APPLICATION OF IMPLICIT ADAMS-MOULTON METHOD TO BURGERS EQUATION

APICHAYA CHAIYASIT

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

DEPARTMENT OF APPLIED MATHEMATICS SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2021

KMITL-2021-SC-M-001-011

Thesis Title	Application of Implicit Adams-Moulton Method to Burgers Equation
Student Name	Miss Apichaya Chaiyasit
Student ID	62605014
Degree	Master Degree of Science in Applied Mathematics
Department	Mathematics
Year	2021
Thesis Advisor	Asst. Prof. Dr. Kanognudge Wuttanachamsri

Abstract

A new scheme for finding the numerical solutions of the one-dimensional Burgers equation is provided. A combination of the finite difference method and the implicit Adams-Moulton method is employed to calculate the solution of the nonlinear equation. The nonlinear term is linearized by using a weighted technique. Because the implicit Adams-Moulton method is used in this study, we have to know two initial conditions. The first initial condition is known from the given model while the second initial condition can be calculated from the first one. In this work, the second one are obtained from three different methods: an explicit method, an implicit method, and the Crank-Nicolson method. Numerical results are obtained and compared with the exact solution. The numerical solutions converge to the exact solution when the number of grid points increases. The L_2-norm errors of the numerical results for different values of parameters are provided to show the direction of the solutions when the parameters are changed. Applications include fluid flow such as traffic flow.

Keywords : Burgers equation, Finite Difference method, Implicit method, Adams-Moulton method

Acknowledgements

This thesis is the second-accurate numerical solution of the Berger equation. The makers must thank you Asst. Prof. Dr. Kanognudge Wuttanachamsri, a thesis advisor is very kind to give advice and be a consultant in solving various problems as well as checking the validity of this thesis.

In addition, the authors would like to thank Asst. Prof. Dr. Jiraphan Suntornchost and Dr. Wannaporn Sanprasert for their kindness and sacrifice in their time to be the directors and experts. It also provides knowledge and advice to make this thesis more accurate and complete. We would like to thank all the teachers for their theoretical and practical knowledge and thank you to the staff department of applied mathematics for facilitating the use of the computer room service in making this thesis very well.

Finally, the authors would like to thank your parents and who has encouraged throughout the successful completion of this thesis and including all friends who have contributed to help in various areas in the making of this thesis.

Miss Apichaya Chaiyasit

Table of Contents

List of Tables

List of Figures

Symbols

x	Spatial coordinate.
t	Time coordinate.
υ	Kinematic viscosity.
$u = u(x,t)$	Speed of the fluid at spatial coordinate and time coordinate.
T	The final time.
N	The number of time steps.
Δt	The difference of the current and the next time.
(A,B)	Interval when A and B are margin.
M	The number of spacing step.
Δx	The difference between the current and the next spatial coordinate.
f_1, f_2 and g	Functions for constrains.

Chapter 1

Introduction

1.1 Research Motivation

The Burgers equation have been was applied in various areas in applied mathematics and fluid flow problems such as meteorological phenomena (rain, wind, floods, etc.), processes in the human body (blood flow, breathing, drinking, etc.) and Computational Fluid Dynamics (CFD) applications (aerodynamic shape design, high-speed train simulations, etc.) [12]. It is similar to the Navier-Stokes equation due to the combination of convection, diffusion, and time-dependent terms [14].

In this thesis, we apply the finite difference method, the implicit Adams-Moulton method to the Burgers equation to find the numerical solution. The central finite difference method is used to discretize the spatial terms of the Burgers equation and the Adams-Moulton method is applied to the time derivative term, which will find the previous time derivative term of the Adams-Moulton method can be found from an explicit method, an implicit method, or the Crank-Nicolson method. The nonlinear term is solved by using a linear extrapolating.

1.2 Objectives of the Study

1) To study the Burgers equation.
2) To use multi-step Adams-Moulton method to find the numerical solutions of the equation.
3) To study the effect of the an explicit method, an implicit method, and the Crank-Nicolson method methods to the solutions of the Burgers equation.

1.3 Scopes of the Study

Find the solution of the Burgers equation using the implicit Adams-Moulton method and finite difference scheme.

1.4 Research Methodology

1) Learn basic knowledge of how to find the solutions of the Burgers equation.

2) Study how to solve the nonlinear term of the Burgers equation.

3) Study the steps of how to find the numerical solutions.

4) Write a computer program to solve the equation.

5) Find and verify the solutions of the Burgers equation.

6) Conclude and discuss about the numerical solutions.

7) Write the thesis.

8) Oral presentation.

Table 1.1. Time frame of research.

Activity	Time frame (2020-2021)											
	Jan-Mar	April-May	June-August	Sep-Oct	Nov	Dec	Jan	Feb	Mar	April	May	June
Step 1												
Step 2												
Step 3												
Step 4												
Step 5												
Step 6												
Step 7												
Step 8												

1.5 Benefits of the Study

1) Obtain a proper solution of the nonlinear Burgers equation.

2) Have the knowledge about the Adams-Moulton method.

3) Become a professional of a computer program writing.

Chapter 2

Basis Knowledge and literature Reviews

In this chapter, we provide the basic knowledge of differential equations, the Burgers equation and essential conditions, time division and grid division, the finite difference method, the implicit Adams-Moulton method, an explicit method, an implicit method, the Crank-Nicolson method and literature reviews.

2.1 Differential equations

Differential equations are the equations which show the relationship between the functions of the independent variable, the dependent variable and the derivative of the dependent variable relative to the independent variable. If the dependent variable is a function of just one independent variable, it is called the ordinary differential equation and if the dependent variable is a function of more than one independent variable, it is called the partial differential equation. We illustrate the examples of the ordinary differential equations and the partial differential equation as follows.

- Examples of the ordinary differential equations (ODE)

$$\left(\frac{d^5y}{dx^5}\right)^2 - y\frac{dy}{dx} = y^5$$

$$\frac{d^3y}{dx^3} - x^2\frac{d^2y}{dx^2} + e^y\frac{dy}{dx} = 1$$

- Examples of the partial differential equations (PDE)

$$\frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 u}{\partial y^2} = 2u + z^2 y$$

$$\frac{\partial^2 u}{\partial z^2} + \left(\frac{\partial u}{\partial y}\right)^2 = 2\frac{\partial^2 u}{\partial t^2}$$

<u>**Definition 1**</u> The **order** of a differential equation is the highest derivative that appears in the equation and the **degree** of the equation is the exponent of the highest derivative term.

Definition 2 A differential equation is a **linear equation** if

1) the dependent variable and its derivative have a power indicator of one,
2) there is no term of a product of the dependent variable and the derivative of the dependent variable, and
3) there is no term in the transcendental function of the dependent variable and the derivative of the dependent variable.

A **nonlinear differential equation** is a differential equation which is not a linear equation.

Examples of the order, the degree, the linear equation and the nonlinear equation are shown in Table 2.1.

Table 2.1. Examples of the differential equations.

Equation	Order	Degree	Linear	Nonlinear
1) $\frac{\partial^2 u}{\partial z^2}+\left(\frac{\partial^2 u}{\partial y^2}\right)^2=\frac{\partial u}{\partial t}$	2	2	-	/
2) $\frac{dy}{dx}=y$	1	1	/	-
3) $\left(\frac{\partial y}{\partial x}\right)^3+\frac{\partial^2 y}{\partial z^2}=y^5$	1	3	-	/
4) $\frac{\partial^3 y}{\partial x^3}+\frac{\partial^2 y}{\partial z^2}=10-\frac{\partial y}{\partial t}$	3	1	/	-
5) $(xz')^3=z$	1	3	-	/

2.2 Burgers Equation and Essential Conditions

We provide the one-dimensional Burgers equation with both initial and boundary conditions. The Burgers equation on domain (A,B) [6] is

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\upsilon\frac{\partial^2 u}{\partial x^2}, \quad A<x<B,\ t\in[0,T], \tag{2.1}$$

where T is the final time, with an initial condition

$$u(x,0)=g(x), \qquad A\le x\le B \tag{2.2}$$

and boundary conditions

$$u(A,t)=f_1(t),\ u(B,t)=f_2(t), \quad t\in[0,T], \tag{2.3}$$

where the variable υ is a kinematic viscosity and the functions f_1, f_2, and g are constrains.

2.3 Time and Grid Divisions

2.3.1 Time Division

We divide the interval $[0,T]$ into N steps where $0 \le t_0 \le t_1 \le \ldots \le t_N \le T$, with time step $\Delta t = T/N$ and $t_n = n * \Delta t$, $n = 1,2,\ldots,N$ where T is the final time and N is the number of time steps. The time division is shown in Figure 2.1.

Figure 2.1. Time division

2.3.2 Grids Division

We divide the interval $[A,B]$ into M steps where $A \le x_0 \le x_1 \le \ldots \le x_M \le B$, with spacing step $\Delta x = (B-A)/M$ and $x_m = m * \Delta x$ when $m = 1,2,\ldots,M$ where A and B are the boundary and M is the number of spacing steps. The grids division is shown in Figure 2.2.

Figure 2.2. Grids division

Figure 2.3 illustrates the time and grids divisions. The t-axis represents time coordinate on $\left[t^0, t^N\right]$, $\Delta t = t^{n+1} - t^n$, $n = 1,2,3,\ldots,N$, and the x-axis represents the spatial coordinate on (x_0, x_M), $\Delta x = x_{m+1} - x_m$, $m = 1,2,3,\ldots,M$.

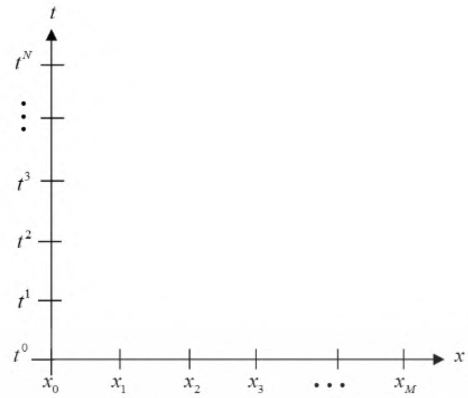


Figure 2.3. The combination of time and grids divisions

2.4 The Finite Difference Method

In this section, we introduce a numerical method called the finite difference method (FDM) used to find the solutions of the partial differential equations (PDEs). We first provide a big-O definition and Taylor series used to estimate the derivatives in the PDEs [5].

<u>Definition 3</u> Let $f(x)$ and $g(x)$ be two functions. Then $f(x)=O(g(x))$ as $x\to a$ if and only if, $\lim_{x\to a}\left|\frac{f(x)}{g(x)}\right|=q\in[0,\infty)$.

<u>Example 1</u> Use Definition 3 to show that $f(x)=O(g(x))$ when $f(x)=x^3+7$ and $g(x)=x^3$

From Definition 3, $\lim_{x\to a}\left|\frac{x^3+7}{x^3}\right|=1\in[0,\infty)$, for any constant $a\in\mathbb{R}-\{0\}$.

Then, $x^3+7=O(x^3)$.

<u>Definition 4</u> Taylor series

Let f be an $n+1$ times differentiable function on an open interval containing the points a and x. Then [4]

$$f(x)=f(a)+f'(a)(x-a)+\frac{f''(a)}{2!}(x-a)^2+\ldots+\frac{f^{(n)}(a)}{n!}(x-a)^n+R_n(x),\quad(2.4)$$

where $R_n(x)=\frac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1}$ for some number c between a and x.

2.4.1 Estimation of Derivatives

From (2.4), we replace f, x, and a by u, $x+h$, and x, respectively.

$$u(x+h,t)=u(x,t)+h\frac{\partial u}{\partial x}+\frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2}+\frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}+\ldots+\frac{h^n}{n!}\frac{\partial^n u}{\partial x^n}+R_n(x),\quad(2.5)$$

$$u(x-h,t)=u(x,t)-h\frac{\partial u}{\partial x}+\frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2}-\frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}+\ldots-\frac{h^n}{n!}\frac{\partial^n u}{\partial x^n}+R_n(x), \quad (2.6)$$

where $h=\Delta x$. We approximate <u>the first order derivatives</u> as follows. From (2.5), we have

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

Thus

$$\frac{\partial u}{\partial x}=\frac{u(x+h,t)-u(x,t)}{h}+O(h) \quad (2.7)$$

The (2.7) is called *forward finite difference method* with the first-order accuracy, $O(h)$. The big O, $O(h)$ in (2.7) is $\frac{h}{2!}\frac{\partial^2 u}{\partial x^2}+\ldots$. Similarly, using (2.6), we have

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

Therefore

$$\frac{\partial u}{\partial x}=\frac{u(x,t)-u(x-h,t)}{h}+O(h). \quad (2.8)$$

The (2.8) is called *backward finite difference method* with the first-order accuracy, $O(h)$. From (2.5) and (2.6), we obtain

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

or

$$\frac{\partial u}{\partial x}=\frac{u(x+h,t)-u(x-h,t)}{2h}+O(h^2). \quad (2.9)$$

The (2.9) is called *central finite difference method* with the truncation error $O(h^2)$.

