+1} \right] - \frac{V_{m}^{n+1}}{2h} \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right\} \right] \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right\} = 0.$$
(16)

Since Eq. (16) is a nonlinear equation, we apply Modified-Newton Raphson method to find the solution of Eq. (16). Let

$$F_{m}(V_{1}^{n+1}, V_{2}^{n+1}, V_{3}^{n+1}, ..., V_{L-1}^{n+1}) = V_{m}^{n+1} - V_{m}^{n} - \frac{k}{2} \left\{ \frac{\mu}{h^{2}} \left[V_{m+1}^{n} - 2V_{m}^{n} + V_{m-1}^{n} \right] - \frac{V_{m}^{n}}{2h} \left[V_{m+1}^{n} - V_{m-1}^{n} \right] \right\} - \frac{k}{2} \left(\frac{\mu}{h^{2}} \left[V_{m+1}^{n+1} - 2V_{m}^{n+1} + V_{m-1}^{n+1} \right] - \left[\frac{V_{m}^{n+1}}{2h} + \frac{k}{2h} \left\{ \frac{\mu}{h^{2}} \left[V_{m+1}^{n+1} - 2V_{m}^{n+1} + V_{m-1}^{n+1} \right] - \frac{V_{m}^{n+1}}{2h} \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right\} \right] \left[V_{m+1}^{n+1} - V_{m-1}^{n+1} \right] \right\}.$$
(17)

Thus

$$F_m(V_1^{n+1}, V_2^{n+1}, V_3^{n+1}, \dots, V_{L-1}^{n+1}) = 0,$$
(18)

where m = 1, 2, ..., L - 1. Therefore, Eq. (18) can be rewritten in a matrix form as

$$J(V^{(0)})\delta^{(P)} = -F(V^{(P)}), \tag{19}$$

where

$$J(V^{(0)}) = \begin{bmatrix} \frac{\partial F_{1}(V^{(0)})}{\partial V_{1}^{n+1}} & \frac{\partial F_{1}(V^{(0)})}{\partial V_{2}^{n+1}} & \cdots & \frac{\partial F_{1}(V^{(0)})}{\partial V_{L-1}^{n+1}} \\ \frac{\partial F_{2}(V^{(0)})}{\partial V_{1}^{n+1}} & \frac{\partial F_{2}(V^{(0)})}{\partial V_{2}^{n+1}} & \cdots & \frac{\partial F_{2}(V^{(0)})}{\partial V_{L-1}^{n+1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_{L-1}(V^{(0)})}{\partial V_{1}^{n+1}} \frac{\partial F_{L-1}(V^{(0)})}{\partial V_{2}^{n+1}} & \cdots & \frac{\partial F_{L-1}(V^{(0)})}{\partial V_{L-1}^{n+1}} \end{bmatrix}$$
$$\delta^{(P)} = \begin{bmatrix} \Delta V_{1}^{n+1} \\ \Delta V_{2}^{n+1} \\ \vdots \\ \Delta V_{L-1}^{n+1} \end{bmatrix}^{(P)} \text{ and } F(V^{(P)}) = \begin{bmatrix} F_{1} \\ F_{2} \\ \vdots \\ F_{L-1} \end{bmatrix}^{(P)},$$

where P is the number of iterations, $V^{(P)} = (V_1^{n+1}, V_2^{n+1}, ..., V_{L-1}^{n+1})^{(P)}$, $(\Delta V_m^{n+1})^{(P)} = (V_m^{n+1})^{(P+1)} - (V_m^{n+1})^{(P)}$ and $V^{(0)}$ is the initial guess. Therefore $V^{(P+1)} = V^{(P)} + \delta^{(P)}$. The iteration is stopped when $\|\delta^{(P)}\|_{\infty} \leq tol$, where tol is a small positive number.

IV. EXACT SOLUTION

To verify our numerical solutions, in this section, we provide the exact solution of the Burger's equation and its conditions. The exact solution of the one-dimensional Burger's equation is given by Wood [18],

$$v(x,t) = \frac{2\mu\pi e^{-\mu\pi^2 t}\sin(\pi x)}{a + e^{-\mu\pi^2 t}\cos(\pi x)}, \quad 0 < x < 1$$
(20)

with the boundary conditions

$$\begin{cases} v(0,t) = 0 \\ v(1,t) = 0 \end{cases}$$
 $t > 0$ (21)

and initial condition

$$v(x,0) = \frac{2\mu\pi\sin(\pi x)}{a + \cos(\pi x)}, \quad a > 1.$$
 (22)

The Eqs. (20) - (22) will be used in Section 5 to demonstrate the accuracy of the numerical solutions.

