A NUMERICAL SOLUTION OF BURGER'S EQUATION BASED ON MILNE METHOD

SUPRANEE CHONLADED

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN APPLIED MATHEMATICS

DEPARTMENT OF MATHEMATICS SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2021
KMITL-2021-SC-M-001-010

Thesis Title	A Numerical Solution of Burger's Equation Based on Milne Method
Student Name	Miss Supranee Chonladed
Student ID	62605015
Degree	Master Degree of Science in Applied Mathematics
Department	Mathematics
Year	2021
Thesis Advisor	Asst. Prof. Dr. 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 method. A linearization method with a weighted technique is employed to handle the nonlinear term. Due to the multi-step approach, the second-order Runge-Kutta and Modified-Newton Raphson schemes are applied to determine another initial condition. The numerical results are compared with the exact solution, where the error norms are used to evaluate the accuracy of the technique. They are in an excellent agreement. Numerical visualizations are fulfilled for different values of constant parameters.

Keywords : Burger's equation, Milne method, Finite difference method, Runge-Kutta method, Modified-Newton Raphson method

Acknowledgements

I would like to express my sincere appreciation to all of you for your advice and assistance in completing this research. My professor, Assistant Professor Dr.Kanognudge Wuttanachamsri, please be a consult, validates the accuracy of this thesis and supports continuous operations.

Besides my advisor, I would like to express my sincere gratitude to chairperson and examiner, Assistant Professor Dr.Jiraphan Suntornchost and Dr.Jiraphat Yokrattanasak for their helpful comments and for giving me various knowledge to make this thesis more accurate and complete. Lastly, I would like to thank you my family, my friends, and the Ph.D. students for their continued support and encouragement over the years of my studies.

Supranee Chonladed

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we discuss the research motivation, objectives, scopes, benefits, and research methodology of the study.

1.1 Research Motivation

Burger's equation is a fundamental partial differential equation. It was firstly given by Harry Bateman in 1915 [3], and in 1948, Johannes Martinus Burgers [5], a Dutch physicist, concluded it in the mathematical modeling in the theory of turbulence. Burger's equation becomes one of the leading figures in the field of fluid mechanics which is suitable for the analysis of various important areas [9]. It is used in several branches of applied mathematics, physics and engineering such as modeling of gas dynamics, heat conduction, traffic flow. Burger's equation is solved by numerous numerical methods such as the finite difference and the finite element methods. In this research, the Milne method combining with the finite difference, Runge-Kutta and Modified-Newton Raphson methods are used to find the solution of the Burger's equation.

1.2 Objectives of the Study

1) To study the Burger's equation.
2) To find the numerical solution of the Burger's equation using the Milne method and other finite difference approaches.
3) To learn how to write computer program to find the numerical solutions.

1.3 Scopes of the Study

1) To study the one-dimensional Burger's equation with boundary conditions and an initial condition.
2) To apply the multi-step Milne method to find the numerical solution of the Burger's equation.

1.4 Benefits of the Study

1) To obtain a proper solution of the nonlinear Burger's equation.
2) To become a professional in coding by using MATLAB.

1.5 Research Methodology

1) Study the fundamentals of the Burger's equation.
2) Study the multi-step Milne method.
3) Study tools to solve the nonlinear term of the Burger's equation.
4) Study the process to find the numerical solutions and solve the Burger's equation by using the Milne method.
5) Write a computer program to find the numerical solutions.
6) Verify the solution and conclusion.
7) Write the thesis book.

Table 1.1: Time frame of research

Activity	Time frame (month of year)								
	2019	2020						2021	
	11 - 12	1 - 2	3 - 4	5 - 6	7 - 8	9 - 10	11 - 12	1 - 3	4 - 6
Step 1	↔	↔							
Step 2	↔	↔							
Step 3		↔	↔						
Step 4			↔	↔					
Step 5				↔	↔	↔			
Step 6							↔	↔	
Step 7								↔	↔

Chapter 2

Basic knowledge and Literature Reviews

In this chapter, we present the basic knowledge applied in our research and the literature reviews.

2.1 Differential Equations: DEs

Differential equations are equations with the derivatives of the dependent variables relative to the independent variables. We consider $y = f(x,t)$ a function of x and t which are independent variables, so y is called as the dependent variable.

2.1.1 Ordinary Differential Equations: ODEs

Ordinary differential equation is defined as an equation composing of the derivative of the dependent variable having only one independent variable. Some examples of ODEs are shown in Table 2.1

Table 2.1: Examples of the ODEs.

Equation	independent variable	dependent variable
$\frac{dy}{dx} = 3x$	x	y
$\frac{d^2y}{dx^2} + 2xy = e^x$	x	y
$\frac{d^2x}{dt^2} + 3\frac{dx}{dt} + 2x^4 = \sin(2t)$	t	x

2.1.2 Partial Differential Equations: PDEs

Partial differential equation is (PDE) an equation consisting of the partial derivatives of a dependent variable having more than one independent variable, such as

$$(i) \qquad \frac{\partial^2 u}{\partial t^2} - c^2\frac{\partial^2 u}{\partial x^2} = 2 + 3x$$

It is a PDE with the dependent variable u and two independent variables t and x.

$$(ii) \qquad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$$

It is a PDE with the dependent variables u and v, and two independent variables x and y.

Linear Partial Differential Equation is a partial differential equation in which the dependent variables or their derivatives are raised to the first order derivative. No term can be a product of the dependent variables or its derivative. In addition, no term contains a transcendental function of the dependent variables or their derivatives in the equation.

Non-Linear Partial Differential Equation refers to the partial differential equation which is not a linear differential equation.

Table 2.2: Examples of second-order partial differential equation.

Equation	Equation model	Example of usage
Laplace's equation	$\nabla^2 u = 0$	potential field
Poisson's equation	$\nabla^2 u = f(x, y, z)$	potential field (Source)
Diffusion equation	$\nabla^2 u = k\frac{\partial u}{\partial t}$	Heat conduction or time-dependent diffusion
Wave equation	$\nabla^2 u = k\frac{\partial^2 u}{\partial t^2}$	Wave propagation

The equations in Table 2.2 can be divided into three main types: Elliptic equation, Parabolic equation, and Hyperbolic equation. Consider the general equation

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial t} + c\frac{\partial^2 u}{\partial t^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial t} + fu + g = 0, \tag{2.1}$$

where $u = u(x, t)$ and terms a, b, c, d, e, f and g are coefficients, which may be constants. The different types of equations are considered as follows.

(i) $b^2 - 4ac > 0$ (two real distinct roots)

The PDE (2.1) is said to be "**Hyperbolic**"

(ii) $b^2 - 4ac = 0$ (one real repeated roots)

The PDE (2.1) is said to be "**Parabolic**"

(iii) $b^2 - 4ac < 0$ (conjugate complex roots)

The PDE (2.1) is said to be "**Elliptic**"

Example 2.1.

Consider $\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^2 u}{\partial x \partial t} + (1 + u^2)\frac{\partial^2 u}{\partial t^2} = 0.$ (2.2)

Then, from (2.1), $a = 1, b = u, c = 1+u^2, d = e = f = g = 0.$

So $b^2 - 4ac = u^2 - 4(1)(1+u^2)$

$= u^2 - 4 - 4u^2$

$= -4 - 3u^2.$

Because $u^2 \geq 0, b^2 - 4ac < 0.$

Then (2.2) is an elliptic PDE.

Initial and Boundary Conditions

To find the solution of partial differential equations, we need to know the boundary conditions (BC) which the resultant problem is called boundary value problems and the initial condition (IC) with which the problem is called initial value problems.

- Initial Value Problems (IVPs)

$$\frac{dy}{dx} = f(x,y) \qquad \text{for all } x \in [a,b]$$

with the initial condition $y(a) = y_a$, where y_a is a known constant.

- Boundary Value Problems (BVPs)

$$\frac{dy}{dx} = f(x,y) \qquad \text{for all } x \in [a,b]$$

with the boundary conditions $y(a) = y_a$ and $y(b) = y_b$, where y_a and y_b are known constants.

2.2 Burger's Equation

We consider the one-dimensional time dependent Burger's equation, which is the nonlinear Burger's equation defined by

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \mu\frac{\partial^2 v}{\partial x^2}, \quad \alpha < x < \beta, \quad 0 \le t \le T \tag{2.3}$$

subject to the boundary conditions

$$v(\alpha,t) = f_1(t), \quad 0 \le t \le T \tag{2.4}$$

$$v(\beta,t) = f_2(t), \quad 0 \le t \le T \tag{2.5}$$

and the initial condition

$$v(x,0) = g(x), \quad \alpha \le x \le \beta, \tag{2.6}$$

where μ is the coefficient of 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 type of Burger's equation can be determined by the formula

$$(b^2 - 4ac) = 0 - 4(\mu)(0) = 0.$$

Thus Burger's equation is a parabolic partial differential equation.

2.3 Finite Difference Method: FDM

The concept of the finite difference method is the approximation of the derivative in the partial differential equations. We assume the derivative is approximated by using the Taylor series.

Theorem 2.2. (Taylor series expansion) If a function f is differentiable through order $n+1$ in an interval I and $x_0, x \in I$, then $f(x)$ can be written as

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \frac{f'''(x_0)}{3!}(x-x_0)^3 + ... + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + E_n, \quad (2.7)$$

where $E_n = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$, and $\xi \in [x_0, x]$ is a remainder term.
Next, we present the definition of Big O, which is used to provide the accuracy of the finite difference method.