<u>The second order derivatives</u>.

From (2.5) and (2.6) we have,

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

or

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

The (2.10) is called *central finite difference method* with the truncation error $O(h^2)$.

Similarly the formulas for estimating the time derivative are *forward finite difference method :*

$$\frac{\partial u}{\partial t}=\frac{u(x,t+l)-u(x,t)}{l}+O(l), \tag{2.11}$$

backward finite difference method :

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

central finite difference method :

$$\frac{\partial u}{\partial t}=\frac{u(x,t+l)-u(x,t-l)}{2l}+O(l^2), \tag{2.13}$$

central finite difference method :

$$\frac{\partial^2 u}{\partial t^2}=\frac{u(x,t+l)-2u(x,t)+u(x,t-l)}{l^2}+O(l^2). \tag{2.14}$$

where l is Δt.

<u>Example 2</u> Define $u(x,y)=ye^x$. The values of $\frac{\partial u}{\partial x}$ using forward finite difference method, backward finite difference method, and central finite difference method where Δx =0.1 at point $(1.3,1)$ [7] are

- *forward finite difference method*

  $$\frac{\partial u}{\partial x}=\frac{u(1.3+\Delta x,1)-u(1.3,1)}{h}=\frac{u(1.4,1)-u(1.3,1)}{0.1}=\frac{e^{1.4}-e^{1.3}}{0.1}=3.8590$$

- *backward finite difference method*

  $$\frac{\partial u}{\partial x}=\frac{u(1.3,1)-u(1.3-\Delta x,1)}{h}=\frac{u(1.3,1)-u(1.2,1)}{0.1}=\frac{e^{1.3}-e^{1.2}}{0.1}=3.4918$$

- *central finite difference method*

  $$\frac{\partial u}{\partial x}=\frac{u(1.3+\Delta x,1)-u(1.3-\Delta x,1)}{2h}=\frac{u(1.4,1)-u(1.2,1)}{0.2}=\frac{e^{1.4}-e^{1.2}}{0.2}=3.6754.$$

2.5 The Implicit Adams-Moulton Method

The Adams–Moulton method is an implicit method to find the solution of the initial value problem. Consider the first-order ODE

$$y'(t)=f(t,y(t)). \tag{2.15}$$

Integrating (2.15) both sides on interval $[t_n,t_{n+1}]$, we have

$$\int_{t_n}^{t_{n+1}} y'(t)dt=\int_{t_n}^{t_{n+1}} f(t,y(t))dt.$$

Since

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} \frac{dy}{dt} dt \,, \tag{2.16}$$

The first order Adams-Moulton or the backward Euler method is [3]

$$y_{n+1} = y_n + hf\left(t_{n+1}, y_{n+1}\right), \tag{2.17}$$

where $h = t_{n+1} - t_n$. For the higher-order Adams-Moulton method, we have the second order Adams-Moulton or the trapezoidal rule is

$$y_{n+1} = y_n + \frac{1}{2}h\Big(f\left(t_{n+1}, y_{n+1}\right) + f\left(t_n, y_n\right)\Big), \tag{2.18}$$

and the third order Adams-Moulton is

$$y_{n+1} = y_n + h\left(\frac{5}{12}f\left(t_{n+1}, y_{n+1}\right) + \frac{2}{3}f\left(t_n, y_n\right) - \frac{1}{12}f\left(t_{n-1}, y_{n-1}\right)\right). \tag{2.19}$$

The implicit Adams-Moulton method also has other orders [3].

<u>Example 3</u> Solve the initial value problems (IVPs) : $y' = y - t^2 + 1,\ 0 \le t \le 2,\ y(0) =$ 0.5 by using the third order Adams-Moulton method with l =0.2. The exact value is given by $y(t) = (t+1)^2 - 0.5e^t$ [2].

The initial conditions are calculated from the exact solution. That is

$$y_0 = y(0) = 0.5$$

$$y_1 = y(0.2) = (0.2+1)^2 - 0.5e^{0.2} = 0.8293.$$

From (2.22), solve for y_2, we have

$$\begin{aligned} y_2 &= 0.8293 + \frac{0.2}{12}\Big(5\big(y_2 - (0.4)^2 + 1\big) + 8\big(0.8293 - (0.2)^2 + 1\big) - (0.5+1)\Big) \\ &= 1.2140. \end{aligned}$$

The exact value of y_2, we get

$$\begin{aligned} y_2 = y(0.4) &= (0.4+1)^2 - 0.5e^{0.4} \\ &= 1.2140. \end{aligned}$$

Therefore, The value of the Adams-Moulton method is close to the Exact value. The process is continued until we obtain y at the desired point.

Before we use the implicit Adams-Moulton method to find the solution of the Burgers equation, we need to know two initial conditions. The first initial condition is known from (2.2). To calculate the second initial condition, we use the three different methods: an explicit method, an implicit method, and the Crank-Nicolson method provided in the next section.

2.6 Methods to find the Second Initial Condition

In this section, we discuss three different methods: an explicit method, an implicit method, the Crank-Nicolson method that are used to find the second initial conditions. The numerical solution presented in Chapter 4 will show the inequality when the different methods are used to find the second initial condition.

2.6.1 An Explicit Method

For an explicit method, we use (2.11) to the term $\frac{\partial u}{\partial t}$. That is

$$\frac{\partial u}{\partial t}=\frac{u(x,t+l)-u(x,t)}{l}, \tag{2.20}$$

where $l=\Delta t$. We apply (2.9) to the term $\frac{\partial u}{\partial x}$ and (2.10) to the term $\frac{\partial^2 u}{\partial x^2}$. Therefore,

$$\frac{\partial u}{\partial x}=\frac{u(x+h,t)-u(x-h,t)}{2h}, \tag{2.21}$$

$$\frac{\partial^2 u}{\partial x^2}=\frac{u(x+h,t)-2u(x,t)+u(x-h,t)}{h^2}, \tag{2.22}$$

where $h=\Delta x$. Substituting (2.20) - (2.22) into (2.1), we have

$$U_m^{n+1}=U_m^n+\frac{\upsilon\Delta t}{(\Delta x)^2}\left(U_{m+1}^n-2U_m^n+U_{m-1}^n\right)-\frac{\Delta t}{2\Delta x}U_m^n\left(U_{m+1}^n-U_{m-1}^n\right) \tag{2.23}$$

where U_m^n is the approximation of u at the point $\left(x_m,t^n\right)$, $m=2,3,\ldots,M-1$. The (2.23) is called an explicit equation.

2.6.2 An Implicit Method

For an implicit method, we apply (2.11) to the term $\frac{\partial u}{\partial t}$. That is

$$\frac{\partial u}{\partial t}=\frac{u(x,t+l)-u(x,t)}{l}, \tag{2.24}$$

where $l=\Delta t$, Applying to (2.9) for the term $\frac{\partial u}{\partial x}$ and (2.10) to the term $\frac{\partial^2 u}{\partial x^2}$, we have

$$\frac{\partial u}{\partial x}=\frac{u(x+h,t)-u(x-h,t)}{2h}, \tag{2.25}$$

$$\frac{\partial^2 u}{\partial x^2}=\frac{u(x+h,t+l)-2u(x,t+l)+u(x-h,t+l)}{h^2}, \tag{2.26}$$

where $h = \Delta x$. Substituting (2.24) - (2.26) into (2.1), we obtain

$$\alpha U_{m+1}^{n+1} + \beta U_{m}^{n+1} + \delta U_{m-1}^{n+1} = \mathrm{I} \tag{2.27}$$

where

$$\alpha = -\frac{\upsilon \Delta t}{(\Delta x)^2},$$

$$\beta = 1 + \frac{2\upsilon \Delta t}{(\Delta x)^2},$$

$$\delta = -\frac{\upsilon \Delta t}{(\Delta x)^2},$$

and $I = U_m^n - \frac{\Delta t}{2\Delta x} U_m^n \left(U_{m+1}^n - U_{m-1}^n \right),$

where U_m^n is the approximation of u at the point (x_m, t^n), $m = 2, 3, \ldots, M-1$. The equation (2.27) is called an implicit equation.

2.6.3 The Crank-Nicolson Method

For the Crank-Nicolson method, we apply (2.11) to the term $\frac{\partial u}{\partial t}$:

$$\frac{\partial u}{\partial t} = \frac{u(x, t+l) - u(x, t)}{l}, \tag{2.28}$$

where $l = \Delta t$ and (2.9) to the term $\frac{\partial u}{\partial x}$. That is

$$\frac{\partial u}{\partial x} = \frac{u(x+h, t) - u(x-h, t)}{2h}, \tag{2.29}$$

For $\frac{\partial^2 u}{\partial x^2}$ in (2.10), we use (2.10) to find the average of $\frac{\partial^2 u}{\partial x^2}$ at time t and $t+l$ as follows.

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{2}\left[\frac{u(x+h, t) - 2u(x, t) + u(x-h, t)}{h^2} + \frac{u(x+h, t+l) - 2u(x, t+l) + u(x-h, t+l)}{h^2} \right], \tag{2.30}$$

where $h = \Delta x$. Substituting (2.28) - (2.30) into (2.1), we have

$$\gamma U_{m+1}^{n+1} + \varphi U_{m}^{n+1} + \lambda U_{m-1}^{n+1} = C \tag{2.31}$$

where

$$\gamma = -\frac{\upsilon \Delta t}{2(\Delta x)^2},$$

$$\varphi = 1 + \frac{\upsilon \Delta t}{(\Delta x)^2},$$

$$\lambda = -\frac{\upsilon \Delta t}{2(\Delta x)^2},$$

$$c = \left(\frac{\upsilon \Delta t}{2(\Delta x)^2} - \frac{\Delta t \, U_m^n}{2\Delta x}\right) U_{m+1}^n + \left(1 - \frac{\upsilon \Delta t}{(\Delta x)^2}\right) U_m^n + \left(\frac{\upsilon \Delta t}{2(\Delta x)^2} + \frac{\Delta t \, U_m^n}{2\Delta x}\right) U_{m-1}^n .$$

where U_m^n is the approximation of u at the point (x_m, t^n), $m = 2,3,\ldots,M-1$. The equation (2.31) is called the Crank-Nicolson equation method. The Crank-Nicolson method different from the explicit method and the implicit method because for the explicit method we use $\frac{\partial^2 u}{\partial x^2}$ at the time t and for the implicit method we uses $\frac{\partial^2 u}{\partial x^2}$ at the time $t+l$, while for the Crank-Nicolson method we find the average of $\frac{\partial^2 u}{\partial x^2}$ at time t and $t+l$.

2.7 Literature Reviews

Researchers have interested in the study of numerical solutions of the Burgers equation especially in the fields of physics and engineering. For example, M. Landajuela [13] studied the flow of cars on a highway by employing the Burgers equation. There were several methods used to find the numerical solution of the Burgers equation such as N.A. Mohamed [14] solving nonlinear one-dimensional Burgers equation by using finite difference method. V. Mukundan and A. Awasthi [15] applied the method of lines (MOL) to solve the Burgers equation. A. Sheikh et al. [1] studied a comparison of the LaxFriedrich and Lax-Wendroffs schemes to find the numerical solution of the Burgers equation. S. Xie et al. [10] used the HopfCole transformation and a reproducing kernel function to calculate the solution of the Burgers equation. P.G. Zhang and J.P. Wang [19] used a predictor-corrector method called MacCormack method to determine the solutions. A. Chaiyasit et al. [7] provided the solution of the Burgers equation by using the finite difference method and the second-order Runge-Kutta method. S. Phairat et al. [18] studied the numerical solution of the Burgers equation by using the sixth-order compact finite difference method and the McCormack method.

Chapter 3

Numerical Method and Procedure

In this chapter, we provide a numerical method and the process to find the numerical solution of the Burgers equation. In section 3.1, we present the central finite difference method to discretize the spatial term of the Burgers equation. In section 3.2 we apply the implicit Adams-Moulton to discretize the time derivative term of the Burgers equation.

3.1 Central Finite Difference Method

We use the central finite difference method to the spatial terms of the Burgers equation. Therefore, the Burgers equation (2.1), becomes

$$\left[\frac{\partial u}{\partial t}\right]_m + U_m(t)\left[\frac{U_{m+1}(t)-U_{m-1}(t)}{2h}\right] = \upsilon\left[\frac{U_{m+1}(t)-2U_m(t)+U_{m-1}(t)}{h^2}\right], \tag{3.1}$$

or

$$\left[\frac{\partial u}{\partial t}\right]_m = \frac{\upsilon}{h^2}\left[U_{m+1}(t)-2U_m(t)+U_{m-1}(t)\right] - \frac{U_m(t)}{2h}\left[U_{m+1}(t)-U_{m-1}(t)\right], \tag{3.2}$$

where $U_m(t)=u(x_m,t)$, $m=2,3,\ldots,M-1$, $\left[\frac{\partial u}{\partial t}\right]_m$ is $\frac{\partial u}{\partial t}$ at the point $x=x_m$ and $h=\Delta x$.