V. THE NUMERICAL RESULTS

In this section, we provide the numerical results obtained from Eq. (15). To verify the solutions, we first compare the results with the exact solution. Fig. 1 illustrates the exact and numerical solutions at different number of grid points for T = 1 with $a = 1.1, \Delta t = 0.01$, and $\mu = 0.001$. In the process of finding the numerical solutions V_m^{n+2} in Eq. (15), the number of iterations to obtain V_m^{n+1} from V_m^n by using the Modified-Newton Raphson and the Runge-Kutta methods with $tol = 10^{-5}$ when L = 10, 20, 40 and 80 are 9, 19, 39 and 79, respectively. The figure shows that when the number of grids increases, the numerical results converge to the exact solution. The numerical solutions at T = 1 are shown in Table I and the L_2 and L_{∞} norms errors of the numerical solutions are illustrated in Table II, where the L_2 and L_{∞} norms are

$$L_{2} = ||v - V||_{2} = \sqrt{\sum_{j=0}^{L} |v_{j} - V_{j}|^{2}},$$
$$L_{\infty} = ||v - V||_{\infty} = \max_{j} |v_{j} - V_{j}|,$$

where v and V represent the values of the exact and numerical solutions, respectively. Notice that the errors decrease when the number of grids increases. Plots of the numerical solution depending on x and t are provided in Figs. 2 and 3, where the later is the contour plot with $a = 1.1, \mu =$ 0.001, h = 0.0125 and $\Delta t = 0.01$. In Figs. 2 and 3, the solution has the maximum value 0.013 at the point x = 0.85and the solutions are zero at the boundaries, which is consistent to the boundary conditions Eq. (21). In Fig. 3, the second left vertical line represents the numerical solution value of 0.002, while the next vertical lines represent incremental numerical solution values of 0.004, 0.006, 0.008, 0.01, and 0.012, respectively. Similarly, the second vertical line from the right is the numerical solution value of 0.002, while the next vertical lines on the left of the second vertical line from the right of Fig. 3 represent incremental numerical solution values of 0.004, 0.006, 0.008, 0.01, and 0.012, respectively. Fig. 4 shows the comparisons of the numerical solutions and the exact solutions for different values of μ which are 0.001, 0.0005 and 0.0001 at T = 1 with $a = 1.1, \Delta t = 0.01$ and h = 0.0125. The decreasing coefficients μ decrease the values of the numerical results. Fig. 5 shows the numerical solutions when the constant a increases with the exact solutions at T = 1, $\mu = 0.001$, $\Delta t = 0.01$ and h = 0.0125. The increasing variables a decrease the values of the numerical results. Note that the graphs of the numerical solutions almost overlap with the exact solutions in both Figs. 4 and 5. The L_2 - norm errors for the different values of μ and a are provided in Table III. It is shown that the L_2 - norm errors decrease when a increase and/or μ decreases for both $\Delta t = 0.01$ and $\Delta t = 0.001$. Moreover, the errors decrease with decreasing Δt . Table IV shows the L_2 - norm errors of our numerical solutions compared with the Euler forward discretization [19] and Mac Cormack discretization [20] at different value of μ for $\Delta t = 0.01$ and 0.001, where the L_2 - norm used in [19] and [20] is

$$L_2 = ||v - V||_2 = \sqrt{\frac{\sum_{j=0}^{L} |v_j - V_j|^2}{N}},$$

which is employed in Table IV, where N is the number of time steps. The numbers in the table show that the errors of our numerical solutions are less than that in [19] and [20].

TABLE I Comparison of the numerical solutions with the exact solution at different values of L when $a=1.1, T=1, \mu=0.001$ and $\Delta t=0.01.$

x		Numerical solution $L = 10$ $L = 20$ $L = 40$ $L = 80$ H161E-04 9.4163E-04 9.4163E-04 9.4163E-04 0		Exact	
	L = 10	L = 20	L = 40	L = 80	solution
0.1	9.4161E-04	9.4163E-04	9.4163E-04	9.4163E-04	0.000942
0.2	1.9235E-03	1.9236E-03	1.9236E-03	1.9236E-03	0.001924
0.3	2.9923E-03	2.9924E-03	2.9924E-03	2.9924E-03	0.002992
0.4	4.2080E-03	4.2083E-03	4.2084E-03	4.2084E-03	0.004208
0.5	5.6548E-03	5.6556E-03	5.6558E-03	5.6559E-03	0.005656
0.6	7.4494E-03	7.4513E-03	7.4518E-03	7.4519E-03	0.007452
0.7	9.7140E-03	9.7160E-03	9.7167E-03	9.7169E-03	0.009717
0.8	1.2271E-02	1.2241E-02	1.2235E-02	1.2234E-02	0.012233
0.9	1.2525E-02	1.2248E-02	1.2172E-02	1.2153E-02	0.012146



Fig. 1. Exact and numerical solutions at different values of L when a=1.1, T=1, and $\mu=0.001$.