Definition 2.3. (Big O) Let $f(x)$ and $g(x)$ be two functions. Then $f(x) = O(g(x))$ as $x \to a$ if and only if

$$\lim_{x \to a} \left| \frac{f(x)}{g(x)} \right| = c \in [0, \infty)$$

From Theorem 2.2 and Definition 2.3, the function f at the point $x_0 + h$, around the point x_0 is

$$f(x_0+h) = f(x_0) + \frac{f'(x_0)}{1!}h + \frac{f''(x_0)}{2!}h^2 + \frac{f'''(x_0)}{3!}h^3 + ... + \frac{f^{(n)}(x_0)}{n!}h^n + O(h^{n+1}), \quad (2.8)$$

where $h = x - x_0$ and $O(h^{n+1})$ denotes the truncation error of order h^{n+1}. Let $u = u(x,t)$ is a function of 2-independent variables. The Taylor series around the point x are given by

$$u(x+h,t) = u(x,t) + h\frac{\partial u}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2} + \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots \quad (2.9)$$

$$u(x-h,t) = u(x,t) - h\frac{\partial u}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2} - \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots \quad (2.10)$$

The first order derivative

From (2.9), we have

$$\frac{\partial u}{\partial x} = \frac{u(x+h,t) - u(x,t)}{h} - \frac{h}{2!}\frac{\partial^2 u}{\partial x^2} - \frac{h^2}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots \quad (2.11)$$

Then

$$E_1 = -\frac{h}{2}\frac{\partial^2 u}{\partial x^2} - \frac{h^2}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots$$

From (2.11), we have

$$\lim_{h\to 0}\left|\frac{-\frac{h}{2}\frac{\partial^2 u}{\partial x^2} - \frac{h^2}{3!}\frac{\partial^3 u}{\partial x^3} + \ldots}{h}\right| = \frac{1}{2}\frac{\partial^2 u}{\partial x^2} = c \in [0,\infty),$$

since we assume that u is differentiable up to order n, $n \in \mathbb{N}$.

Thus, $-\frac{h}{2}\frac{\partial^2 u}{\partial x^2} - \frac{h^2}{3!}\frac{\partial^3 u}{\partial x^3} + \ldots = O(h)$. Then the value of $\frac{\partial u}{\partial x}$ can be approximated in the form

$$\frac{\partial u}{\partial x} = \frac{u(x+h,t) - u(x,t)}{h} + O(h) \qquad (2.12)$$

which is called **Forward Difference Method**, where $O(h)$ denotes the truncation error of order h. Similarly, we approximate the value of $\frac{\partial u}{\partial x}$ by **Backward Difference Method** in the form

$$\frac{\partial u}{\partial x} = \frac{u(x,t) - u(x-h,t)}{h} + O(h). \qquad (2.13)$$

Subtracting (2.9) by (2.10), we have

$$u(x+h,t) - u(x-h,t) = 2h\frac{\partial u}{\partial x} + \frac{2h^3}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots$$

Then we approximate the value of $\frac{\partial u}{\partial x}$ by **Central Difference Method** in the form

$$\frac{\partial u}{\partial x} = \frac{u(x+h,t) - u(x-h,t)}{2h} + O(h^2), \qquad (2.14)$$

where $O(h^2)$ denotes the truncation error of order h^2.

The second order derivative

The summation of (2.9) and (2.10) is

$$u(x+h,t) + u(x-h,t) = 2u(x,t) + \frac{2h^2}{2!}\frac{\partial^2 u}{\partial x^2} + \frac{2h^4}{4!}\frac{\partial^4 u}{\partial x^4} + \cdots$$

Then we approximate the value of $\frac{\partial^2 u}{\partial x^2}$ by **Central Difference Method** in the form

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} + O(h^2). \qquad (2.15)$$

Similarly, the formulae for approximating the derivative of the unknown u with respect to t are

- Forward Finite Difference Method

$$\frac{\partial u}{\partial t} = \frac{u(x,t+k) - u(x,t)}{k} + O(k) \tag{2.16}$$

- Backward Finite Difference Method

$$\frac{\partial u}{\partial t} = \frac{u(x,t) - u(x,t-k)}{k} + O(k) \tag{2.17}$$

- Central Finite Difference Method

$$\frac{\partial u}{\partial t} = \frac{u(x,t+k) - u(x,t-k)}{2k} + O(k^2) \tag{2.18}$$

- Central Finite Difference Method

$$\frac{\partial u}{\partial t} = \frac{u(x,t+k) - 2u(x,t) + u(x,t-k)}{k^2} + O(k^2), \tag{2.19}$$

where k is Δt.

Before we present the Milne-Simpson method, we generate the mesh as follows.

Grid spacing

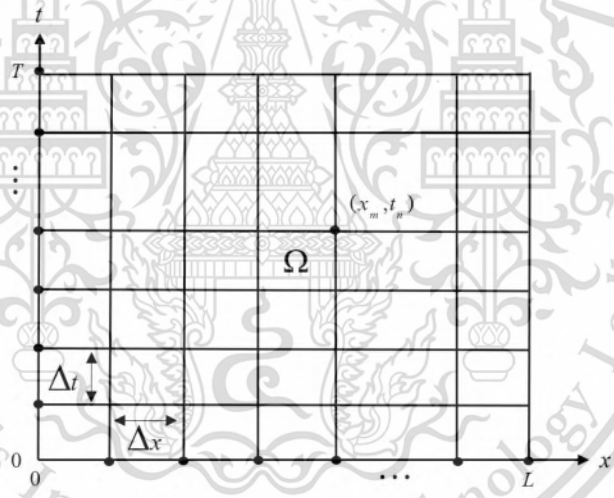


Figure 2.1: Two-dimensional grid points for the finite difference method.

From Figure 2.1, the $x-$axis is the space x and $y-$axis is the time t. The interval $[0, L]$ on $x-$axis is divided into M subintervals, with equal length $h = \Delta x = \frac{L-0}{M}$. That is $0 = x_0 < x_1 < x_2 < ... < x_{M-1} < x_M = L$. Then $x_m = mh = m * \Delta x$, where $m = 0, 1, 2, ..., M$. Similarly, the interval $[0, T]$ on $y-$axis is divided into N subintervals, with equal length $k = \Delta t = \frac{T-0}{N}$. We get $t_n = nk = n * \Delta t$, where $n = 0, 1, 2, ..., N$. The solution at a grid point (x_m, t_n) is denoted by

$$u(x_m, t_n) = u(m * \Delta x, n * \Delta t)$$

and in this study, we define U_m^n to be the approximation of $u(x_m, t_n)$.

2.4 Milne-Simpson method

In this section, we introduce the multi-step Milne-Simpson method to solve the ODE $y' = f(x, y)$. The multi-step method is described by the equation

$$a_w y_n + a_{w-1} y_{n-1} + a_{w-2} y_{n-2} + ... + a_0 y_{n-w} = h \left[b_w f_n + b_{w-1} f_{n-1} + b_{w-2} f_{n-2} + ... + b_0 f_{n-w} \right], \tag{2.20}$$

where $a_0, ..., a_w$ and $b_0, ..., b_w$ are constants. A multi-step method with $b_w \neq 0$ is an **implicit** scheme. If $b_w = 0$, the method is an **explicit** approach. From (2.20), we have two polynomials

$$\begin{aligned} p_1(x) &= a_w x^w + a_{w-1} x^{w-1} + ... + a_0 \\ p_2(x) &= b_w x^w + b_{w-1} x^{w-1} + ... + b_0, \end{aligned} \tag{2.21}$$

which are used to verify the stability and consistency of the method. Next, we derive the Milne method which begins with Simpson's rule. We integrate the differential equation $y' = f(x, y)$ in the interval $[x_n, x_{n+2}]$. That is

$$\int_{x_n}^{x_{n+2}} dy = \int_{x_n}^{x_{n+2}} f(x, y) dx$$

or

$$y(x_{n+2}) = y(x_n) + \int_{x_n}^{x_{n+2}} f(x, y) dx. \tag{2.22}$$

Since the Simpson's rule is

$$\int_a^b f(x, y) dx = \frac{b-a}{6} \left[f(b) + 4f(\frac{a+b}{2}) + f(a) \right] - \frac{f^{(4)}(\xi)}{2880} (b-a)^5, \tag{2.23}$$

where ξ is a number between a and b. Let $h = \dfrac{b-a}{2}, a = x_n$, and $b = x_{n+2}$. We have

$$\int_{x_n}^{x_{n+2}} f(x, y) dx = \frac{h}{3} \left[f(x_{n+2}) + 4f(x_{n+1}) + f(x_n) \right] - \frac{h^5}{90} f^{(4)}(\xi), \tag{2.24}$$

where $-\dfrac{h^5}{90} f^{(4)}(\xi) = O(h^5)$, the truncation error of order 5. Substituting (2.24) into (2.22), we have

$$y_{n+2} - y_n = h \left(\frac{1}{3} f_{n+2} + \frac{4}{3} f_{n+1} + \frac{1}{3} f_n \right), \tag{2.25}$$

for all $n = 1, 2, 3, ...$. The equation (2.25) is called **Milne** or **Milne-Simpson method**.

Stability and Consistency

The Milne method is **stable** if all roots of p_1 lie in the disk $|x| \leq 1$. The method is **consistent** if $p_1(1) = 0$ and $p_1'(1) = p_2(1)$ [8].

From (2.21), the Milne Method is characterized by two polynomials

$$p_1(x) = x^2 - 1$$
$$p_2(x) = \frac{1}{3}x^2 + \frac{4}{3}x + \frac{1}{3}.$$

The roots of p_1 are 1 and -1. Furthermore, $p_1(1) = 0, p_1'(x) = 2x$, $p_1'(1) = 2$ and $p_2(1) = 2$. Thus, the conditions of consistency and stability are fulfilled.

2.5 Literature Reviews

Chaiyasit A. et al. (2019) provided the numerical solution of the one-dimensional Burger's equation by using the second-order Runge-Kutta method, the central finite difference method and Newton-Raphson method. Ali Z.Y. (2018) used a new iterative method to find the solution of the Burger's equation and claimed that the method was accurate. Mohamed N.A. (2018), provided a new numerical scheme based on the finite difference method for solving the nonlinear one-dimensional Burgers' equation and Mohamed N.A. (2019) introduced new fully implicit schemes for solving the one-dimensional and two-dimensional unsteady Burger's equation. Zang P.G. and Wang J.P. (2012) proposed a compact predictor-corrector finite difference scheme to solve the Burger's equation. Ali K. et al. (2016) found the solution of Burger's equation by using Cole-Hopf transformation. Ucar Y. et al. (2017) and Çelikten G. et al. (2017) obtained a numerical solution of the modified Burger's equation by using the finite difference methods and explicit exponential finite difference schemes based on four different linearization techniques, respectively. Sheikh M.A. et al. (2014) compared the numerical solutions of Burger's equation by using Lax-Friedrich and Lax-Wendroff schemes.

Most of the above literatures used single-step finite difference methods and cole-Hopf transformation to find the numerical solution of Burger's equation. In this work, we study the one-dimensional Burger's equation by using the central finite difference scheme with the spatial term and the multi-step Milne method for time-dependent expression. Then we change the nonlinear term into a linear form by using a linearization method with a weighted technique. The numerical results are given in chapter 4.