3.2 Implicit Adams-Moulton Method

We apply the Adams-Moulton method to the time derivative term of (3.2). The third-order implicit Adams-Moulton used to find the solution of the equation $u'=f(t,u(t))$ is [1].

$$u_{n+2}=u_{n+1}+l\left(\frac{5}{12}f\left(t_{n+2},u_{n+2}\right)+\frac{2}{3}f\left(t_{n+1},u_{n+1}\right)-\frac{1}{12}f\left(t_n,u_n\right)\right), \tag{3.3}$$

where $l=\Delta t$, $u_n=u\left(t_n\right)$, $n=0,1,\ldots,N$, and $0\le t_n\le T$. The stability and consistency of (3.3) can be verified from the polynomials [11]

$$k(w)=w^2-w,$$

$$r(w)=\frac{5}{12}w^2+\frac{2}{3}w-\frac{1}{12}.$$

The roots of k are 0 and 1, which are simple roots and in the unit disk. Moreover, $k'(w)=2w-1$ and $k'(1)=1=r(1).$ From Chapter 8 in [11], the Adams-Moulton method is stable and consistence. Applying (3.3) to (3.2), we have

$$\begin{aligned}\mathrm{U}_m^{n+2}=\mathrm{U}_m^{n+1}&+\frac{5l}{12}\left[\frac{\upsilon}{h^2}\left(\mathrm{U}_{m+1}^{n+2}-2\mathrm{U}_m^{n+2}+\mathrm{U}_{m-1}^{n+2}\right)-\frac{\mathrm{U}_m^{n+2}}{2h}\left(\mathrm{U}_{m+1}^{n+2}-\mathrm{U}_{m-1}^{n+2}\right)\right]\\&+\frac{2l}{3}\left[\frac{\upsilon}{h^2}\left(\mathrm{U}_{m+1}^{n+1}-2\mathrm{U}_m^{n+1}+\mathrm{U}_{m-1}^{n+1}\right)-\frac{\mathrm{U}_m^{n+1}}{2h}\left(\mathrm{U}_{m+1}^{n+1}-\mathrm{U}_{m-1}^{n+1}\right)\right]\\&-\frac{l}{12}\left[\frac{\upsilon}{h^2}\left(\mathrm{U}_{m+1}^{n}-2\mathrm{U}_m^{n}+\mathrm{U}_{m-1}^{n}\right)-\frac{\mathrm{U}_m^{n}}{2h}\left(\mathrm{U}_{m+1}^{n}-\mathrm{U}_{m-1}^{n}\right)\right].\end{aligned}\tag{3.4}$$

Since (3.4) consists of a nonlinear term at time t_{n+2},

$$\mathrm{U}_m^{n+2}\left(\mathrm{U}_{m+1}^{n+2}-\mathrm{U}_{m-1}^{n+2}\right),\tag{3.5}$$

we approximate U_m^{n+2} by using a linear extrapolation of U_m^{n+1} and U_m^{n} [14],

$$\mathrm{U}_m^{n+2}\cong\left(1+\left(\frac{j_{n+2}}{j_{n+1}}\right)\right)\mathrm{U}_m^{n+1}-\left(\frac{j_{n+2}}{j_{n+1}}\right)\mathrm{U}_m^{n},\tag{3.6}$$

where $j_{n+2}=t_{n+2}-t_{n+1}$. Then (3.5) becomes

$$\left(\left(1+\left(\frac{j_{n+2}}{j_{n+1}}\right)\right)\mathrm{U}_m^{n+1}-\left(\frac{j_{n+2}}{j_{n+1}}\right)\mathrm{U}_m^{n}\right)\left(\mathrm{U}_{m+1}^{n+2}-\mathrm{U}_{m-1}^{n+2}\right).\tag{3.7}$$

Substituting (3.7) into (3.4), we have

$$\begin{aligned}\mathrm{U}_m^{n+2}=\mathrm{U}_m^{n+1}&+\frac{5\upsilon l}{12h^2}\left[\mathrm{U}_{m+1}^{n+2}-2\mathrm{U}_m^{n+2}+\mathrm{U}_{m-1}^{n+2}\right]\\&-\frac{5l}{24h}\left(\left(1+\left(\frac{j_{n+2}}{j_{n+1}}\right)\right)\mathrm{U}_m^{n+1}-\left(\frac{j_{n+2}}{j_{n+1}}\right)\mathrm{U}_m^{n}\right)\left[\mathrm{U}_{m+1}^{n+2}-\mathrm{U}_{m-1}^{n+2}\right]\\&+\frac{2\upsilon l}{3h^2}\left[\mathrm{U}_{m+1}^{n+1}-2\mathrm{U}_m^{n+1}+\mathrm{U}_{m-1}^{n+1}\right]-\frac{l}{3h}\mathrm{U}_m^{n+1}\left[\mathrm{U}_{m+1}^{n+1}-\mathrm{U}_{m-1}^{n+1}\right]\\&-\frac{\upsilon l}{12h^2}\left[\mathrm{U}_{m+1}^{n}-2\mathrm{U}_m^{n}+\mathrm{U}_{m-1}^{n}\right]+\frac{l}{24h}\mathrm{U}_m^{n}\left[\mathrm{U}_{m+1}^{n}-\mathrm{U}_{m-1}^{n}\right].\end{aligned}\tag{3.8}$$

In this work, we use equally spaced in time. Then $j_{n+1}=\Delta t$, for all n. Multiplying (3.8) by $24h^2$ yields

$$a_m\mathrm{U}_{m+1}^{n+2}+b_m\mathrm{U}_m^{n+2}+c_m\mathrm{U}_{m-1}^{n+2}=f_m,\tag{3.9}$$

where

$$a_m=10hl\mathrm{U}_m^{n+1}-5hl\mathrm{U}_m^{n}-10\upsilon l,$$

$$b_m=24h^2+20\upsilon l,$$

$$\begin{aligned} c_m &= 5hl\bigcup_m^n - 10hl\bigcup_m^{n+1} - 10\upsilon l, \\ f_m &= \left(16\upsilon l - 8hl\bigcup_m^{n+1}\right)\bigcup_{m+1}^{n+1} + \left(24h^2 - 32\upsilon l\right)\bigcup_m^{n+1} \\ &\quad + \left(16\upsilon l + 8hl\bigcup_m^{n+1}\right)\bigcup_{m-1}^{n+1} + \left(hl\bigcup_m^n - 2\upsilon l\right)\bigcup_{m+1}^n \\ &\quad + 4\upsilon l\bigcup_m^n - \left(2\upsilon l + hl\bigcup_m^n\right)\bigcup_{m-1}^n, \end{aligned}$$

where $m = 2,3,\ldots,M-1$. From (3.9), to find $\bigcup_m^{n+2}$, it is necessary to know $\bigcup_m^n$ and $\bigcup_m^{n+1}$, $m = 1,2,\ldots,M-1$. However, $\bigcup_m^n$ can be obtained from the initial condition. To calculate the value $\bigcup_m^{n+1}$, we apply the finite different method again to (2.1) by using an explicit equation, an implicit equation, and the Crank-Nicolson equation. Now we have $\bigcup_m^n$ and $\bigcup_m^{n+1}$. Then we are really to find the numerical solutions.

Chapter 4

Numerical results

This chapter provides the one-dimensional exact solution of the Burgers equation and the numerical results obtained from the methods provided in chapter 3.

4.1 Exact Solution

In this section we provide the exact solution, which is used to verify the numerical result. The exact result of Burgers equation is [20]

$$u(x,t)=\frac{2\upsilon\pi e^{-\pi^2\upsilon t}\sin(\pi x)}{p+e^{-\pi^2\upsilon t}\cos(\pi x)}, \quad (4.1)$$

where the initial condition,

$$u(x,0)=\frac{2\upsilon\pi\sin(\pi x)}{p+\cos(\pi x)},\ \ 0\le x\le 1, \quad (4.2)$$

and the boundary conditions,

$$u(0,t)=u(1,t)=0,\quad t>0, \quad (4.3)$$

where p is a constant.

Since we use Adams-Moulton implicit method to find the numerical solutions, two initial conditions are required to calculate the result. However, now we have only one initial condition. To calculate the second initial condition we use three different methods : the explicit method, the implicit method, and the Crank-Nicolson method as provided in Section 2.6. The numerical solutions of each method combining with the Adams-Moulton scheme are provided in Sections 4.2-4.4. From (3.9), the initial U_m^0 is obtained from the initial condition and U_m^1 is calculated from (2.23), (2.27) or (2.31). Substituting U_m^0 and U_m^1 into (3.9), we have U_m^2. Then U_m^3 can be determined from U_m^1 and U_m^2. Continuing this process, we have U_m^n, $1\le m\le L$, for all n. Therefore, the numerical results are illustrated as follows.

4.2 An Explicit method

This section shows the numerical results obtained from (3.9) and (2.23) as well as the L_2-norm error of the solutions.

Figure 4.1 shows the numerical solutions U_m^n, $1 \le m \le L$, for all n at (i) t = 0.01 and (ii) t = 1 with the initial condition (4.2) and the boundary conditions (4.3). The values of variables are Δt =0.005, Δx =0.05, 0.02, 0.0125, and 0.001 or L =20, 50, 80, and 100, respectively, υ =0.01, p =2, A =0, B =1, and T =1. At t = 0.01, showing the numerical solution using the explicit method scheme, it shows the numerical results converge to the exact solution. It means the solution obtained from (2.23) are accurated so that it can be used to find the solution for the next time step. Then the initial condition and the solution at t = 0.01 are placed in (3.9) to find the numerical solution for the next time step. We continue this process until t = 1.

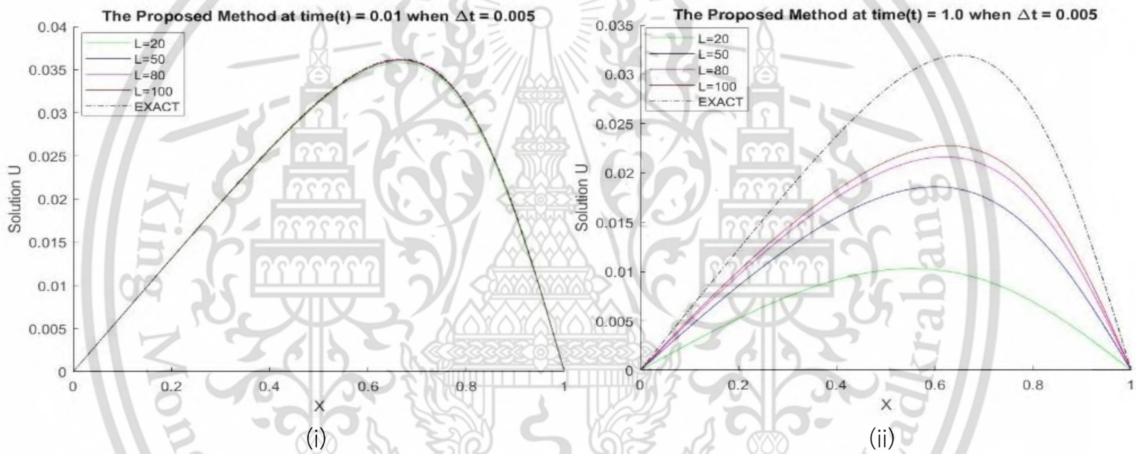


Figure 4.1. Exact solution and Numerical solutions at (i) t = 0.01 and (ii) t = 1.0.

From Figure 4.1, The numerical results at T = 1.0 converge to the exact solution when the number of grid points increases. The results at each point x and each L are presented in Tables 4.1 and 4.2. The L_2-norm errors, $L_2 = \|u_{exact} - u_{solutions}\|_2$, of each L are provided in the last row of the tables. Notice that the errors decrease when L increases. The smallest errors are occurred when L =100 for both t = 0.01 and t = 1.0.