TABLE II Values of the L_2 -norm and L_∞ -norm errors of the solutions in table I for different values of L at T=1.

	L = 10	L = 20	L = 40	L = 80
L_2 -norm	3.8091E-04	1.8308E-04	6.6726E-05	2.3750E-05
L_{∞} -norm	3.7899E-04	1.4753E-04	3.8173E-05	9.6834E-06



Fig. 2. The surface plot of the numerical solution with $a = 1.1, \mu = 0.001, \Delta t = 0.01$ and h = 0.0125.



Fig. 3. The contour plot of Fig. 2 when $a = 1.1, \mu = 0.001, \Delta t = 0.01$ and h = 0.0125.



Fig. 4. Numerical solutions for different values of μ with $a=1.1,T=1,\Delta t=0.01$ and h=0.0125.

VI. CONCLUSION

In this research, we propose a new method to find the numerical solutions of the Burger's equation which is the combination of the multi-step Milne method and the central finite difference method. Since we use V_m^n and V_m^{n+1} to determine the solution V_m^{n+2} , and V_m^n can be obtained from the initial condition, the second-order Runge-Kutta method



Fig. 5. Numerical solutions for different values of a with $T = 1, \mu = 0.001, \Delta t = 0.01$ and h = 0.0125.

TABLE III VALUES OF L_2 -norm errors for different values of a and μ with $\Delta t=0.01$ and $\Delta t=0.001.$

Δt		a = 1.1	a=2	a = 4
	$\mu=0.001$	2.37502E-05	2.24741E-07	3.02647E-08
0.01	$\mu=0.0005$	6.86689E-06	5.76347E-08	7.67045E-09
	$\mu=0.0001$	3.12268E-07	2.35362E-09	3.10223E-10
	$\mu = 0.001$	2.37478E-05	2.24737E-07	3.02644E-08
0.001	$\mu=0.0005$	6.86671E-06	5.76344E-08	7.67043E-09
	$\mu=0.0001$	3.12268E-07	2.35362E-09	3.10223E-10

TABLE IVCOMPARISON OF THE L_2 -NORM ERRORS OF OUR NUMERICALSOLUTIONS WITH [19] AND [20] FOR a = 1.1, h = 0.0125, T = 1, and $\mu = 0.001$ and 0.0001.

μ	Δt	[19]	[20]	Numerical
				solution
0.001	0.01	4.9467E-06	1.0399E-05	2.3750E-06
	0.001	2.6372E-06	1.9502E-06	7.5097E-07
0.0001	0.01	2.6372E-07	1.9502E-07	3.1227E-08
	0.001	3.4891E-08	2.6251E-08	9.8748E-09

and Modified-Newton Raphson scheme are used to calculate V_m^{n+1} . The numerical solutions obtained are compared with the exact solutions to verify the results. The comparisons are illustrated in Fig. 1 and Table I at T = 1. They show that the numerical results converge to the exact solutions when the number of grids increases. The L_2 and L_{∞} norms errors are shown in Table II to determine the accuracy of our numerical solutions. The surface and contour plots of the numerical solution depending on both independent variables x and t is illustrated in Figs. 2 and 3, where the highest value of the numerical solution occurs at x = 0.85, approximately, and it smoothly decreases to zero to both boundaries of the domain for each time t. The numerical results and exact solutions reduce in height for a small value of μ but vice versa with the constant a as shown in Figs. 4 and 5, respectively. The L_2 -norm errors of the numerical results in Figs. 4 and 5 are illustrated in Table III. For the different values of the constant a in the initial condition and the coefficient μ in the governing equation, the numerical and exact solutions

are in excellent agreement. As compared to other numerical methods shown in Table IV, our numerical solutions are more accurate.

ACKNOWLEDGMENT

The author is highly thankful to Asst.Prof.Kanognudge Wuttanachamsri, my advisor, to help and support me for the publication.