Chapter 3

Research Methodology

In this chapter, we introduce methods to find the solution of the Burger's equation. We first employ the second-order finite difference method applied to the spatial derivatives.

3.1 Second-order Finite Difference Method

Consider Burger's equation

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \mu\frac{\partial^2 v}{\partial x^2}, \quad \alpha < x < \beta. \tag{3.1}$$

We use the central finite difference method with the derivative terms $\frac{\partial v}{\partial x}$ and $\frac{\partial^2 v}{\partial x^2}$ in (3.1). Therefore the closed interval $[\alpha, \beta]$ is divided into L subintervals. That is $\alpha = x_0 < x_1 < ... < x_L = \beta$. Then,

$$\frac{\partial v}{\partial x} \approx \frac{V_{m+1}(t) - V_{m-1}(t)}{2h} \tag{3.2}$$

$$\frac{\partial^2 v}{\partial x^2} \approx \frac{V_{m+1}(t) - 2V_m(t) + V_{m-1}(t)}{h^2}, \tag{3.3}$$

where $V_m(t)$ is the approximation of v at the point x_m, $m = 1, 2, ..., L-1$. Substituting (3.2) and (3.3) into (3.1), we obtain

$$\left[\frac{\partial v}{\partial t}\right]_m + V_m(t)\left[\frac{V_{m+1}(t) - V_{m-1}(t)}{2h}\right] = \mu\left[\frac{V_{m+1}(t) - 2V_m(t) + V_{m-1}(t)}{h^2}\right], \tag{3.4}$$

or

$$\left[\frac{\partial v}{\partial t}\right]_m = \frac{\mu}{h^2}\left[V_{m+1}(t) - 2V_m(t) + V_{m-1}(t)\right] - \frac{V_m(t)}{2h}\left[V_{m+1}(t) - V_{m-1}(t)\right], \tag{3.5}$$

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 and h is the step size Δx.

3.2 Milne Method

Applying the Milne method for the time-dependent expression (2.25) to the Burger's equation, (3.1), we have

$$V_m^{n+2} - V_m^n = \frac{k}{3}\left\{\frac{\mu}{h^2}\left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2}\right] - \frac{V_m^{n+2}}{2h}\left[V_{m+1}^{n+2} - V_{m-1}^{n+2}\right] + 4\left(\frac{\mu}{h^2}\left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1}\right] - \frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right]\right) + \frac{\mu}{h^2}\left[V_{m+1}^n - 2V_m^n + V_{m-1}^n\right] - \frac{V_m^n}{2h}\left[V_{m+1}^n - V_{m-1}^n\right]\right\}, \quad (3.6)$$

where the positive integer n is the n^{th} time step and $k = t_n - t_{n-1}$. From (3.6), the nonlinear term $V_m^{n+2}\left[V_{m+1}^{n+2} - V_{m-1}^{n+2}\right]$ is solved by using the linearization method provided in [10] such that $V_m^{n+2}(V_{m+1}^{n+2} - V_{m-1}^{n+2}) \approx U_m^{n+2}(V_{m+1}^{n+2} - V_{m-1}^{n+2})$ approximate V_m^{n+2} by U_m^{n+2}, where U_m^{n+2} is computed by linear extrapolation of V_m^{n+1} and V_m^n. That is

$$V_m^{n+2} \cong U_m^{n+2} = \left(1 + \left(\frac{b_{n+2}}{b_{n+1}}\right)\right)V_m^{n+1} - \left(\frac{b_{n+2}}{b_{n+1}}\right)V_m^n, \quad (3.7)$$

where $b_{n+2} = t_{n+2} - t_{n+1}$ and $b_{n+1} = t_{n+1} - t_n$. Substituting (3.7) into (3.6), we have

$$V_m^{n+2} - V_m^n = \frac{k}{3}\left\{\frac{\mu}{h^2}\left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2}\right] - \frac{1}{2h}\left[\left(1 + \left(\frac{b_{n+2}}{b_{n+1}}\right)\right)V_m^{n+1} - \left(\frac{b_{n+2}}{b_{n+1}}\right)V_m^n\right]\left[V_{m+1}^{n+2} - V_{m-1}^{n+2}\right] + 4\left(\frac{\mu}{h^2}\left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1}\right] - \frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right]\right) + \frac{\mu}{h^2}\left[V_{m+1}^n - 2V_m^n + V_{m-1}^n\right] - \frac{V_m^n}{2h}\left[V_{m+1}^n - V_{m-1}^n\right]\right\}. \quad (3.8)$$

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 (3.8) is rewritten as

$$V_m^{n+2} - V_m^n = \frac{k}{3}\left\{\frac{\mu}{h^2}\left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2}\right] - \frac{1}{2h}\left[2V_m^{n+1} - V_m^n\right]\left[V_{m+1}^{n+2} - V_{m-1}^{n+2}\right] + 4\left(\frac{\mu}{h^2}\left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1}\right] - \frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right]\right) + \frac{\mu}{h^2}\left[V_{m+1}^n - 2V_m^n + V_{m-1}^n\right] - \frac{V_m^n}{2h}\left[V_{m+1}^n - V_{m-1}^n\right]\right\}. \quad (3.9)$$

Then (3.9) can be rewritten in the form

$$\gamma_m V_{m-1}^{n+2} + \delta_m V_m^{n+2} + \lambda_m V_{m+1}^{n+2} = g_m, \quad (3.10)$$

where

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

Since V_m^{n+1} and V_m^n are used to calculate V_m^{n+2} and we have only the initial condition V_m^n. To find V_m^{n+1}, we apply the Runge-Kutta method and Modified-Newton Raphson method [7] to the Burger's equation as described in the next section.

3.3 Modified-Newton Raphson Method

The Burger's equation (3.1) with the central finite difference and the second-order Runge-Kutta method [7] is

$$\begin{aligned}V_m^{n+1}-V_m^n &- \frac{k}{2}\left\{\frac{\mu}{h^2}\left[V_{m+1}^n-2V_m^n+V_{m-1}^n\right]-\frac{V_m^n}{2h}\left[V_{m+1}^n-V_{m-1}^n\right]\right\}\\ &-\frac{k}{2}\left(\frac{\mu}{h^2}\left[V_{m+1}^{n+1}-2V_m^{n+1}+V_{m-1}^{n+1}\right]-\left[\frac{V_m^{n+1}}{2h}+\frac{k}{2h}\left\{\frac{\mu}{h^2}\left[V_{m+1}^{n+1}-2V_m^{n+1}+V_{m-1}^{n+1}\right]\right.\right.\right.\\ &\left.\left.\left.-\frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1}-V_{m-1}^{n+1}\right]\right\}\right]\left[V_{m+1}^{n+1}-V_{m-1}^{n+1}\right]\right)=0.\end{aligned} \tag{3.11}$$

Since (3.11) is a nonlinear equation, we apply Modified-Newton Raphson Method to find the solution of (3.11). Let

$$\begin{aligned}&F_m(V_1^{n+1},V_2^{n+1},V_3^{n+1},...,V_{L-1}^{n+1})\\ &=V_m^{n+1}-V_m^n-\frac{k}{2}\left\{\frac{\mu}{h^2}\left[V_{m+1}^n-2V_m^n+V_{m-1}^n\right]-\frac{V_m^n}{2h}\left[V_{m+1}^n-V_{m-1}^n\right]\right\}\\ &-\frac{k}{2}\left(\frac{\mu}{h^2}\left[V_{m+1}^{n+1}-2V_m^{n+1}+V_{m-1}^{n+1}\right]-\left[\frac{V_m^{n+1}}{2h}+\frac{k}{2h}\left\{\frac{\mu}{h^2}\left[V_{m+1}^{n+1}-2V_m^{n+1}+V_{m-1}^{n+1}\right]\right.\right.\right.\\ &\left.\left.\left.-\frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1}-V_{m-1}^{n+1}\right]\right\}\right]\left[V_{m+1}^{n+1}-V_{m-1}^{n+1}\right]\right).\end{aligned} \tag{3.12}$$

Thus

$$F_m(V_1^{n+1},V_2^{n+1},V_3^{n+1},...,V_{L-1}^{n+1})=0, \tag{3.13}$$

where $m=1,2,...,L-1$. Therefore, (3.13) can be rewritten in a matrix form as

$$J(V^{(0)})\delta^{(P)}=-F(V^{(P)}), \tag{3.14}$$

where

$$J(V^{(0)})=\begin{bmatrix}\frac{\partial F_1(V^{(0)})}{\partial V_1^{n+1}} & \frac{\partial F_1(V^{(0)})}{\partial V_2^{n+1}} & \cdots & \frac{\partial F_1(V^{(0)})}{\partial V_{L-1}^{n+1}}\\ \frac{\partial F_2(V^{(0)})}{\partial V_1^{n+1}} & \frac{\partial F_2(V^{(0)})}{\partial V_2^{n+1}} & \cdots & \frac{\partial F_2(V^{(0)})}{\partial V_{L-1}^{n+1}}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial F_{L-1}(V^{(0)})}{\partial V_1^{n+1}} & \frac{\partial F_{L-1}(V^{(0)})}{\partial V_2^{n+1}} & \cdots & \frac{\partial F_{L-1}(V^{(0)})}{\partial V_{L-1}^{n+1}}\end{bmatrix}$$

$$\delta^{(P)}=\begin{bmatrix}\Delta V_1^{n+1}\\ \Delta V_2^{n+1}\\ \vdots\\ \Delta V_{L-1}^{n+1}\end{bmatrix}^{(P)} \quad \text{and} \quad F(V^{(P)})=\begin{bmatrix}F_1\\ F_2\\ \vdots\\ F_{L-1}\end{bmatrix}^{(P)},$$

where P is the number of iterations, $V^{(0)} = (V_1^{n+1}, V_2^{n+1}, ..., V_{L-1}^{n+1})^{(0)}$, $(\Delta V_m^{n+1})^{(P)} = (V_m^{n+1})^{(P+1)} - (V_m^{n+1})^{(P)}$ 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 real number. Then V_m^{n+1} can be calculated from (3.14). Now, we have V_m^n and V_m^{n+1}. Therefore, V_m^{n+2} can be determined from (3.10), which the numerical solutions are presented in the next chapter.