Table 4.1. Exact solutions and the numerical solutions at each point x at time t = 0.01.

x	Exact solutions	Numerical Solutions			
		$L=20$	$L=50$	$L=80$	$L=100$
0.1	0.00657	0.00655	0.00657	0.00657	0.00657
0.2	0.01314	0.01308	0.01312	0.01312	0.01313
0.3	0.01963	0.01954	0.01959	0.01960	0.01960
0.4	0.02586	0.02572	0.02579	0.02581	0.02582
0.5	0.03138	0.03120	0.03130	0.03132	0.03133
0.6	0.03530	0.03506	0.03518	0.03521	0.03522
0.7	0.03594	0.03567	0.03582	0.03585	0.03587
0.8	0.03096	0.03069	0.03084	0.03088	0.03089
0.9	0.01848	0.01829	0.01841	0.01844	0.01844
L_2-norm		7.7463e-04	5.5379e-04	4.9057e-04	4.7098e-04

Table 4.2. Exact solutions and the numerical solutions at each point x at time t = 1.0.

x	Exact solutions	Numerical Solutions			
		$L=20$	$L=50$	$L=80$	$L=100$
0.1	0.00615	0.00271	0.00438	0.00494	0.00515
0.2	0.01224	0.00525	0.00855	0.00963	0.01003
0.3	0.01819	0.00747	0.01232	0.01389	0.01445
0.4	0.02375	0.00918	0.01547	0.01751	0.01823
0.5	0.02846	0.01016	0.01771	0.02023	0.02113
0.6	0.03148	0.01021	0.01860	0.02158	0.02266
0.7	0.03138	0.00916	0.01757	0.02084	0.02207
0.8	0.02641	0.00696	0.01405	0.01710	0.01832
0.9	0.01545	0.00376	0.00791	0.00987	0.01069
L_2-norm		6.6091e-02	6.3303e-02	6.0682e-02	5.9709e-02

Figure 4.2 shows the L_2-norm errors for Δt =0.01, 0.005, and 0.0025 when L =100, p =2, A =0, B =1, T =1, and υ = 0.01. The graphs illustrate that L_2-norm errors decrease for larger Δt. The L_2-norm errors in Figure 4.2 are presented in Table 4.3 The errors when Δt =0.01 is better than that of Δt =0.005 and 0.0025 for each L. This is because of the accumulated error of the increasing number of iterations. The error when L =100 is smallest as expected.

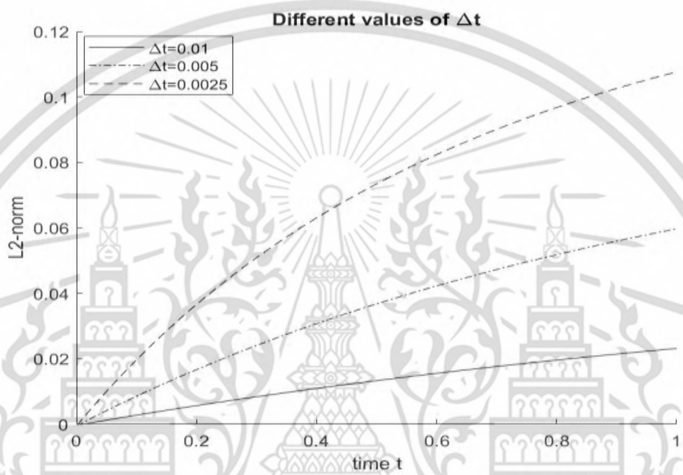


Figure 4.2. L_2-norm errors for different values of Δt.

Table 4.3. L_2-norm errors for different Δt at t = 1.0.

L	L_2-norm errors		
	Δt = 0.01	Δt = 0.005	Δt = 0.0025
20	4.1229e-02	6.6091e-02	8.6869e-02
50	3.0995e-02	6.3303e-02	1.0080e-01
80	2.5542e-02	6.0682e-02	1.0527e-01
100	2.3162e-02	5.9709e-02	1.0763e-01

Figure 4.3 illustrates the L_2-norm errors for different values of coefficient υ when Δx =0.01, p =2, A =0, B =1, T =1, Δt =0.005, and υ = 0.01, 0.005, and 0.001. It shows that when υ increases the L_2-norm errors are increase.

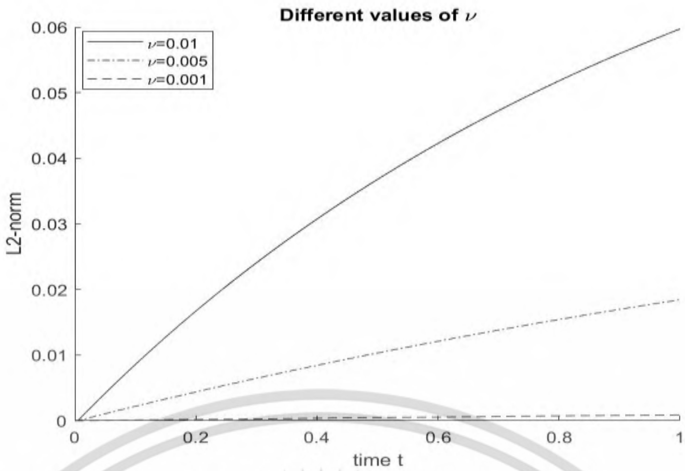
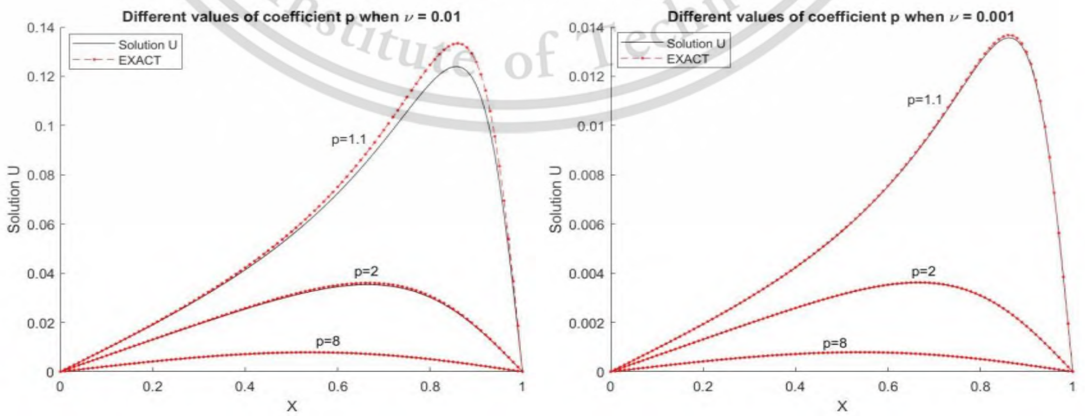


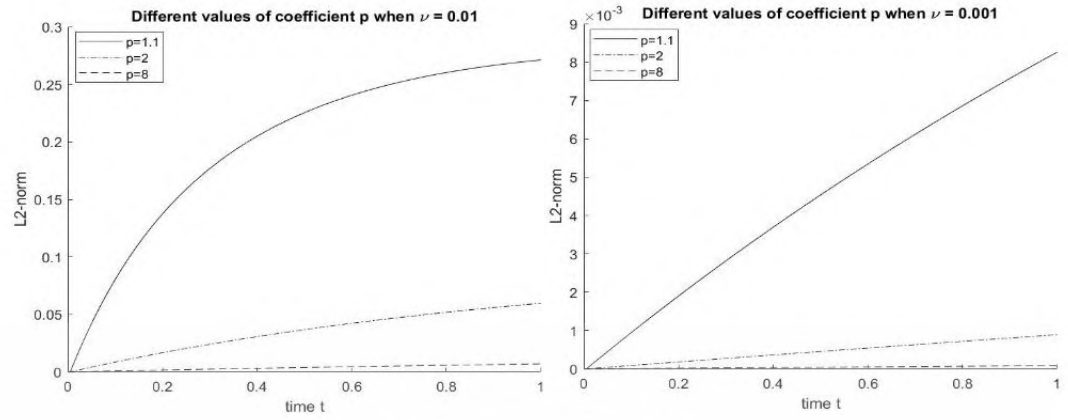
Figure 4.3. L_2-norm errors for different values of the coefficient υ.

Figure 4.4 illustrates the exact and numerical solutions for different values of p which are 1.1, 2 and 8, with (i) υ =0.01 and (ii) υ =0.001 when Δt =0.005, Δx =0.01, A =0, B =1, and T =1.0. The numerical results are closed to the exact solutions for each p. Figure 4.5 shows the L_2- norm errors of the numerical solutions provided in Figure 4.4. The results in Figure 4.5 show that the errors decrease for larger p for both υ =0.01 and 0.001. The L_2- norm errors presented in Figure 4.5 are illustrated in Table 4.4, where we also provide the errors of the numerical results when t =1. Notice that the errors decrease when the coefficient p increases for the different values of υ. Moreover, the errors of υ =0.001 are less than that of υ =0.01 for each p for both t =0.05 and t =1.



(i) (ii)

Figure 4.4. The exact and numerical solutions for different values of coefficient p, (i) υ = 0.01 and (ii) υ = 0.001 at t = 0.05.



(i) (ii)

Figure 4.5. L_2-norm errors for different values of the coefficient p, (i) υ = 0.01 and (ii) υ = 0.001 at t = 0.05.

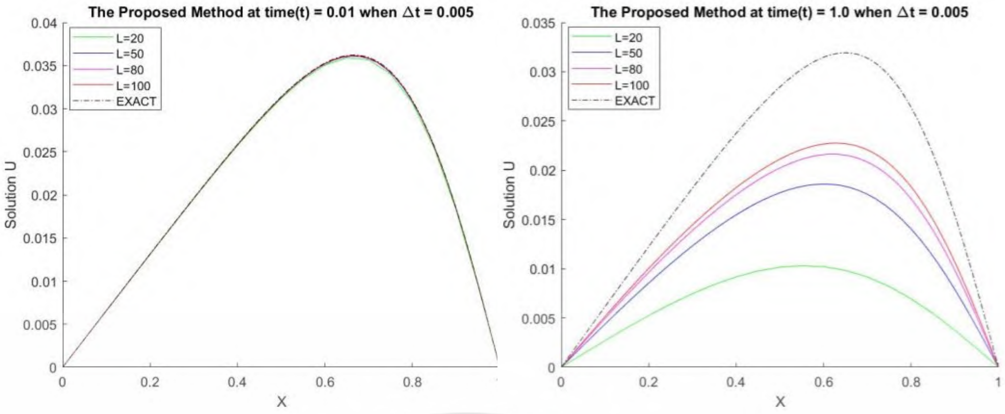
Table 4.4. L_2-norm errors for different values of coefficient p.

p	t = 0.05		t = 1.0	
	υ = 0.01	υ = 0.001	υ = 0.01	υ = 0.001
1.1	5.0472e-03	5.0719e-05	2.7116e-01	8.2592e-03
2	4.7098e-04	4.7139e-06	5.9709e-02	8.9120e-04
8	4.3613e-05	4.3643e-07	6.9800e-03	8.4904e-05

4.3 An Implicit method

This section shows the numerical results obtained from (3.9) and (2.27) as well as the L_2-norm error of the solutions.

Figure 4.6 shows the numerical solutions U_m^n, $1 \le m \le L$, for all n at (i) t = 0.01 and (ii) t = 1 with the initial condition (4.2) and the boundary conditions (4.3). The values of variables are Δt =0.005, Δx =0.05, 0.02, 0.0125, and 0.001 or L =20, 50, 80, and 100, respectively, υ =0.01, p =2, A =0, B =1, and T =1. At t = 0.01, showing the numerical solution using the implicit method scheme, it shows the numerical results converge to the exact solution. It means the solution obtained from (2.27) are accurated so that it can be used to find the solution for the next time step. Then the initial condition and the solution at t = 0.01 are placed in (3.9) to find the numerical solution for the next time step. We continue this process until t = 1.



(i) (ii)

Figure 4.6. Exact solution and Numerical solutions

at (i) t = 0.01 and (ii) t = 1.0.

From Figure 4.6, The numerical results at T = 1.0 converge to the exact solution when the number of grid points increases. The results at each point x and each L are presented in Tables 4.5 and 4.6. The L_2-norm errors, $L_2 = \|u_{exact} - u_{solutions}\|_2$, of each L are provided in the last row of the tables. Notice that the errors decrease when L increases. The smallest errors are occurred when L =100 for both t = 0.01 and t = 1.0.