REFERENCES

- H. Bateman, "Some recent researches on the motion of fluids," *Monthly Weather Review*, vol. 43, no. 4, pp. 163-170, 1915.
- [2] B. Mayur, A. Ashish, C. Lakshmi, M. Vijitha, and V.S. Aswin, "A systematic literature review of Burgers equation with recent advances," *Pramana*, vol. 90, no. 6, pp. 1-21, 2018.
- [3] A. Chaiyasit, T. Klinthub, K. Arleemeen, W. Sanprasert and K. Wuttanachamsri, "A Second-Order Accurate Numerical Approach for Burger's Equation," *The 11th NPRU National Academic Conference Nakhon Pathom Rajabhat University*, Nakhon Pathom, Thailand, no. 1, pp. 82-89, 2019.
- [4] Z.Y. Ali, "Application New Iterative Method For Solving Nonlinear Burger's Equation And Coupled Burger's Equations," *International Journal of Computer Science Issues (IJCSI)*, vol. 15, no. 3, pp. 31-35, 2018.
- [5] N.A. Mohamed, "Fully implicit scheme for solving Burgers' equation based on finite difference method," *Egyptian Journal for Engineering Sciences and Technology*, vol. 26, pp. 38-44, 2018.
- [6] N.A. Mohamed, "Solving one-and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes" *Arab Journal* of Basic and Applied Sciences, vol. 26, no. 1, pp. 254-268, 2019.
- [7] P.G. Zang and J.P. Wang, "A predictor–corrector compact finite difference scheme for Burgers equation," *Applied Mathematics and Computation*, vol. 219, no. 3, pp. 892-898, 2012.
- [8] S.M. Zulkifli, N.I. Zainuddin, S.N. Syamimi, and J. Rosly, "Mathematical Modelling Of Burger's Equation Applied In Traffic Flow," 2017.
- [9] K. Ali,C. Yucel, and T. Orkun, "Exact solution for the conformable burgers equation by the hopf-cole transform," *Çankaya Üniversitesi Bilim ve Mühendislik Dergisi*, vol. 13, no. 2, 2016.
- [10] Y. Ucar, N.M. Yagmurlu, and O. Tasbozan, "Numerical solutions of the modified Burgers equation by finite difference methods," *Journal* of Applied Mathematics, Statistics and Informatics, vol. 13, no. 1, pp. 19-30, 2017.
- [11] G. Çelikten and E. AKSAN, "Explicit exponential finite difference methods for the numerical solution of modified Burgers equation," *Eastern Anatolian Journal of Science*, vol. 3, no. 1, pp. 45-50, 2017.
- [12] S. Sungnul, B. Jitsom and M. Punpocha, "Numerical solution of the modified Burgers equation using FTCS implicit scheme," *IAENG International Journal of Applied Mathematics*, vol. 48, no. 1, pp. 53-61, 2018.
- [13] M.A. Sheikh, L. Andallah, A. Kowser, "A Comparative Study of Finite Difference Scheme for Burger's Equation," *Gazi University Journal of Science*, vol. 27, no. 4, pp. 1045-1051, 2014.
- [14] D. Deng and J. Xie, "High-order exponential time differencing methods for solving one-dimensional Burgers equation," *IAENG International Journal of Computer Science*, vol. 43, no. 2, pp. 167-175, 2016.
- [15] D. Deng and T. Pan, "A fourth-order singly diagonally implicit Runge-Kutta method for solving one-dimensional Burgers equation," *IAENG International Journal of Applied Mathematics*, vol. 45, no. 4, pp. 327-333, 2015.
- [16] P.W. Li, "Space-time generalized finite difference nonlinear model for solving unsteady Burgers equations," *Applied Mathematics Letters*, vol. 114, pp. 106896, 2021.
- [17] D. R. Kincaid and E. W. Cheney, "Mathematics of Scientific Computing," Published by American Mathematical Society, 1991.
- [18] W.L. Wood, "An exact solution for Burger's equation," *Communications in Numerical Methods in Engineering*, vol. 22, no. 7, pp. 797-798, 2006.
- [19] T. Bui, "Explicit and Implicit Methods In Solving Differential Equations," *Honors Scholar Theses*, 119, 2010.
- [20] A. Perrin, H. H. Hu, "An explicit finite-difference scheme for simulation of moving particles," *Journal of Computational Physics*, vol. 212, no. 1, pp. 166-187, 2006.