Chapter 4

The Numerical Results

In this chapter, we find the numerical solution of the Burger's equation by using a computer program. The numerical solutions obtained are compared with the exact solutions to verify the results.

4.1 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 [15],

$$v(x,t) = \frac{2\mu\pi e^{-\mu\pi^2 t}\sin(\pi x)}{a + e^{-\mu\pi^2 t}\cos(\pi x)}, \qquad 0 < x < 1 \tag{4.1}$$

with boundary conditions

$$\left.\begin{aligned} v(0,t) &= 0 \\ v(1,t) &= 0 \end{aligned}\right\} \quad t > 0 \tag{4.2}$$

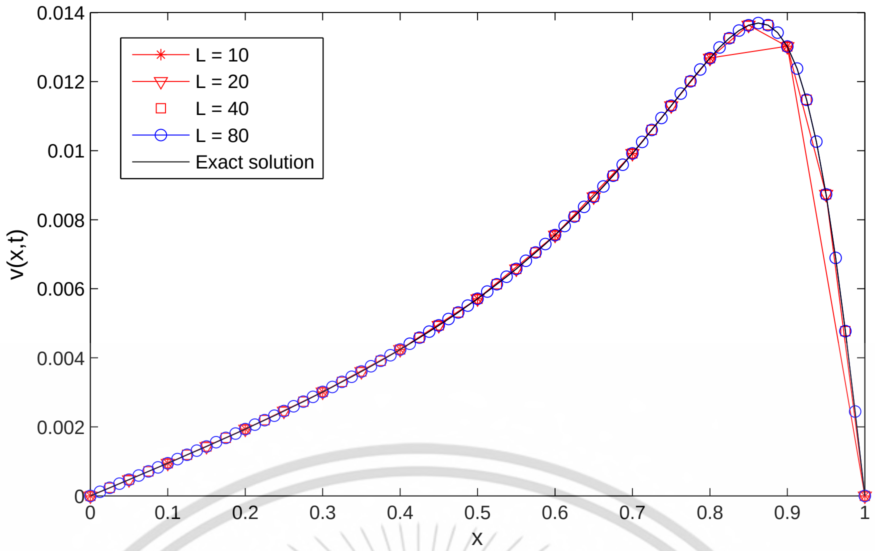
and the initial condition

$$v(x,0) = \frac{2\mu\pi \sin(\pi x)}{a + \cos(\pi x)}, \qquad a > 1. \tag{4.3}$$

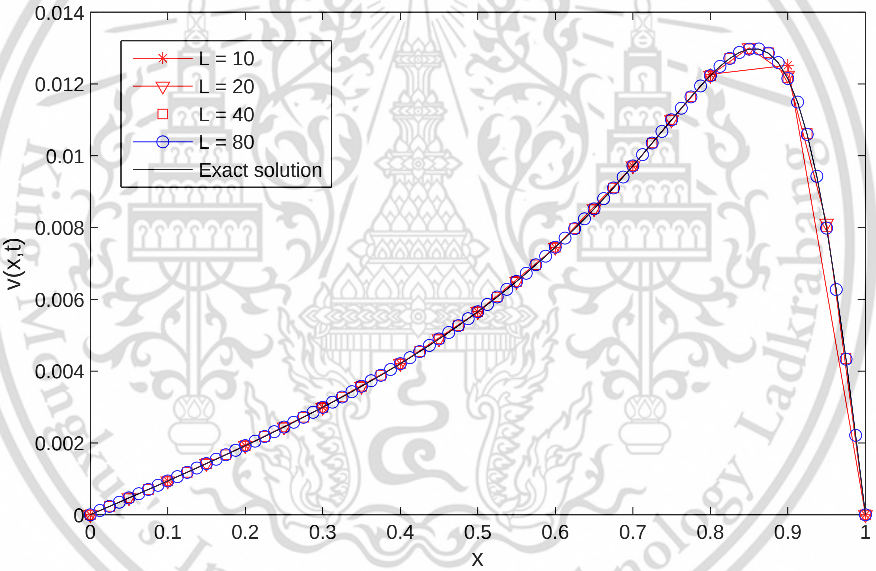
The equation (4.1) - (4.3) will be used in Section 4.2 to demonstrate the accuracy of the numerical solutions.

4.2 Numerical solutions

In this section, we provide the numerical results obtained from (3.10). To verify the solutions, we first compare the results with the exact solution. Figures 4.1a and 4.1b illustrate the exact and numerical solutions at different number of grid points with boundary conditions (4.2) and initial condition (4.3) for $t = 0.02$ and $t = 1$, respectively, 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 (3.10), 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.



(a)



(b)

Figure 4.1: The exact and numerical solutions at different values of L when $a = 1.1, \mu = 0.001$ for (4.1a) $t = 0.02$ and (4.1b) $t = 1$.

Figures 4.1a and 4.1b show that when the number of grids increases, the numerical results converge to the exact solution. The discrete results of the numerical and exact solutions at $t = 0.02$ and $t = 1$ are shown in Tables 4.1 and 4.2, respectively, in which we show the values at the point $x = 0.1, 0.2, ..., 0.9$.

Table 4.1: Comparison of the numerical solutions and the exact solution at different values of L when $a = 1.1, t = 0.02, \mu = 0.001$ and $\Delta t = 0.01$.

x	Numerical solution				Exact solution
	$L = 10$	$L = 20$	$L = 40$	$L = 80$	
0.1	9.4654E-04	9.4654E-04	9.4654E-04	9.4654E-04	0.000947
0.2	1.9344E-03	1.9344E-03	1.9344E-03	1.9344E-03	0.001934
0.3	3.0114E-03	3.0114E-03	3.0114E-03	3.0114E-03	0.003011
0.4	4.2404E-03	4.2404E-03	4.2404E-03	4.2404E-03	0.004240
0.5	5.7108E-03	5.7109E-03	5.7109E-03	5.7109E-03	0.005711
0.6	7.5526E-03	7.5526E-03	7.5527E-03	7.5527E-03	0.007553
0.7	9.9197E-03	9.9197E-03	9.9198E-03	9.9198E-03	0.009920
0.8	1.2683E-02	1.2683E-02	1.2683E-02	1.2683E-02	0.012683
0.9	1.3026E-02	1.3019E-02	1.3018E-02	1.3017E-02	0.013017

Table 4.2: Comparison of the numerical solutions and 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				Exact solution
	$L = 10$	$L = 20$	$L = 40$	$L = 80$	
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

The errors of the solutions at $t = 0.02$ and $t = 1$ are illustrated in Table 4.3, where L_2-norm and $L_\infty-$norm are

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

$$L_\infty - \text{norm} = ||v - V||_\infty = \max_j |v_j - V_j|,$$

where v and V represent the values of the exact and numerical solutions, respectively.

Table 4.3: The values of errors in L_2-norm and $L_\infty-$norm of the numerical solutions with different values of L at $t = 0.02$ and $t = 1$.

L	$t = 0.02$		$t = 1$	
	L_2-norm	$L_\infty-$norm	L_2-norm	$L_\infty-$norm
10	8.7898E-06	8.7632E-06	3.8091E-04	3.7899E-04
20	4.6896E-06	3.9726E-06	1.8308E-04	1.4753E-04
40	1.7822E-06	1.0899E-06	6.6726E-05	3.8173E-05
80	6.4187E-07	2.7900E-07	2.3750E-05	9.6834E-06

Table 4.3 shows the errors L_2-norm and $L_\infty-$norm of the numerical solutions with four different numbers of grid points $L = 10, 20, 40$ and 80. It is clear from the table that the errors decrease when the number of grids increases. Figure 4.2 shows the numerical solutions when the constant a in the initial condition increases with the exact solutions, in which $a = 1.1, 2, 4,$ and, 10 for $T = 1$, $\mu = 0.001, \Delta t = 0.01$ and $h = 0.0125$. From Figure 4.2, the different values of a affect the pattern of the graph and the numerical solutions are still closed to the exact solutions for each value of a. Figure 4.3 compares the numerical solutions and the exact solutions with four different values of a. The increasing variables a decrease the values of the numerical results. Figure 4.4 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.

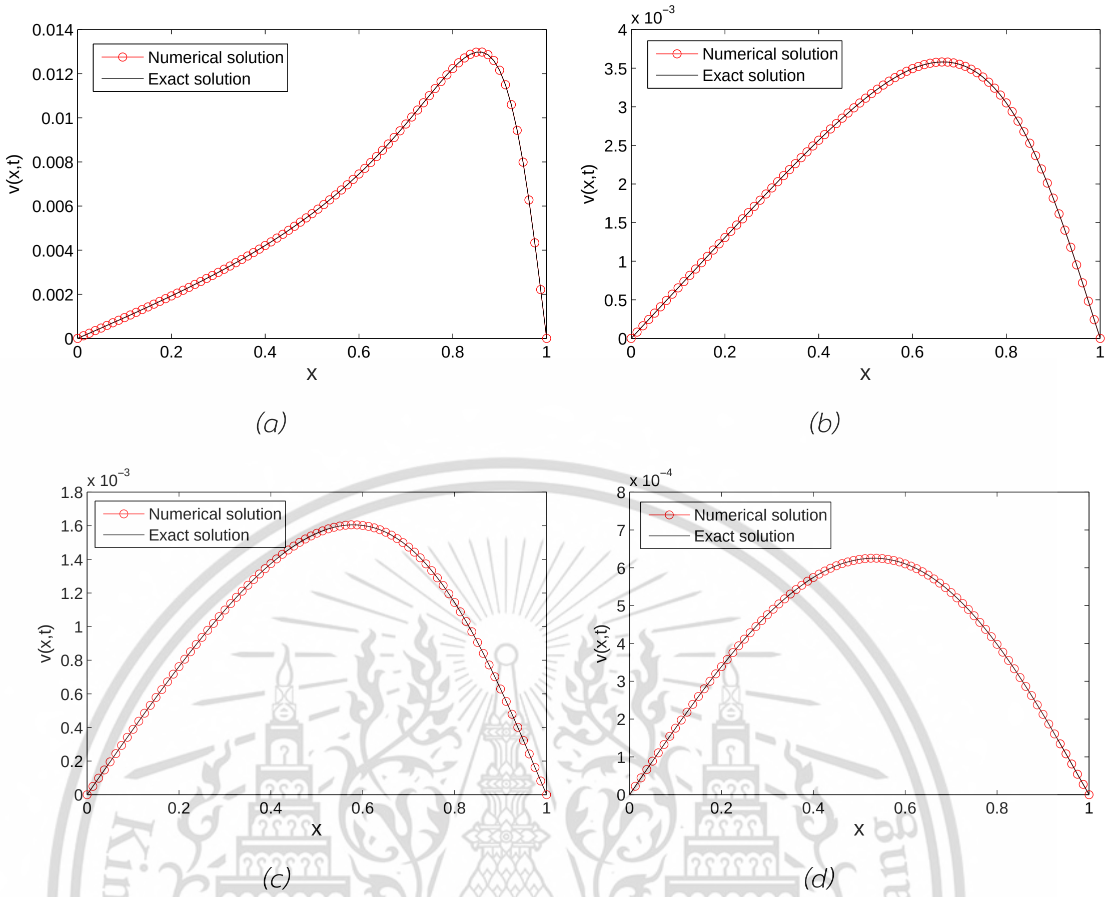


Figure 4.2: The exact and numerical solutions at $T = 1, \mu = 0.001$, and $h = 0.0125$ for (4.2a) $a = 1.1$, (4.2b) $a = 2$, (4.2c) $a = 4$ and (4.2d) $a = 10$.