Table 4.5. Exact solutions and the numerical solutions

at each point x at time t = 0.01.

x	Exact solutions	Numerical Solutions			
		$L=20$	$L=50$	$L=80$	$L=100$
0.1	0.00657	0.00655	0.00657	0.00657	0.00657
0.2	0.01314	0.01308	0.01312	0.01312	0.01313
0.3	0.01963	0.01954	0.01959	0.01960	0.01960
0.4	0.02586	0.02572	0.02579	0.02581	0.02582
0.5	0.03138	0.03120	0.03130	0.03132	0.03133
0.6	0.03530	0.03506	0.03518	0.03521	0.03522
0.7	0.03594	0.03567	0.03582	0.03585	0.03587
0.8	0.03096	0.03069	0.03084	0.03088	0.03089
0.9	0.01848	0.01829	0.01841	0.01844	0.01844
L_2-norm		7.7457e-04	5.5369e-04	4.9046e-04	4.7086e-04

Table 4.6. Exact solutions and the numerical solutions at each point x at time t = 1.0.

x	Exact solutions	Numerical Solutions			
		$L=20$	$L=50$	$L=80$	$L=100$
0.1	0.00615	0.00271	0.00438	0.00494	0.00515
0.2	0.01224	0.00525	0.00855	0.00963	0.01003
0.3	0.01819	0.00747	0.01232	0.01389	0.01445
0.4	0.02375	0.00918	0.01547	0.01751	0.01823
0.5	0.02846	0.01016	0.01771	0.02023	0.02113
0.6	0.03148	0.01021	0.01860	0.02158	0.02266
0.7	0.03138	0.00916	0.01757	0.02084	0.02207
0.8	0.02641	0.00696	0.01405	0.01710	0.01832
0.9	0.01545	0.00376	0.00791	0.00987	0.01069
L_2-norm		6.6091e-02	6.3303e-02	6.0682e-02	5.9709e-02

Figure 4.7 shows the L_2-norm errors for Δt =0.01, 0.005, and 0.0025 when L =100, p =2, A =0, B =1, T =1, and υ = 0.01. The graphs illustrate that L_2-norm errors decrease for larger Δt. The L_2-norm errors in Figure 4.7 are presented in Table 4.7 The errors when Δt =0.01 is better than that of Δt =0.005 and 0.0025 for each L. This is because of the accumulated error of the increasing number of iterations. The error when L =100 is smallest as expected.

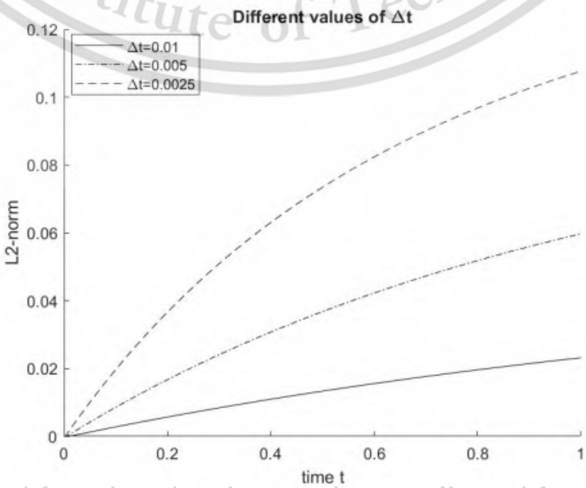


Figure 4.7. L_2-norm errors for different values of Δt.

Table 4.7. L_2 - norm errors for different Δt at t = 1.0.

L	L_2-norm errors		
	Δt = 0.01	Δt = 0.005	Δt = 0.0025
20	4.1229e-02	6.6091e-02	8.6869e-02
50	3.0995e-02	6.3303e-02	1.0080e-01
80	2.5542e-02	6.0682e-02	1.0527e-01
100	2.3162e-02	5.9709e-02	1.0763e-01

Figure 4.8 illustrates the L_2-norm errors for different values of coefficient υ when Δx =0.01, p =2, A =0, B =1, T =1, Δt =0.005, and υ = 0.01, 0.005, and 0.001. It shows that when υ increases the L_2-norm errors are increase.

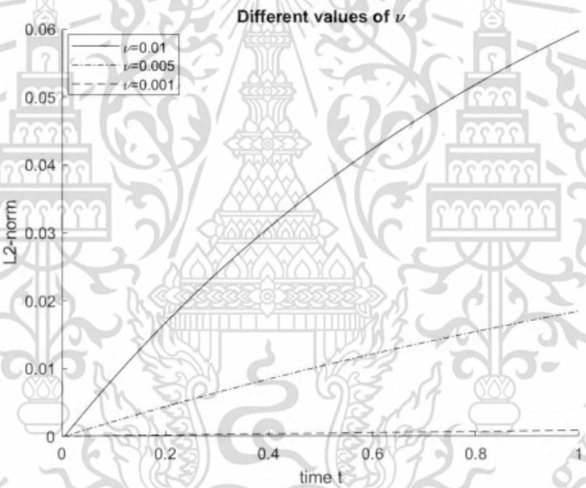
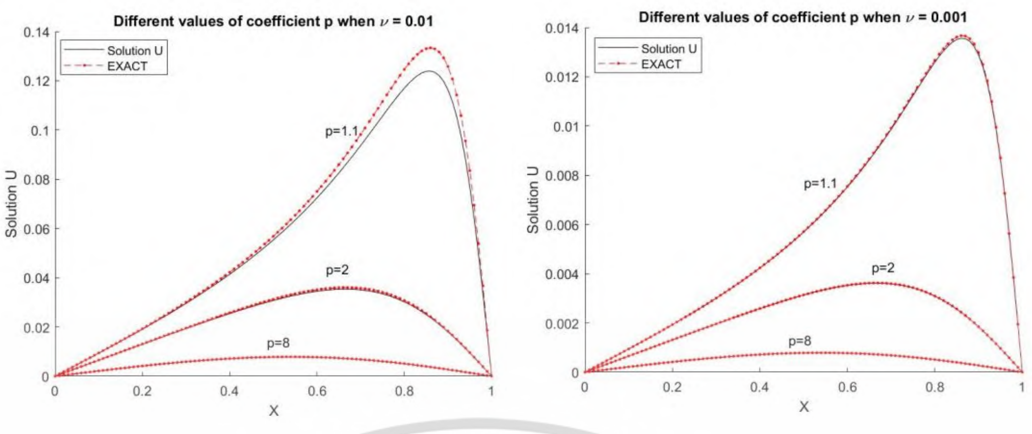


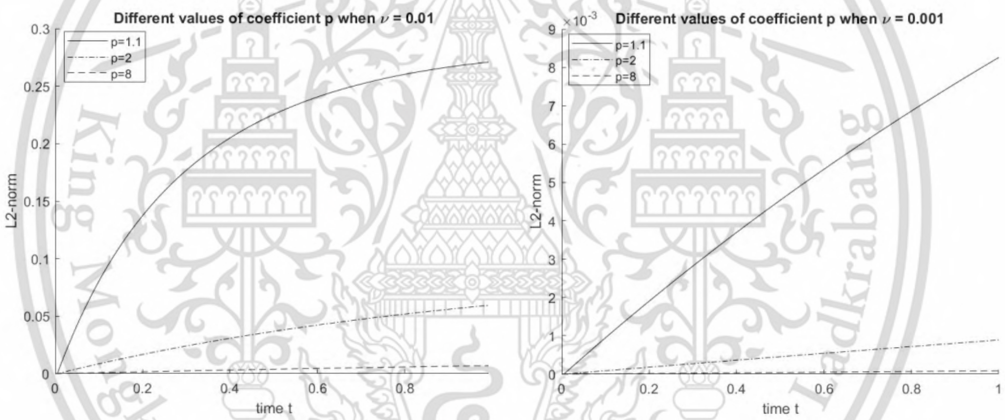
Figure 4.8. L_2-norm errors for different values of the coefficient υ.

Figure 4.9 illustrates the exact and numerical solutions for different values of p which are 1.1, 2, and 8, with (i) υ =0.01 and (ii) υ =0.001 when Δt =0.005, Δx =0.01, A =0, B =1, and T =1.0. The numerical results are closed to the exact solutions for each p. Figure 4.10 shows the L_2- norm errors of the numerical solutions provided in Figure 4.9. The results in Figure 4.10 show that the errors decrease for larger p for both υ =0.01 and 0.001. The L_2- norm errors presented in Figure 4.10 are illustrated in Table 4.8, where we also provide the errors of the numerical results when t =1. Notice that the errors decrease when the coefficient p increases for the different values of υ. Moreover, the errors of υ =0.001 are less than that of υ =0.01 for each p for both t =0.05 and t =1.



(i) (ii)

Figure 4.9. The exact and numerical solutions for different values of coefficient p, (i) υ = 0.01 and (ii) υ =0.001 at t = 0.05.



(i) (ii)

Figure 4.10. L_2-norm errors for different values of the coefficient p, (i) υ = 0.01 and (ii) υ =0.001 at t = 0.05.

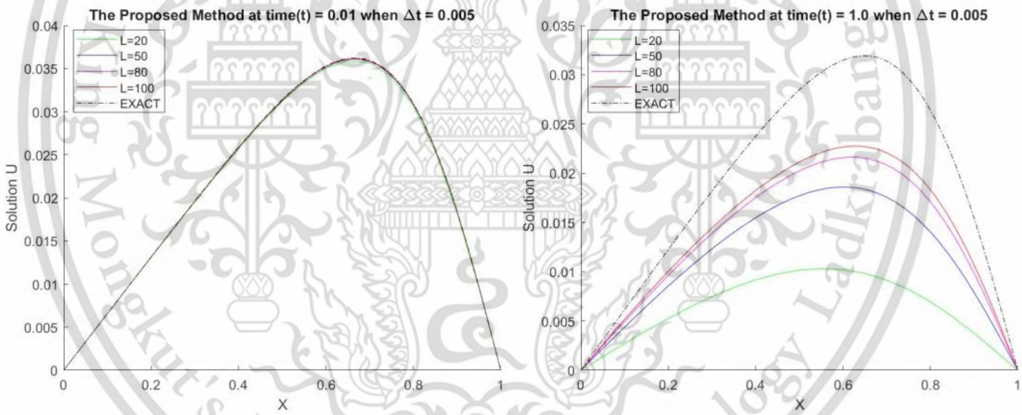
Table 4.8. L_2-norm errors for different values of coefficient p.

p	t = 0.05		t = 1.0	
	υ = 0.01	υ = 0.001	υ = 0.01	υ = 0.001
1.1	5.0394e-03	5.0711e-05	2.7116e-01	8.2592e-03
2	4.7086e-04	4.7137e-06	5.9709e-02	8.9120e-04
8	4.3599e-05	4.3642e-07	6.9800e-03	8.4904e-05

4.4 The Crank-Nicolson method

This section shows the numerical results obtained from (3.9) and (2.31) as well as the L_2-norm error of the solutions.

Figure 4.11 shows the numerical solutions U_m^n, $1 \le m \le L$, for all n at (i) t = 0.01 and (ii) t = 1 with the initial condition (4.2) and the boundary conditions (4.3). The values of variables are Δt =0.005, Δx =0.05, 0.02, 0.0125, and 0.001 or L =20, 50, 80, and 100, respectively, υ =0.01, p =2, A =0, B =1, and T =1. At t = 0.01, showing the numerical solution using the Crank-Nicolson method scheme, it shows the numerical results converge to the exact solution. It means the solution obtained from (2.31) are accurated so that it can be used to find the solution for the next time step. Then the initial condition and the solution at t = 0.01 are placed in (3.9) to find the numerical solution for the next time step. We continue this process until t = 1.



(i) (ii)

Figure 4.11. Exact solution and Numerical solutions at (i) t = 0.01 and (ii) t = 1.0.

From Figure 4.11, The numerical results at T = 1.0 converge to the exact solution when the number of grid points increases. The results at each point x and each L are presented in Tables 4.9 and 4.10. The L_2 -norm errors, $L_2 = \|u_{exact} - u_{solutions}\|_2$, of each L are provided in the last row of the tables. Notice that the errors decrease when L increases. The smallest errors are occurred when L =100 for both t = 0.01 and t = 1.0.

Table 4.9. Exact solutions and the numerical solutions at each point x at time t = 0.01.

x	Exact solutions	Numerical Solutions			
		$L=20$	$L=50$	$L=80$	$L=100$
0.1	0.00657	0.00655	0.00657	0.00657	0.00657
0.2	0.01314	0.01308	0.01312	0.01312	0.01313
0.3	0.01963	0.01954	0.01959	0.01960	0.01960
0.4	0.02586	0.02572	0.02579	0.02581	0.02582
0.5	0.03138	0.03120	0.03130	0.03132	0.03133
0.6	0.03530	0.03506	0.03518	0.03521	0.03522
0.7	0.03594	0.03567	0.03582	0.03585	0.03587
0.8	0.03096	0.03069	0.03084	0.03088	0.03089
0.9	0.01848	0.01829	0.01841	0.01844	0.01844
L_2-norm		7.7460e-04	5.5374e-04	4.9052e-04	4.7092e-04

Table 4.10. Exact solutions and the numerical solutions at each point x at time t = 1.0.

x	Exact solutions	Numerical Solutions			
		$L=20$	$L=50$	$L=80$	$L=100$
0.1	0.00615	0.00271	0.00438	0.00494	0.00515
0.2	0.01224	0.00525	0.00855	0.00963	0.01003
0.3	0.01819	0.00747	0.01232	0.01389	0.01445
0.4	0.02375	0.00918	0.01547	0.01751	0.01823
0.5	0.02846	0.01016	0.01771	0.02023	0.02113
0.6	0.03148	0.01021	0.01860	0.02158	0.02266
0.7	0.03138	0.00916	0.01757	0.02084	0.02207
0.8	0.02641	0.00696	0.01405	0.01710	0.01832
0.9	0.01545	0.00376	0.00791	0.00987	0.01069
L_2-norm		6.6091e-02	6.3303e-02	6.0682e-02	5.9709e-02

Figure 4.13 shows the L_2-norm errors for Δt =0.01, 0.005, and 0.0025 when L =100, p =2, A =0, B =1, T =1, and υ = 0.01. The graphs illustrate that L_2-norm errors decrease for larger Δt. The L_2-norm errors in Figure 4.13 are presented in Table 4.12 The errors when Δt =0.01 is better than that of Δt =0.005 and 0.0025 for each L. This is because of the accumulated error of the increasing number of iterations. The error when L =100 is smallest as expected.