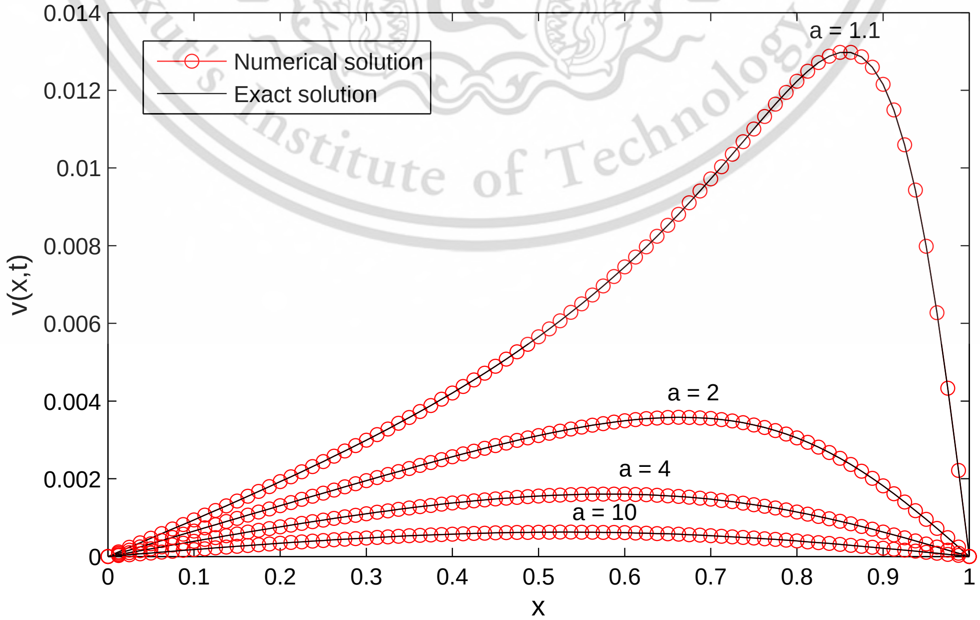


Figure 4.3: Numerical solutions for different values of a with $\mu = 0.001, T = 1$ and $h = 0.0125$.

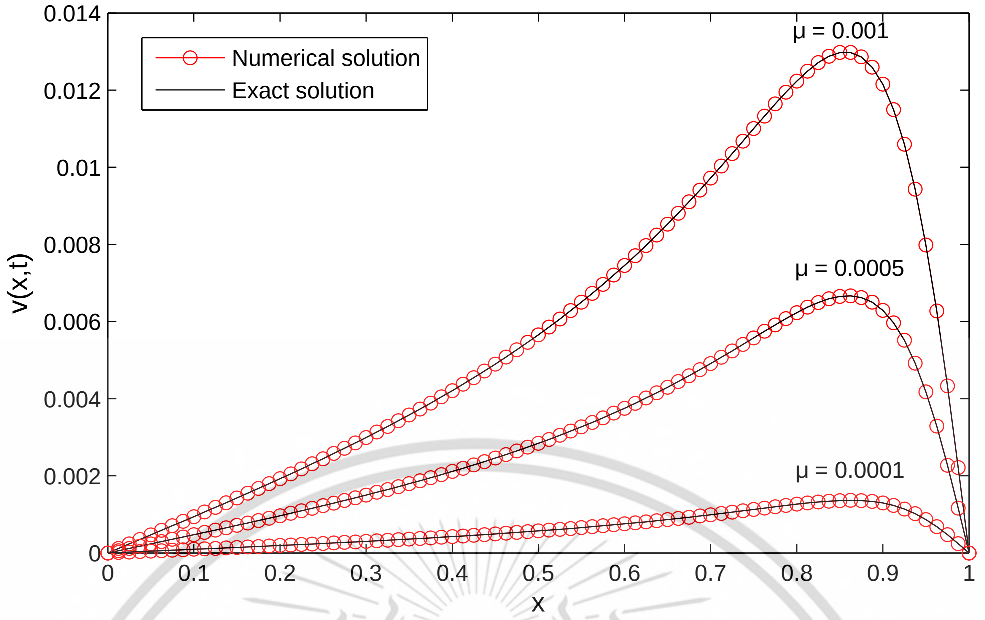
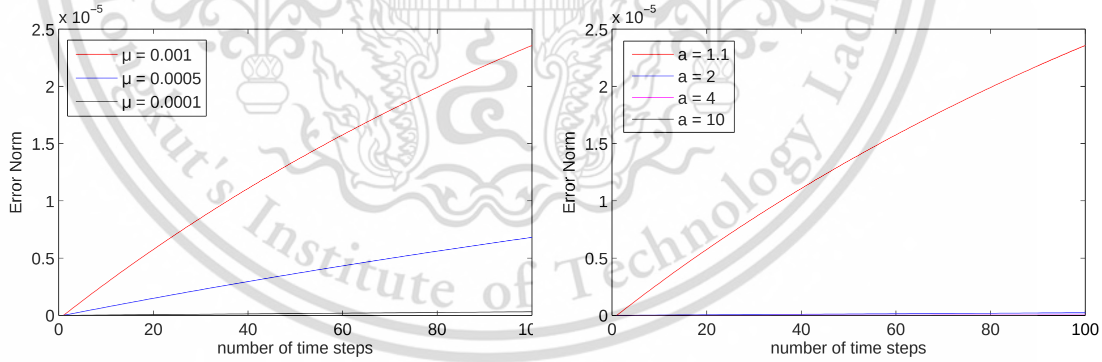


Figure 4.4: Numerical solutions for different values of μ with $a = 1.1, T = 1$ and $h = 0.0125$.

The L_2-norm errors of the numerical results in Figures 4.2 - 4.4 are illustrated in Figure 4.5 and Table 4.4 for the different values of a and μ. It is shown that the L_2-norm errors decrease when a increases and/or μ decreases for both $\Delta t = 0.01$ and $\Delta t = 0.001$. Moreover, the errors decrease with decreasing Δt.



(a) (b)

Figure 4.5: L_2-norm errors for $\Delta t = 0.01, h = 0.0125$ at (4.5a) different μ with fixed $a = 1.1$ and (4.5b) different a with fixed $\mu = 0.001$.

Table 4.4: The L_2-norm errors for different values of a and μ with $\Delta t = 0.01$ and $\Delta t = 0.001$.

Δt		$a = 1.1$	$a = 2$	$a = 4$	$a = 10$
0.01	$\mu = 0.001$	2.37502E-05	2.24741E-07	3.02647E-08	6.16947E-09
	$\mu = 0.0005$	6.86689E-06	5.76347E-08	7.67045E-09	1.55427E-09
	$\mu = 0.0001$	3.12268E-07	2.35362E-09	3.10223E-10	6.28273E-11
0.001	$\mu = 0.001$	2.37478E-05	2.24737E-07	3.02644E-08	6.16944E-09
	$\mu = 0.0005$	6.86671E-06	5.76344E-08	7.67043E-09	1.55427E-09
	$\mu = 0.0001$	3.12268E-07	2.35362E-09	3.10223E-10	6.25833E-11

Table 4.5 shows the L_2-norm errors of our numerical solutions compared with the Euler forward discretization [4] and Mac Cormack discretization [12] at different value of μ for $\Delta t = 0.01$ and 0.001, where the L_2-norm used in [4] and [12] is

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

which is employed in Table 4.5, 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 [4] and [12].

Table 4.5: Comparison of the L_2-norm errors of our numerical solutions with [4] and [12] for $a = 1.1, h = 0.0125, T = 1$, and $\mu = 0.001$ and 0.0001.

μ	Δt	[4]	[12]	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

Chapter 5

Conclusion

5.1 Conclusion

In this study, 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 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 Figure 4.1 with $t = 0.02$ and $t = 1$, where the discrete errors are provided in Tables 4.1 and 4.2. They show that the numerical results converge to the exact solutions when the number of grids increases. The errors L_2-norm and $L_\infty-$norm are shown in Table 4.3 to determine the accuracy of our numerical solutions. Figure 4.2 shows that the values of a in the initial condition affect the pattern of graph of the solution of the Burger's equation. The numerical results and exact solutions reduces in height for small value of μ but vice versa with the constant a as shown in Figures 4.3 and 4.4, respectively. It is cleared from Figure 4.5 that the L_2-norm errors decreases with decreasing values of μ and increasing values of a. 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, where the errors are provided in Table 4.4. As compared to other numerical methods shown in Table 4.5, our numerical solutions are more accurate.

5.2 Suggestion

Although the methods in this research work well with the Burger's equation, the same idea can be applied to time-dependent nonlinear PDE problems. Equations that are comparable to the Burger's equation can be solved by using these approaches.

References

[1] Ali, K. Yucel, C. and Orkun, T. 2016. "Exact Solution for the Conformable Burgers' Equation by the Hopf-Cole Transform." *Cankaya University Journal of Science and Engineering*. 13(2) : 18-23.

[2] Ali, Z.Y. 2018. "Application New Iterative Method For Solving Nonlinear Burger'a Equation And Coupled Burger's Equations." *International Journal of Computer Science Issues*. 15(3) : 31-35

[3] Bateman, M. 1915. "Some recent researches on the motion of fluids." *Monthly Weather Review*. 43(4) : 163-170.

[4] Bui, T. 2010. "Explicit and Implicit Methods In Solving Differential Equations." *Honors Scholar Theses*. 119.

[5] Burgers, J.M. 1948. "A Mathematical Model Illustrating the Theory of Turbulence." *Advances in Applied Mechanics*. 171-199.

[6] Çelikten, G. and AKSAN, E. 2017. "Explicit exponential finite difference methods for the numerical solution of modified Burgers' equation." *Eastern Anatolian Journal of Science*. 3(1) : 45-50.

[7] Chaiyasit, A. Klinthub, T. Arleemeen, K. Sanprasert, W. and Wuttanachamsri, K. 2019. "A Second-Order Accurate Numerical Approach for Burger's Equation." *The 11th NPRU National Acadmic Conference Nakorn Pathom Rajabhat University, Nakhon Pathom, Thailand*. 82-89.

[8] Kincaid, D. and Cheney, W. 1991. "Mathematics of Scientific Computing." *Published by American Mathematical Society*.

[9] Mayur, P. Ashish, A. Lakshmi, C. Vijitha, M. and Aswin, V.S. 2018. "A systematic literature review of Burger's equation with recent advasnce." *Pramana*. 90(6) : 1-21.

[10] Mohamed, N.A. 2018. "Fully Implicit Scheme for Solving Burgers' Equation based on Finite Difference Method." *Egyptian International Journal of Engineering Sciences and Technology*. 26 : 38-44.

[11] Mohamed, N.A. 2019. "Solving one- and two-dimensional Burgers equation using fully implicit finite difference schemes." *Arab Journal of Basic and Applied Sciences*. 26(1) : 254-268.

[12] Perrin, A. Hu, H. H. 2006. "An explicit finite-difference scheme for simulation of moving particles." *Journal of Computational Physics*. 212(1), 166-187.

[13] Sheikh, M.A. Andallah, L. and Kowser, A. 2014. “A Comparative Study of Finite Difference Scheme for Burger’s Equation.” *Gazi University Journal of Science*. 27(4) : 1045-1051.

[14] Ucar, Y. Yagmurlu, N.M. and Tasbozan, O. 2017. “Numerical Solution of the Modified Burgers’ Equation by Finite Difference Methods.” *JOURNAL OF APPLIED MATHEMATICS STATISTICS AND INFORMATICS*. 13(1) : 19-30.

[15] Wood, W.L. 2006. “An exact solution for Burger’s equation.” *Communications in Numerical Methods in Engineering*. 22(7) : 797-798.

[16] Zang, P.G. and Wang, J.P. 2012. “A predictor-corrector compact finite difference scheme for Burgers’ equation.” *Journal of Applied Mathematics and Computation*. 219(3) : 892-898.

Appendix

Appendix A

The exact and numerical solutions for $L = 10, 20, 40$, and 80 at $t = 0.02$ and $t = 1$, respectively, are compared in the following tables as well as the point wise errors.

Table 5.1*:* Comparison of the exact solutions and the numerical solutions at $t = 0.02$ and $L = 10$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.1	0.0009465392	0.0009465389	0.0000000003
0.2	0.0019343692	0.0019343681	0.0000000010
0.3	0.0030113723	0.0030113693	0.0000000030
0.4	0.0042403630	0.0042403548	0.0000000082
0.5	0.0057108593	0.0057108375	0.0000000217
0.6	0.0075526582	0.0075526030	0.0000000551
0.7	0.0099197643	0.0099196713	0.0000000930
0.8	0.0126825606	0.0126832354	0.0000006748
0.9	0.0130169131	0.0130256762	0.0000087632
1	0	0	0

Table 5.2*:* Comparison of the exact solutions and the numerical solutions at $t = 0.02$ and $L = 20$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.05	0.0004707621	0.0004707620	0.000000000032
0.10	0.0009465392	0.0009465392	0.000000000075
0.15	0.0014325394	0.0014325393	0.000000000144
0.20	0.0019343692	0.0019343689	0.000000000257
0.25	0.0024582660	0.0024582655	0.000000000443
0.30	0.0030113723	0.0030113716	0.000000000745
0.35	0.0036020664	0.0036020651	0.000000001233
0.40	0.0042403630	0.0042403610	0.000000002023
0.45	0.0049383845	0.0049383812	0.000000003301
0.50	0.0057108593	0.0057108539	0.000000005367
0.55	0.0065754850	0.0065754763	0.000000008682
0.60	0.0075526582	0.0075526443	0.000000013829
0.65	0.0086631510	0.0086631301	0.000000020980
0.70	0.0099197643	0.0099197373	0.000000027022

x	Exact solution	Numerical solution	error value
0.75	0.0113018716	0.0113018592	0.000000012413
0.80	0.0126825606	0.0126826737	0.000000113097
0.85	0.0136346153	0.0136353008	0.000000685506
0.90	0.0130169131	0.0130193060	0.000002392938
0.95	0.0087347117	0.0087386843	0.000003972580
1	0	0	0

Table 5.3: Comparison of the exact solutions and the numerical solutions at $t = 0.02$ and $L = 40$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.03	0.0002350698	0.0002350698	0.0000000000038
0.05	0.0004707621	0.0004707621	0.0000000000079
0.08	0.0007077051	0.0007077051	0.0000000000128
0.10	0.0009465392	0.0009465392	0.0000000000188
0.13	0.0011879230	0.0011879229	0.0000000000263
0.15	0.0014325394	0.0014325394	0.0000000000359
0.18	0.0016811035	0.0016811035	0.0000000000483
0.20	0.0019343692	0.0019343691	0.0000000000641
0.23	0.0021931376	0.0021931375	0.0000000000844
0.25	0.0024582660	0.0024582659	0.0000000001104
0.28	0.0027306776	0.0027306775	0.0000000001435
0.30	0.0030113723	0.0030113721	0.0000000001857
0.33	0.0033014385	0.0033014383	0.0000000002393
0.35	0.0036020664	0.0036020661	0.0000000003074
0.38	0.0039145620	0.0039145616	0.0000000003941
0.40	0.0042403630	0.0042403625	0.0000000005042
0.43	0.0045810543	0.0045810536	0.0000000006442
0.45	0.0049383845	0.0049383837	0.0000000008224
0.48	0.0053142806	0.0053142796	0.0000000010492
0.50	0.0057108593	0.0057108579	0.0000000013376
0.53	0.0061304301	0.0061304284	0.0000000017035
0.55	0.0065754850	0.0065754828	0.0000000021657
0.58	0.0070486609	0.0070486581	0.0000000027445
0.60	0.0075526582	0.0075526547	0.0000000034579

x	Exact solution	Numerical solution	error value
0.63	0.0080900815	0.0080900772	0.0000000043125
0.65	0.0086631510	0.0086631458	0.0000000052817
0.68	0.0092731933	0.0092731871	0.0000000062588
0.70	0.0099197643	0.0099197574	0.0000000069574
0.73	0.0105991550	0.0105991483	0.0000000067010
0.75	0.0113018716	0.0113018676	0.0000000039894
0.78	0.0120084405	0.0120084449	0.0000000043692
0.80	0.0126825606	0.0126825855	0.0000000249047
0.83	0.0132603411	0.0132604113	0.0000000702547
0.85	0.0136346153	0.0136347779	0.0000001625592
0.88	0.0136356094	0.0136359426	0.0000003331233
0.90	0.0130169131	0.0130175169	0.0000006037844
0.93	0.0114729154	0.0114738408	0.0000009253561
0.95	0.0087347117	0.0087358016	0.0000010899038
0.98	0.0047722546	0.0047730448	0.0000007901877
1	0	0	0

Table 5.4: Comparison of the exact solutions and the numerical solutions at $t = 0.02$ and $L = 80$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.01	0.0001174961	0.0001174961	0.0000000000005
0.03	0.0002350698	0.0002350698	0.0000000000009
0.04	0.0003527991	0.0003527991	0.0000000000015
0.05	0.0004707621	0.0004707621	0.0000000000020
0.06	0.0005890376	0.0005890376	0.0000000000026
0.08	0.0007077051	0.0007077051	0.0000000000032
0.09	0.0008268451	0.0008268451	0.0000000000039
0.10	0.0009465392	0.0009465392	0.0000000000047
0.11	0.0010668704	0.0010668704	0.0000000000056
0.13	0.0011879230	0.0011879230	0.0000000000066
0.14	0.0013097832	0.0013097832	0.0000000000077
0.15	0.0014325394	0.0014325394	0.0000000000090
0.16	0.0015562819	0.0015562819	0.0000000000104
0.18	0.0016811035	0.0016811035	0.0000000000121

x	Exact solution	Numerical solution	error value
0.19	0.0018070998	0.0018070998	0.0000000000139
0.20	0.0019343692	0.0019343691	0.0000000000160
0.21	0.0020630134	0.0020630133	0.0000000000184
0.23	0.0021931376	0.0021931375	0.0000000000211
0.24	0.0023248507	0.0023248507	0.0000000000241
0.25	0.0024582660	0.0024582660	0.0000000000276
0.26	0.0025935008	0.0025935008	0.0000000000315
0.28	0.0027306776	0.0027306776	0.0000000000359
0.29	0.0028699238	0.0028699237	0.0000000000408
0.30	0.0030113723	0.0030113723	0.0000000000464
0.31	0.0031551621	0.0031551621	0.0000000000527
0.33	0.0033014385	0.0033014384	0.0000000000598
0.34	0.0034503535	0.0034503534	0.0000000000678
0.35	0.0036020664	0.0036020663	0.0000000000768
0.36	0.0037567441	0.0037567440	0.0000000000870
0.38	0.0039145620	0.0039145619	0.0000000000984
0.39	0.0040757039	0.0040757038	0.0000000001114
0.40	0.0042403630	0.0042403629	0.0000000001259
0.41	0.0044087421	0.0044087419	0.0000000001424
0.43	0.0045810543	0.0045810541	0.0000000001609
0.44	0.0047575234	0.0047575233	0.0000000001818
0.45	0.0049383845	0.0049383843	0.0000000002054
0.46	0.0051238841	0.0051238838	0.0000000002321
0.48	0.0053142806	0.0053142804	0.0000000002621
0.49	0.0055098448	0.0055098445	0.0000000002959
0.50	0.0057108593	0.0057108589	0.0000000003341
0.51	0.0059176189	0.0059176185	0.0000000003771
0.53	0.0061304301	0.0061304297	0.0000000004256
0.54	0.0063496099	0.0063496095	0.0000000004800
0.55	0.0065754850	0.0065754844	0.0000000005411
0.56	0.0068083891	0.0068083885	0.0000000006095
0.58	0.0070486609	0.0070486602	0.0000000006859
0.59	0.0072966392	0.0072966385	0.0000000007707
0.60	0.0075526582	0.0075526573	0.0000000008645
0.61	0.0078170392	0.0078170382	0.0000000009673
0.63	0.0080900815	0.0080900805	0.0000000010788