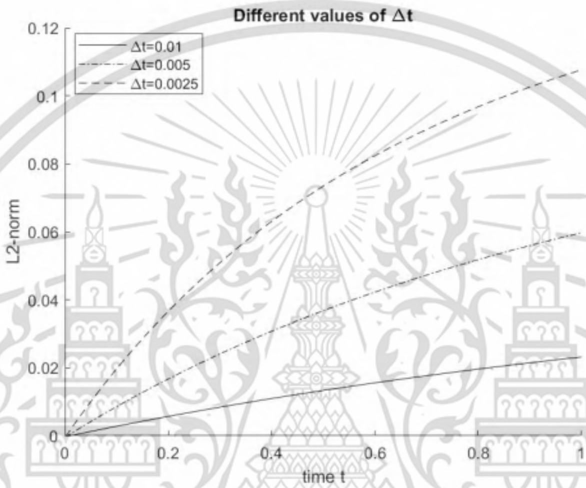


Figure 4.12. L_2-norm errors for different values of Δt.

Table 4.11. L_2- norm errors for different Δt at t = 1.0.

L	L_2-norm errors		
	Δt = 0.01	Δt = 0.005	Δt = 0.0025
20	4.1229e-02	6.6091e-02	8.6869e-02
50	3.0995e-02	6.3303e-02	1.0080e-01
80	2.5542e-02	6.0682e-02	1.0527e-01
100	2.3162e-02	5.9709e-02	1.0763e-01

Figure 4.13 illustrates the L_2-norm errors for different values of coefficient υ when Δx =0.01, P =2, A =0, B =1, T =1, Δt =0.005, and υ = 0.01, 0.005, and 0.001. It shows that when υ increases the L_2-norm errors are increase.

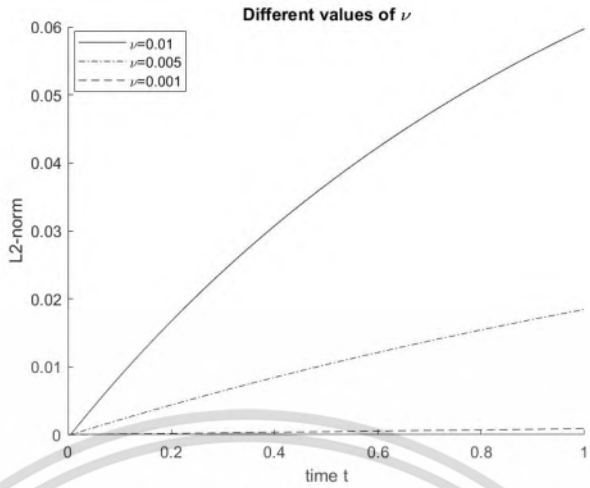
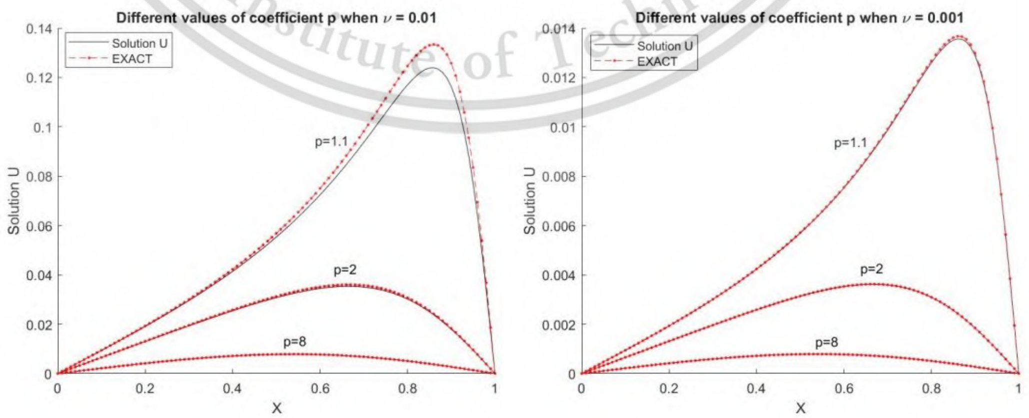


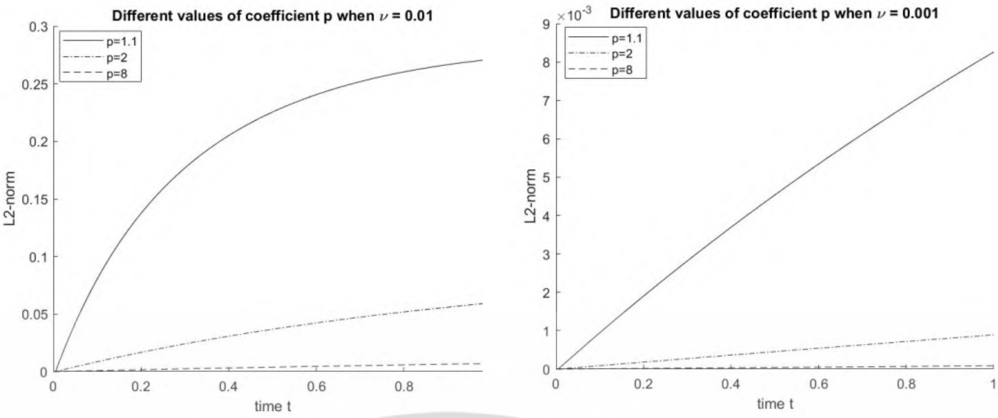
Figure 4.13. L_2-norm errors for different values of the coefficient υ.

Figure 4.14 illustrates the exact and numerical solutions for different values of p which are 1.1, 2, and 8, with (i) υ =0.01 and (ii) υ =0.001 when Δt =0.005, Δx =0.01, A =0, B =1, and T =1.0. The numerical results are closed to the exact solutions for each p. Figure 4.15 shows the L_2- norm errors of the numerical solutions provided in Figure 4.14. The results in Figure 4.15 show that the errors decrease for larger p for both υ =0.01 and 0.001. The L_2- norm errors presented in Figure 4.15 are illustrated in Table 4.12, where we also provide the errors of the numerical results when t =1. Notice that the errors decrease when the coefficient p increases for the different values of υ. Moreover, the errors of υ =0.001 are less than that of υ =0.01 for each p for both t =0.05 and t =1.



(i) (ii)

Figure 4.14. The exact and numerical solutions for different values of coefficient p, (i) υ = 0.01 and (ii) υ = 0.001 at t = 0.05.



(i) (ii)

Figure 4.15. L_2-norm errors for different values of the coefficient p, (i) υ = 0.01 and (ii) υ = 0.001 at t = 0.05.

Table 4.12. L_2-norm errors for different values of coefficient p.

p	t = 0.05		t = 1.0	
	υ = 0.01	υ = 0.001	υ = 0.01	υ = 0.001
1.1	5.0433e-03	5.0715e-05	2.7116e-01	8.2592e-03
2	4.7092e-04	4.7138e-06	5.9709e-02	8.9120e-04
8	4.3606e-05	4.3643e-07	6.9800e-03	8.4904e-05

Next, we compare the L_2-norm errors of the numerical solutions obtained from the three different methods. Tables 4.13 and 4.14 show the L_2-norm errors at t = 0.01 and t = 1, respectively. The values of variables are Δt =0.005, Δx =0.05, 0.02, 0.0125, and 0.01 or L =20, 50, 80, and 100, υ =0.01, p =2, A =0, and B =1. The L_2-norm errors at time t = 0.01 when using the Implicit method are better than that of the Crank-Nicolson method and the L_2-norm errors obtained from the Crank-Nicolson method are better than the explicit method.

Table 4.13. L_2-norm errors for each method at time t = 0.01.

Method	L_2-norm errors			
	$L = 20$	$L = 50$	$L = 80$	$L = 100$
Explicit	0.00077463	0.00055379	0.00049057	0.00047098
Implicit	0.00077457	0.00055369	0.00049046	0.00047086
Crank-Nicolson	0.00077460	0.00055374	0.00049052	0.00047092

Although the L_2-norm errors obtained from the three different schemes are different, the numerical solutions for larger t are very closed, see Table 4.14. Therefore, the method to find the second initial condition may effect the numerical solutions but it is not much different, which shows that this method is consistency.

Table 4.14. L_2-norm errors for each method at time t = 1.0.

Method	L_2-norm errors			
	$L=20$	$L=50$	$L=80$	$L=100$
Explicit	0.0660910205	0.0633032801	0.0606818648	0.0597094974
Implicit	0.0660910151	0.0633032601	0.0606818300	0.0597094537
Crank-Nicolson	0.0660910178	0.0633032701	0.0606818474	0.0597094755

Chapter 5

Conclusion

In this work, we propose efficient numerical schemes based on the finite difference and the implicit Adams-Moulton methods to find the numerical solutions of the one-dimensional Burgers equation. Due to the multi-step method, we apply the explicit method, the implicit method or the Crank-Nicolson method to calculate the approximation U_m^1 from the initial condition U_m^0, $m \in \mathbb{N}$ Then U_m^n, $n \geq 2$ is obtained by using the implicit Adams-Moulton method. The numerical results are compared with the exact solutions. The numerical solutions converge to the exact solution, where the L_2-norm errors are shown in Tables 4.1, 4.2, 4.5, 4.6, 4.9, and 4.10. Because of the implicit Adams-Moulton method, there is no need to have honesty small Δt to ensure the stability. The L_2-norm errors for different Δt at t =1.0 are shown in Tables 4.3, 4.7, and 4.11. The error is the largest at the smallest Δt because of the accumulated errors from the number of iterations. For different coefficients p and parameter υ, the L_2-norm errors of the least υ and largest p are smallest as shown in Tables 4.4, 4.8, and 4.12. The method used for finding U_m^1 influences the solutions at t =0.01 as shown in Table 4.13 and for greater t, t =1.0 in Table 4.14, the values of the errors of the three different methods are almost equal.

References

[1] Andallah, L. Sheikhl, A. and Kowser, A. 2014. "A comparative study of finite difference scheme for burgers equation." *Gazi University Journal of Science*. 27(4) : 1045–1052.

[2] Anonymous. 2020. Multistep Methods. [Online]. Available : https://www3.nd.edu/~zxu2/acms40390F12/Lec-5.6.pdf.

[3] Anonymous. 2020. Multistep Methods. [Online]. Available : http://www.math.iit.edu /~fass/478578_Chapter_2.pdf.

[4] Anonymous. 2020. Taylor's Theorem. [Online]. Available : https://www.dcs.warwick.ac.uk/people/academic/Steve.Russ/cs131/NOTE26.PDF

[5] Anonymous. 2020. The finite difference method. [Online]. Available : https://www.ljll.math.upmc.fr/frey/cours/UdC/ma691/ma691_ch6.pdf.

[6] Arleemeen, K. Klinthub, T. Chaiyasit, A. Sanprasert, W. and Wuttanachamsri, K. 2019. "A second-order accurate numerical approach for burgers equation." *The 11th NPRU National Academic Conference Nakhon Pathom Rajabhat University*, 82(11).

[7] Arleemeen, K. Klinthub, T. Chaiyasit, A. Sanprasert, W. and Wuttanachamsri, K. 2018. "A second-order accurate numerical approach for burgers equation." Bachelor of Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang.

[8] Bui, T. 2020. Explicit and Implicit Methods In Solving Differential Equations. [Online]. Available : http://digitalcommons.uconn.edu/srhonors_theses/119.

[9] Deturck, D. and Wilf, H.S. 2002. "Lectures on numerical analysis." 74–79. Department of Mathematics University of Pennsylvania. Philadelphia.

[10] Kimc, S. Wooc, G. Xiea, S. Heob, S. and Yic, S. 2008. "Numerical solution of one-dimensional burgers equation using reproducing kernel function" *Journal of Computational and Applied Mathematics.* 214 : 417-434.

[11] Kincaid, D. and Cheney, W. 1991. "Mathematics of Scientific Computing", Inc..

[12] Kuzmin, D. 2020. Introduction to computational fluid dynamics. [Online]. Available : https://www.mathematik.uni-dortmund.de/kuzmin/cfdintro /lecture1.

[13] Landajuela, M. 2011. "Burgers equation." 1-20. Basque center for applied mathematics. BCAM Internship - Summer 2011.

[14] Mohamed, N.A. 2018. "Fully implicit scheme for solving burgers' equation based on finite difference method." *The Egyptian International Journal of Engineering Sciences and Technology*. 26 : 38–44.

[15] Mukundan, V. and Awasthi, A. 2016. "Linearized implicit numerical method for burgers equation." *Walter de Gruyter.* 5(4) : 219–234.

[16] Perrin, A. and Hu, H.H. 2006. "An Explicit Finite-Difference Scheme for Simulation of Moving Particles." *Journal of Computational Physics*. 212(1) : 166-187.

[17] Rosly, S. Zulkifli, S. Zainuddin, N. and Yacob, J. 2020. Mathematical modelling of burgers equation applied in traffic flow. [Online]. Available : https://www.research gate.net/publication/320398257.

[18] Uauwongsarot. N. Phairat, S. and Kaewbanjak, N. 2017. "An improvement of a numerical method for solving burgers equation." *The 22nd Annual Meeting in Mathematics.* 22 : 182.

[19] Webb, K. 2020. Section 4: Roundoff and Truncation errors. [Online]. Available : http://web.engr.oregonstate.edu/~webbky/MAE4020_5020_files.

[20] Zhang, P.G. and Wang, J.P. 2012. "A predictor-corrector compact finite difference scheme for burgers equation." *Applied Mathematics and Computation.* 219 : 892–898.

Appendix

Appendix A

Far East Journal of Mathematical Sciences (FJMS)
© 2021 Pushpa Publishing House, Prayagraj, India
http://www.pphmj.com
http://dx.doi.org/10.17654/MS129010021
Volume 129, Number 1, 2021, Pages 21-34 ISSN: 0972-0871

APPLICATION OF IMPLICIT ADAMS-MOULTON METHOD TO BURGERS EQUATION

Apichaya Chaiyasit and Kanognudge Wuttanachamsri

Department of Mathematics
Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
Bangkok, 10520, Thailand

Abstract

In this paper, a new scheme for finding the numerical solutions of the one-dimensional Burgers equation is provided. A combination of the finite difference method and the implicit Adams-Moulton method is employed to calculate the solution of the nonlinear equation. The nonlinear term is linearized by using a weighted technique. Because of the multi-step scheme, the central finite difference method is used to find another initial condition. Numerical results are obtained and compared with the exact solution. The numerical solutions converge to the exact solution when the number of grid points increases. The L_2-norm errors of the numerical results for different values of parameters are provided to demonstrate the L_2-norm errors of the results and show the direction of the solutions when the parameters are changed. Applications include fluid flow such as traffic flow.

Received: November 23, 2020; Accepted: December 17, 2020
2020 Mathematics Subject Classification: 80M20, 39A14.
Keywords and phrases: Burgers equation, finite difference method, implicit method, Adams-Moulton method.

1. Introduction

The Burgers equation was applied in various areas in applied mathematics and fluid flow problems such as meteorological phenomena (rain, wind, floods, etc.), processes in the human body (blood flow, breathing, drinking, etc.) and Computational Fluid Dynamics (CFD) applications (aerodynamic shape design, high-speed train simulations, etc.) [4]. It is similar to the Navier-Stokes equation due to the combination of convection, diffusion and time-dependent terms [8]. Researchers have been interested in the study of numerical solutions of the Burgers equation especially in the fields of physics and engineering. For example, Landajuela [7] studied the flow of cars on a highway by employing the Burgers equation. There were several methods used to find the numerical solution of the Burgers equation such as Mohamed [8] solving nonlinear one-dimensional Burgers equation by using finite difference method. Mukundan and Awasthi [13] applied the method of lines (MOL) to solve the Burgers equation. Andallah et al. [6] studied a comparison of the LaxFriedrich and Lax-Wendroffs schemes to find the numerical solution of the Burgers equation. Kimc et al. [11] used the HopfCole transformation and a reproducing kernel function to calculate the solution of the Burgers equation. Zhang and Wang [10] used a predictor-corrector method called *MacCormack method* to determine the solutions. Arleemeen et al. [5] provided the solution of the Burgers equation by using the finite difference method and the second-order Runge-Kutta method. Uauwongsarot et al. [9] studied the numerical solution of the Burgers equation by using the sixth-order compact finite difference method and the McCormack method.

In this paper, we apply the finite difference method, the implicit Adams-Moulton method to the Burgers equation to find the numerical solutions. The central finite difference method is used to discretize the spatial terms of the Burgers equation and the Adams-Moulton method is applied to the time derivative term. The nonlinear term is solved by using a linear extrapolation. In Section 2, we present the Burgers equation and initial and boundary conditions. The numerical methods for finding the solutions are indicated in Section 3. The exact solution is presented in Section 4. Numerical solutions and the verifications are provided in Section 5. In the last section, we draw conclusion.

2. Burgers Equation and Essential Conditions

In this section, we provide the one-dimensional Burgers equation with both initial and boundary conditions. The Burgers equation on domain (A, B) [5] is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \upsilon\frac{\partial^2 u}{\partial x^2} \quad A < x < B, \quad t \in [0, T], \tag{2.1}$$

where T is the final time, with an initial condition

$$u(x, 0) = g(x), \quad A \le x \le B \tag{2.2}$$

and boundary conditions

$$u(A, t) = f_1(t), u(B, t) = f_2(t), t \in [0, T], \tag{2.3}$$

where the variable υ is the kinematic viscosity and functions f_1, f_2 and g are constraints.

3. Numerical Method and Procedure

To apply a second-order finite difference method and the Adams-Moulton method to the Burgers equation, we first divide the domain into L grid points and define $x_m = x_{m-1} + \Delta x$, $m = 2, 3, ..., L$ with a constant spacing $\Delta x = x_m - x_{m-1}$, where x_1 and x_L represent the boundary of the domain and $\Delta t = t_n - t_{n-1}$, $n = 1, 2, ..., M$.

3.1. Finite difference method

We use the central finite difference method to the spatial terms of the Burgers equation. Therefore, the Burgers equation (2.1), becomes

$$\left[\frac{\partial u}{\partial t}\right]_m + \bigcup_m(t)\left[\frac{\bigcup_{m+1}(t) - \bigcup_{m-1}(t)}{2h}\right]$$

$$= \upsilon\left[\frac{\bigcup_{m+1}(t) - 2\bigcup_m(t) + \bigcup_{m-1}(t)}{h^2}\right], \qquad (3.1)$$

or

$$\left[\frac{\partial u}{\partial t}\right]_m = \frac{\upsilon}{h^2}\left[\bigcup_{m+1}(t) - 2\bigcup_m(t) + \bigcup_{m-1}(t)\right]$$

$$-\frac{\bigcup_m(t)}{2h}\left[\bigcup_{m+1}(t) - \bigcup_{m-1}(t)\right], \qquad (3.2)$$

where $\bigcup_m(t) = u(x_m, t)$, $m = 2, 3, \ldots, L-1$, $\left[\frac{\partial u}{\partial t}\right]_m$ is $\frac{\partial u}{\partial t}$ at the point $x = x_m$ and $h = \Delta x$.

3.2. Implicit Adams-Moulton method

We apply the Adams-Moulton method to the time derivative term of equation (3.2). The third-order implicit Adams-Moulton used to find the solution of the equation $u' = f(t, u(t))$ is available in [3]. We have

$$u_{n+1} - u_n = l\left(\frac{5}{12}f(t_{n+1}, u_{n+1}) + \frac{2}{3}f(t_n, u_n) - \frac{1}{12}f(t_{n-1}, u_{n-1})\right), \qquad (3.3)$$

where $u_n = u(t_n)$, $n = 1, 2, \ldots, M-1$ and $0 \le t_n \le T$. The consistency and stability of equation (3.3) can be verified from the polynomials [3]:

$$k(w) = w^2 - w,$$

$$r(w) = \frac{5}{12}w^2 + \frac{2}{3}w - \frac{1}{12}.$$

The roots of k are 0 and +1, which are simple roots lying in the unit disk. Moreover, $k'(w) = 2w - 1$ and $k'(1) = 1 = r(1)$. From Chapter 8 in [3], the Adams-Moulton method is stable and consistence. Applying equation (3.3) to equation (3.2), we have

$$
\begin{aligned}
U_m^{n+1} = U_m^n
&+ \frac{5l}{12}\left[\frac{\upsilon}{h^2}\left(U_{m+1}^{n+1} - 2U_m^{n+1} + U_{m-1}^{n+1}\right) - \frac{U_m^{n+1}}{2h}\left(U_{m+1}^{n+1} - U_{m-1}^{n+1}\right)\right] \\
&+ \frac{2l}{3}\left[\frac{\upsilon}{h^2}\left(U_{m+1}^{n} - 2U_m^{n} + U_{m-1}^{n}\right) - \frac{U_m^{n}}{2h}\left(U_{m+1}^{n} - U_{m-1}^{n}\right)\right] \\
&- \frac{l}{12}\left[\frac{\upsilon}{h^2}\left(U_{m+1}^{n-1} - 2U_m^{n-1} + U_{m-1}^{n-1}\right) - \frac{U_m^{n-1}}{2h}\left(U_{m+1}^{n-1} - U_{m-1}^{n-1}\right)\right].
\end{aligned} \tag{3.4}
$$

Since equation (3.4) consists of a nonlinear term at time t_{n+1},

$$U_m^{n+1}\left(U_{m+1}^{n+1} - U_{m-1}^{n+1}\right), \tag{3.5}$$

we approximate U_m^{n+1} by using a linear extrapolation of U_m^n and U_m^{n-1} [8]:

$$U_m^{n+1} \cong \left(1 + \left(\frac{j_{n+1}}{j_n}\right)\right)U_m^n - \left(\frac{j_{n+1}}{j_n}\right)U_m^{n-1}, \tag{3.6}$$

where $j_{n+1} = t_{n+1} - t_n$. Then equation (3.5) becomes

$$\left(\left(1 + \left(\frac{j_{n+1}}{j_n}\right)\right)U_m^n - \left(\frac{j_{n+1}}{j_n}\right)U_m^{n-1}\right)\left(U_{m+1}^{n+1} - U_{m-1}^{n+1}\right). \tag{3.7}$$

Substituting equation (3.7) into equation (3.4), we have

$$\bigcup_m^{n+1} = \bigcup_m^n + \frac{5\upsilon l}{12h^2}\left[\bigcup_{m+1}^{n+1} - 2\bigcup_m^{n+1} + \bigcup_{m-1}^{n+1}\right]$$

$$- \frac{5l}{24h}\left(\left(1+\left(\frac{j_{n+1}}{j_n}\right)\right)\bigcup_m^n - \left(\frac{j_{n+1}}{j_n}\right)\bigcup_m^{n-1}\right)\left[\bigcup_{m+1}^{n+1} - \bigcup_{m-1}^{n+1}\right]$$

$$+ \frac{2\upsilon l}{3h^2}\left[\bigcup_{m+1}^{n} - 2\bigcup_m^{n} + \bigcup_{m-1}^{n}\right] - \frac{l}{3h}\bigcup_m^n\left[\bigcup_{m+1}^{n} - \bigcup_{m-1}^{n}\right]$$

$$- \frac{\upsilon l}{12h^2}\left[\bigcup_{m+1}^{n-1} - 2\bigcup_m^{n-1} + \bigcup_{m-1}^{n-1}\right] + \frac{l}{24h}\left[\bigcup_{m+1}^{n-1} - \bigcup_{m-1}^{n-1}\right]. \quad (3.8)$$

In this work, we use equally spaced time. Then $j_n = \Delta t$, for all n. Multiplying equation (3.8) by $24h^2$, we have

$$a_m\bigcup_{m+1}^{n+1} + b_m\bigcup_m^{n+1} + c_m\bigcup_{m-1}^{n+1} = f_m, \quad (3.9)$$

where

$$a_m = 10hl\bigcup_m^n - 10\upsilon - 5h\bigcup_m^{n-1},$$

$$b_m = 24h^2 + 20\upsilon l,$$

$$c_m = 5hl\bigcup_m^{n-1} - 10\upsilon l - 10hl\bigcup_m^n,$$

$$f_m = \left(16\upsilon l - 8hl\bigcup_m^n\right)\bigcup_{m+1}^n + (24h^2 - 32\upsilon l)\bigcup_m^n + \left(16\upsilon l + 8hl\bigcup_m^n\right)\bigcup_{m-1}^n$$

$$+ \left(hl\bigcup_m^{n-1} - 2\upsilon l\right)\bigcup_{m+1}^{n-1} + 4\upsilon l\bigcup_m^{n-1} - \left(hl\bigcup_m^{n-1} + 2\upsilon l\right)\bigcup_{m-1}^{n-1},$$

and $m = 2, 3, \ldots, L-1$. From equation (3.9), to find $\bigcup_m^{n+1}$, it is necessary to know $\bigcup_m^{n-1}$ and $\bigcup_m^n$, $m = 1, 2, \ldots, L$. However, $\bigcup_m^{n-1}$ can be obtained from the initial condition. To calculate the value U_m^n, we apply the finite difference method again to equation (2.1). This yields

$$U_m^n = U_m^{n-1} + \frac{\Delta t \upsilon}{h^2}\left[U_{m+1}^{n-1} - 2U_m^{n-1} + U_{m-1}^{n-1}\right]$$

$$-\frac{\Delta t U_m^{n-1}}{2h}\left[U_{m+1}^{n-1} - U_{m-1}^{n-1}\right], \quad (3.10)$$

where $m = 2, 3, ..., L-1$. Now we have U_m^{n-1} and U_m^n. Then we are really to find the numerical solutions.