x	Exact solution	Numerical solution	error value
0.64	0.0083720487	0.0083720475	0.0000000011980
0.65	0.0086631510	0.0086631497	0.0000000013226
0.66	0.0089635231	0.0089635217	0.0000000014488
0.68	0.0092731933	0.0092731918	0.0000000015701
0.69	0.0095920448	0.0095920431	0.0000000016763
0.70	0.0099197643	0.0099197626	0.0000000017516
0.71	0.0102557760	0.0102557742	0.0000000017719
0.73	0.0105991550	0.0105991533	0.0000000017014
0.74	0.0109485163	0.0109485148	0.0000000014871
0.75	0.0113018716	0.0113018705	0.0000000010514
0.76	0.0116564452	0.0116564450	0.0000000002818
0.78	0.0120084405	0.0120084415	0.0000000009837
0.79	0.0123527446	0.0123527475	0.0000000029754
0.80	0.0126825606	0.0126825666	0.0000000060168
0.81	0.0129889588	0.0129889693	0.0000000105542
0.83	0.0132603411	0.0132603583	0.0000000171884
0.84	0.0134818301	0.0134818568	0.0000000267016
0.85	0.0136346153	0.0136346554	0.0000000400597
0.86	0.0136953310	0.0136953893	0.0000000583605
0.88	0.0136356094	0.0136356921	0.0000000826719
0.89	0.0134220438	0.0134221575	0.0000001136895
0.90	0.0130169131	0.0130170642	0.0000001511399
0.91	0.0123801316	0.0123803245	0.0000001929184
0.93	0.0114729154	0.0114731495	0.0000002341275
0.94	0.0102634968	0.0102637633	0.0000002664962
0.95	0.0087347117	0.0087349907	0.0000002789982
0.96	0.0068923774	0.0068926379	0.0000002604761
0.98	0.0047722546	0.0047724588	0.0000002042320
0.99	0.0024426309	0.0024427438	0.0000001128354
1	0	0	0

Table 5.5: Comparison of the exact solutions and the numerical solutions at $t = 1$ and $L = 10$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.1	0.0009416306	0.0009416141	0.0000000165
0.2	0.0019235956	0.0019235407	0.0000000549
0.3	0.0029924159	0.0029922603	0.0000001556
0.4	0.0042084297	0.0042080149	0.0000004148
0.5	0.0056558889	0.0056548198	0.0000010691
0.6	0.0074519442	0.0074494065	0.0000025377
0.7	0.0097169951	0.0097140443	0.0000029508
0.8	0.0122333400	0.0122712930	0.0000379530
0.9	0.0121461675	0.0125251618	0.0003789943
1	0	0	0

Table 5.6: Comparison of the exact solutions and the numerical solutions at $t = 1$ and $L = 20$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.05	0.0004683634	0.0004683616	0.000000001753
0.10	0.0009416306	0.0009416265	0.000000004093
0.15	0.0014248876	0.0014248799	0.000000007719
0.20	0.0019235956	0.0019235820	0.000000013586
0.25	0.0024438070	0.0024437839	0.000000023101
0.30	0.0029924159	0.0029923775	0.000000038440
0.35	0.0035774530	0.0035773899	0.000000063044
0.40	0.0042084297	0.0042083273	0.000000102423
0.45	0.0048967171	0.0048965518	0.000000165349
0.50	0.0056558889	0.0056556235	0.000000265362
0.55	0.0065018186	0.0065013974	0.000000421269
0.60	0.0074519442	0.0074512941	0.000000650128
0.65	0.0085221346	0.0085212097	0.000000924941
0.70	0.0097169951	0.0097160090	0.000000986106
0.75	0.0110025855	0.0110029872	0.000000401684
0.80	0.0122333400	0.0122413078	0.000007967822
0.85	0.0129716109	0.0130072468	0.000035635910

x	Exact solution	Numerical solution	error value
0.90	0.0121461675	0.0122482299	0.000102062394
0.95	0.0079767162	0.0081242414	0.000147525199
1	0	0	0

Table 5.7: Comparison of the exact solutions and the numerical solutions at $t = 1$ and $L = 40$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.03	0.0002338773	0.0002338771	0.0000000002100
0.05	0.0004683634	0.0004683629	0.0000000004376
0.08	0.0007040726	0.0007040719	0.0000000007014
0.10	0.0009416306	0.0009416296	0.0000000010214
0.13	0.0011816802	0.0011816788	0.0000000014207
0.15	0.0014248876	0.0014248857	0.0000000019259
0.18	0.0016719485	0.0016719459	0.0000000025688
0.20	0.0019235956	0.0019235922	0.0000000033884
0.23	0.0021806054	0.0021806010	0.0000000044322
0.25	0.0024438070	0.0024438013	0.0000000057596
0.28	0.0027140903	0.0027140829	0.0000000074448
0.30	0.0029924159	0.0029924064	0.0000000095813
0.33	0.0032798255	0.0032798132	0.0000000122870
0.35	0.0035774530	0.0035774373	0.0000000157113
0.38	0.0038865363	0.0038865162	0.0000000200437
0.40	0.0042084297	0.0042084042	0.0000000255239
0.43	0.0045446160	0.0045445836	0.0000000324545
0.45	0.0048967171	0.0048966759	0.0000000412148
0.48	0.0052665020	0.0052664497	0.0000000522734
0.50	0.0056558889	0.0056558227	0.0000000661965
0.53	0.0060669368	0.0060668532	0.0000000836392
0.55	0.0065018186	0.0065017133	0.0000001052989
0.58	0.0069627629	0.0069626311	0.0000001317826
0.60	0.0074519442	0.0074517810	0.0000001632924
0.63	0.0079712881	0.0079710892	0.0000001989336
0.65	0.0085221346	0.0085218993	0.0000002352535
0.68	0.0091046723	0.0091044090	0.0000002632288

x	Exact solution	Numerical solution	error value
0.70	0.0097169951	0.0097167329	0.0000002621885
0.73	0.0103535477	0.0103533598	0.0000001878391
0.75	0.0110025855	0.0110026360	0.0000000504815
0.78	0.0116420878	0.0116427182	0.0000006303823
0.80	0.0122333400	0.0122352207	0.0000018806275
0.83	0.0127113229	0.0127156809	0.0000043579822
0.85	0.0129716109	0.0129804752	0.0000088642578
0.88	0.0128560169	0.0128722018	0.0000161849502
0.90	0.0121461675	0.0121723122	0.0000261446814
0.93	0.0105873848	0.0106231759	0.0000357911714
0.95	0.0079767162	0.0080148896	0.0000381733760
0.98	0.0043245911	0.0043504145	0.0000258233948
1	0	0	0

Table 5.8: Comparison of the exact solutions and the numerical solutions at $t = 1$ and $L = 80$.

x	Exact solution	Numerical solution	error value
0	0	0	0
0.01	0.0001169007	0.0001169006	0.0000000000260
0.03	0.0002338773	0.0002338772	0.0000000000525
0.04	0.0003510060	0.0003510060	0.0000000000801
0.05	0.0004683634	0.0004683633	0.0000000001094
0.06	0.0005860263	0.0005860262	0.0000000001409
0.08	0.0007040726	0.0007040724	0.0000000001753
0.09	0.0008225808	0.0008225806	0.0000000002132
0.10	0.0009416306	0.0009416303	0.0000000002552
0.11	0.0010613029	0.0010613026	0.0000000003023
0.13	0.0011816802	0.0011816799	0.0000000003550
0.14	0.0013028465	0.0013028461	0.0000000004143
0.15	0.0014248876	0.0014248871	0.0000000004812
0.16	0.0015478915	0.0015478910	0.0000000005567
0.18	0.0016719485	0.0016719479	0.0000000006418
0.19	0.0017971514	0.0017971506	0.0000000007380
0.20	0.0019235956	0.0019235947	0.0000000008466
0.21	0.0020513797	0.0020513787	0.0000000009691

x	Exact solution	Numerical solution	error value
0.23	0.0021806054	0.0021806043	0.0000000011073
0.24	0.0023113782	0.0023113769	0.0000000012632
0.25	0.0024438070	0.0024438056	0.0000000014389
0.26	0.0025780052	0.0025780036	0.0000000016369
0.28	0.0027140903	0.0027140885	0.0000000018598
0.29	0.0028521848	0.0028521826	0.0000000021109
0.30	0.0029924159	0.0029924135	0.0000000023935
0.31	0.0031349167	0.0031349140	0.0000000027115
0.33	0.0032798255	0.0032798225	0.0000000030693
0.34	0.0034272872	0.0034272838	0.0000000034719
0.35	0.0035774530	0.0035774491	0.0000000039247
0.36	0.0037304809	0.0037304764	0.0000000044340
0.38	0.0038865363	0.0038865313	0.0000000050068
0.39	0.0040457922	0.0040457866	0.0000000056511
0.40	0.0042084297	0.0042084234	0.0000000063757
0.41	0.0043746383	0.0043746311	0.0000000071907
0.43	0.0045446160	0.0045446079	0.0000000081071
0.44	0.0047185702	0.0047185610	0.0000000091375
0.45	0.0048967171	0.0048967068	0.0000000102958
0.46	0.0050792826	0.0050792710	0.0000000115974
0.48	0.0052665020	0.0052664889	0.0000000130593
0.49	0.0054586196	0.0054586049	0.0000000147000
0.50	0.0056558889	0.0056558723	0.0000000165396
0.51	0.0058585716	0.0058585530	0.0000000185994
0.53	0.0060669368	0.0060669159	0.0000000209016
0.54	0.0062812596	0.0062812361	0.0000000234687
0.55	0.0065018186	0.0065017923	0.0000000263221
0.56	0.0067288940	0.0067288645	0.0000000294805
0.58	0.0069627629	0.0069627299	0.0000000329573
0.59	0.0072036947	0.0072036579	0.0000000367565
0.60	0.0074519442	0.0074519034	0.0000000408666
0.61	0.0077077432	0.0077076979	0.0000000452527
0.63	0.0079712881	0.0079712383	0.0000000498433
0.64	0.0082427260	0.0082426715	0.0000000545128
0.65	0.0085221346	0.0085220755	0.0000000590558
0.66	0.0088094976	0.0088094344	0.0000000631505

x	Exact solution	Numerical solution	error value
0.68	0.0091046723	0.0091046060	0.0000000663063
0.69	0.0094073484	0.0094072806	0.0000000677908
0.70	0.0097169951	0.0097169286	0.0000000665262
0.71	0.0100327930	0.0100327320	0.0000000609457
0.73	0.0103535477	0.0103534989	0.0000000487955
0.74	0.0106775797	0.0106775529	0.0000000268655
0.75	0.0110025855	0.0110025949	0.0000000093677
0.76	0.0113254619	0.0113255281	0.0000000662048
0.78	0.0116420878	0.0116422401	0.0000001522985
0.79	0.0119470548	0.0119473341	0.0000002793243
0.80	0.0122333400	0.0122338027	0.0000004626810
0.81	0.0124919193	0.0124926414	0.0000007220685
0.83	0.0127113229	0.0127124046	0.0000010816438
0.84	0.0128771541	0.0128787234	0.0000015692480
0.85	0.0129716109	0.0129738249	0.0000022139217
0.86	0.0129730916	0.0129761323	0.0000030406718
0.88	0.0128560169	0.0128600783	0.0000040613597
0.89	0.0125910748	0.0125963358	0.0000052610079
0.90	0.0121461675	0.0121527478	0.0000065802246
0.91	0.0114883953	0.0114962926	0.0000078973022
0.93	0.0105873848	0.0105964026	0.0000090177868
0.94	0.0094200830	0.0094297665	0.0000096834265
0.95	0.0079767162	0.0079863289	0.0000096127287
0.96	0.0062669518	0.0062755286	0.0000085768465
0.98	0.0043245911	0.0043310861	0.0000064950327
0.99	0.0022087560	0.0022122666	0.0000035106485
1	0	0	0

Appendix B

IAENG International Journal of Applied Mathematics, 51:2, IJAM_51_2_20

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 multi-step 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

BURGER'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 by using the finite difference methods. G. Çelikten 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 fourth-order compact finite difference method for solving the one-dimensional 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 time-dependent Burger's equation

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

subjected to the boundary conditions

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

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

and the initial condition

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

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.

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

A. The Second-Order Finite Difference Method

Consider the Burger's equation

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

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 $\alpha = x_0 \le x_1 \le ... \le x_L = \beta$. 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

$$\begin{aligned}\left[\frac{\partial v}{\partial t}\right]_m &+ V_m(t)\left[\frac{V_{m+1}(t) - V_{m-1}(t)}{2h}\right] \\ &= \mu\left[\frac{V_{m+1}(t) - 2V_m(t) + V_{m-1}(t)}{h^2}\right],\end{aligned} \tag{8}$$

or

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

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

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

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

$$p_1(x) = x^2 - 1$$
$$p_2(x) = \frac{1}{3}x^2 + \frac{4}{3}x + \frac{1}{3}.$$

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

$$\begin{aligned}V_m^{n+2} - V_m^n = &\frac{k}{3}\Bigg\{\frac{\mu}{h^2}\left[V_{m+1}^{n+2} - 2V_m^{n+2} + V_{m-1}^{n+2}\right] - \frac{V_m^{n+2}}{2h} \\ &\left[V_{m+1}^{n+2} - V_{m-1}^{n+2}\right] + 4\Bigg(\frac{\mu}{h^2}\left[V_{m+1}^{n+1} - 2V_m^{n+1}\right. \\ &\left. + V_{m-1}^{n+1}\right] - \frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right]\Bigg) + \frac{\mu}{h^2} \\ &\left[V_{m+1}^n - 2V_m^n + V_{m-1}^n\right] - \frac{V_m^n}{2h}\left[V_{m+1}^n\right. \\ &\left. - V_{m-1}^n\right]\Bigg\}.\end{aligned} \tag{11}$$

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

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

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

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

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

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

where

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

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 performed in [3] is

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

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

$$\begin{aligned} &F_m(V_1^{n+1}, V_2^{n+1}, V_3^{n+1}, ..., V_{L-1}^{n+1}) \\ &= V_m^{n+1} - V_m^n - \frac{k}{2}\left\{\frac{\mu}{h^2}\left[V_{m+1}^n - 2V_m^n + V_{m-1}^n\right] - \frac{V_m^n}{2h}\left[V_{m+1}^n - V_{m-1}^n\right]\right\} - \frac{k}{2}\left(\frac{\mu}{h^2}\left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1}\right] - \left[\frac{V_m^{n+1}}{2h} + \frac{k}{2h}\left\{\frac{\mu}{h^2}\left[V_{m+1}^{n+1} - 2V_m^{n+1} + V_{m-1}^{n+1}\right] - \frac{V_m^{n+1}}{2h}\left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right]\right\}\right]\left[V_{m+1}^{n+1} - V_{m-1}^{n+1}\right]\right). \end{aligned} \tag{17}$$

Thus

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

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],

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

with the boundary conditions

$$\left.\begin{matrix} v(0,t) = 0 \\ v(1,t) = 0 \end{matrix}\right\} \quad t > 0 \tag{21}$$

and initial condition

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

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 $x = 0.85$ and 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				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

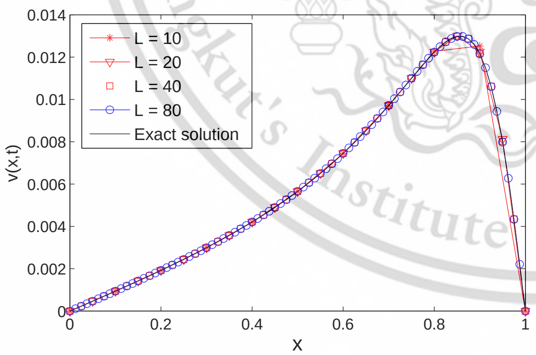


Fig. 1. Exact and numerical solutions at different values of L when $a = 1.1, T = 1$, and $\mu = 0.001$.

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

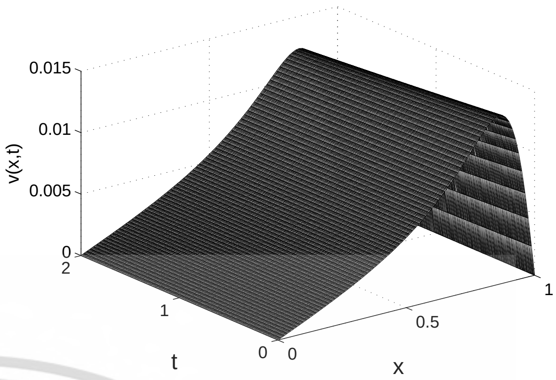


Fig. 2. The surface plot of the numerical solution with $a = 1.1, \mu = 0.001, \Delta t = 0.01$ and $h = 0.0125$.

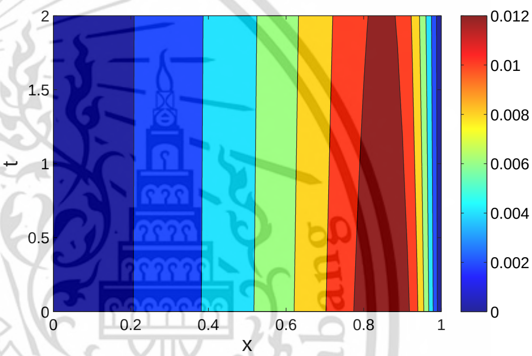


Fig. 3. The contour plot of Fig. 2 when $a = 1.1, \mu = 0.001, \Delta t = 0.01$ and $h = 0.0125$.

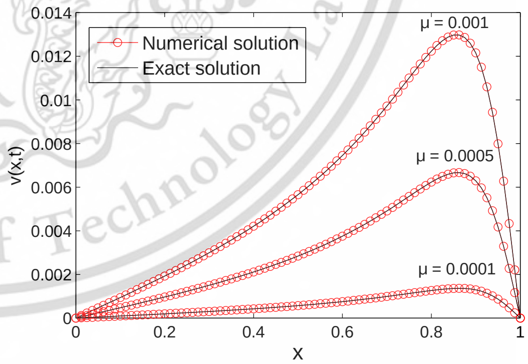


Fig. 4. Numerical solutions for different values of μ with $a = 1.1, T = 1, \Delta t = 0.01$ and $h = 0.0125$.

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

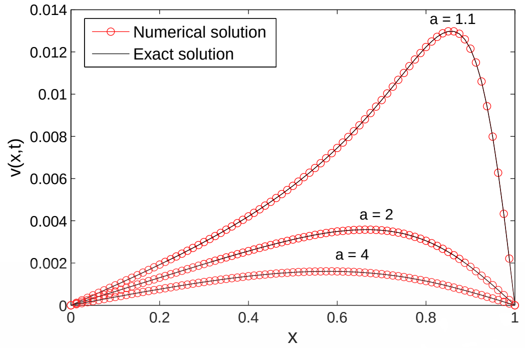


Fig. 5. Numerical solutions for different values of a with $T = 1, \mu = 0.001, \Delta t = 0.01$ and $h = 0.0125$.

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 IV
COMPARISON OF THE L_2-NORM ERRORS OF OUR NUMERICAL SOLUTIONS 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

[1] 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.

Author Biography

Name	Miss Supranee Chonladed
Date of Birth	22 October 1996
Address	24/130 moo 5 Pet Kasem Road, Yaicha, Sam Phran, Nakhon Pathom, 73110
Education	(2018) Bachelor of Science in Applied Mathematics GPA 3.21 King Mongkut's Institute of Technology Ladkrabang
Scholarship	Scholarships for Graduate Student, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang.

Academic Publication(s)

1. Supranee Chonladed and Kanognudge Wuttanachamsri, "A Numerical Solution of Burger's Equation Based on Milne Method," *IAENG International Journal of Applied Mathematics, vol. 51, no. 2, pp. 411 - 415, 2021.*