4. Exact Solutions

To verify the numerical result, we compare the solutions with the exact solution, which is as in [10]:

$$u(x, t) = \frac{2\upsilon\pi e^{-\pi^2\upsilon t}\sin(\pi x)}{p + e^{-\pi^2\upsilon t}\cos(\pi x)} \quad (4.1)$$

with the initial condition

$$u(x, 0) = \frac{2\upsilon\pi\sin(\pi x)}{p + \cos(\pi x)}, \quad 0 \le x \le 1, \quad (4.2)$$

and the boundary conditions

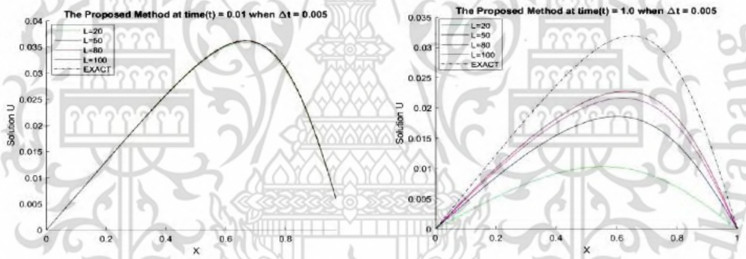
$$u(0, t) = u(1, t) = 0, \quad t > 0, \quad (4.3)$$

where p is a constant.

5. Numerical Results

The solution of the Burgers equation is calculated by using equation (3.9). The initial U_m^0 is obtained from the initial condition and U_m^1 is calculated from equation (3.10). Substituting $\bigcup_m^0$ and $\bigcup_m^1$ in equation (3.9), we have $\bigcup_m^2$. Then $\bigcup_m^3$ can be determined from $\bigcup_m^1$ and $\bigcup_m^2$. Continuing this process, we have $\bigcup_m^n$, $1 \le m \le L$, for all n.

Figure 1 shows the numerical solutions of equation (3.9) at (i) $T = 0.01$ and (ii) $T = 1$ with the initial condition in equation (4.2) and the boundary conditions in equation (4.3). The values of variables are $\Delta t = 0.005$, $h = 0.05, 0.02, 0.0125$ and 0.001 or $L = 20, 50, 80$ and 100, respectively, $\upsilon = 0.01$, $p = 2$, $A = 0$ and $B = 1$. At $T = 0.01$, it shows the numerical results converge to the exact solution. It means that the solution obtained from equation (3.10) is accurate so that it can be used to find the solution for the next time step. Then the initial condition and the solution at $T = 0.01$ are placed in equation (3.9) to find the numerical solution for the next time step. We continue this process until $T = 1$.



(i) (ii)

Figure 1. Exact solution and numerical solutions at (i) $T = 0.01$ and (ii) $T = 1.0$.

From Figure 1, the numerical results at $T = 1.0$ converge to the exact solution when the number of grid points increases. The results at each point x and each L are presented in Tables 1 and 2. The L_2-norm errors, $L_2 = \| u_{exact} - u_{solutions} \|_2$, of each L are provided in the last row of the Tables. Notice that the smallest errors occurred at $L = 100$ for both $T = 0.01$ and $T = 1.0$.

Table 1. Exact solutions and the numerical solutions at each point x at time $T = 0.01$

X	Exact solutions	Numerical solutions			
		$L = 20$	$L = 50$	$L = 80$	$L = 100$
0.1	0.00657	0.00655	0.00657	0.00657	0.00657
0.2	0.01314	0.01308	0.01312	0.01312	0.01313
0.3	0.01963	0.01954	0.01959	0.01960	0.01960
0.4	0.02586	0.02572	0.02579	0.02581	0.02582
0.5	0.03138	0.03120	0.03130	0.03132	0.03133
0.6	0.03530	0.03506	0.03518	0.03521	0.03522
0.7	0.03594	0.03567	0.03582	0.03585	0.03587
0.8	0.03096	0.03069	0.03084	0.03088	0.03089
0.9	0.01848	0.01829	0.01841	0.01844	0.01844
L_2-norm		7.7463e-04	5.5379e-04	4.9057e-04	4.7098e-04

Table 2. Exact solutions and the numerical solutions at each point x at time $T = 1.0$

x	Exact solutions	Numerical solutions			
		$L = 20$	$L = 50$	$L = 80$	$L = 100$
0.1	0.00615	0.00271	0.00438	0.00494	0.00515
0.2	0.01224	0.00525	0.00855	0.00963	0.01003
0.3	0.01819	0.00747	0.01232	0.01389	0.01445
0.4	0.02375	0.00918	0.01547	0.01751	0.01823
0.5	0.02846	0.01016	0.01771	0.02023	0.02113
0.6	0.03148	0.01021	0.01860	0.02158	0.02266
0.7	0.03138	0.00916	0.01757	0.02084	0.02207
0.8	0.02641	0.00696	0.01405	0.01710	0.01832
0.9	0.01545	0.00376	0.00791	0.00987	0.01069
L_2-norm		6.6091e-02	6.3303e-02	6.0682e-02	5.9709e-02

Figure 2 shows the L_2-norm errors for $\Delta t = 0.01$, 0.005 and 0.0025 when $L = 100$, $p = 2$, $A = 0$, $B = 1$, $T = 1$ and $\upsilon = 0.01$. The graphs illustrate that L_2-norm errors decrease for larger Δt.

The L_2-norm errors in Figure 2 are presented in Table 3. The error when $\Delta t = 0.01$ is better than that of $\Delta t = 0.005$ and 0.0025 for each L. This is because of the implicit Adams-Moulton method and the accumulated errors of the increasing number of iterations. The error when $L = 100$ is smallest as expected.

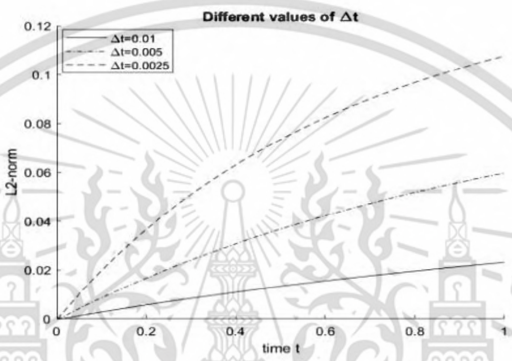


Figure 2. L_2-norm errors for different values of Δt.

Table 3. L_2-norm errors for different Δt at $T = 1.0$

L	L_2-norm errors		
	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.0025$
20	4.1229e-02	6.6091e-02	8.6869e-02
50	3.0995e-02	6.3303e-02	1.1080e-01
80	2.5542e-02	6.0682e-02	1.0527e-01
100	2.3162e-02	5.9709e-02	1.0063e-01

Figure 3 illustrates L_2-norm errors for different values of coefficient υ when $\Delta x = 0.01$, $p = 2$, $A = 0$, $B = 1$, $T = 1$, $\Delta t = 0.005$ and $\upsilon = 0.01$, 0.005 and 0.001. It shows that when υ increases the L_2-norm errors increase.

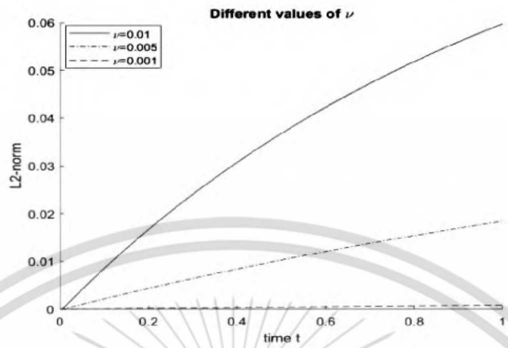
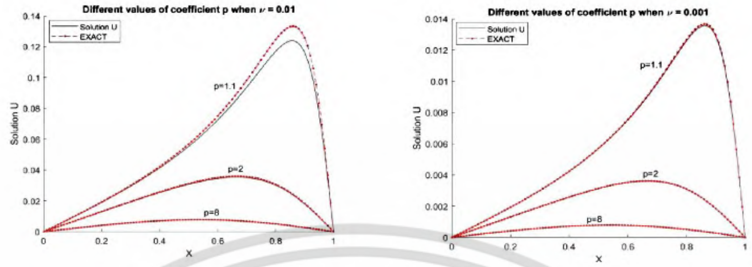


Figure 3. L_2-norm errors for different values of the coefficient υ.

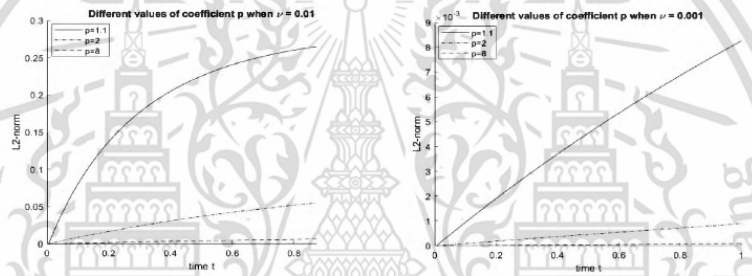
Figure 4 illustrates the exact and numerical solutions for different values of p which are 1.1, 2 and 8, with (i) $\upsilon = 0.01$ and (ii) $\upsilon = 0.001$ when $\Delta t = 0.005$, $\Delta x = 0.01$, $A = 0$, $B = 1$ and $T = 0.05$. The numerical results are close to the exact solution. Figure 5 shows the L_2-norm errors of the numerical solutions provided in Figure 4. The results in Figure 5 show that the errors decrease for larger p for both $\upsilon = 0.01$ and 0.001.

The L_2-norm errors presented in Figure 5 are illustrated in Table 4, where we also provide the errors of the numerical results when $T = 1$. Notice that the errors decrease when the coefficient p increases for the different values of υ. Moreover, the errors of $\upsilon = 0.001$ are less than that of $\upsilon = 0.01$ for each p for both $T = 0.05$ and $T = 1$.



(i) (ii)

Figure 4. The exact and numerical solutions for different values of coefficient p, with (i) $\upsilon = 0.01$ and (ii) $\upsilon = 0.001$ at $T = 0.05$.



(i) (ii)

Figure 5. L_2-norm errors for different values of the coefficient p, with (i) $\upsilon = 0.01$ and (ii) $\upsilon = 0.001$ at $T = 0.05$.

Table 4. L_2-norm errors for different values of coefficient p

p	$T = 0.05$		$T = 1.0$	
	$\upsilon = 0.01$	$\upsilon = 0.001$	$\upsilon = 0.01$	$\upsilon = 0.001$
1.1	5.0472e-03	5.0719e-05	2.7116e-01	8.2592e-03
2	4.7098e-04	4.7139e-06	5.9709e-02	8.9120e-04
8	4.3613e-05	4.3643e-07	6.9800e-03	8.4904e-05

6. Conclusion

In this work, we propose an efficient numerical scheme based on the finite difference and the implicit Adams-Moulton methods to find the numerical solutions of the one-dimensional Burgers equation. Due to the multi-step method, we apply the second-order central difference to calculate the approximation $\bigcup_m^1$ from the initial condition $\bigcup_m^0$, $m \in \mathbb{N}$. Then $\bigcup_m^n$, $n \geq 2$ is obtained by using the implicit method. The numerical results are compared with the exact solutions. The numerical solutions converge to the exact solution, where the L_2-norm errors are shown in Tables 1 and 2. The L_2-norm errors for different Δt are shown in Table 3. Because of the implicit method, there is no need to have small Δt to ensure the stability. The L_2-norm errors for different Δt at $T = 1.0$ are shown in Table 3. The error is the largest for the smallest Δt because of the accumulated errors from the number of iterations. For different coefficients p and parameters υ, the L_2-norm errors of the least υ and the largest p are the smallest as shown in Table 4.

References

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

[2] D. Deturck and H. S. Wilf, Lectures on numerical analysis, Department of Mathematics, University of Pennsylvania, Philadelphia, 2002, pp. 74-79.

[3] D. Kincaid and W. Cheney, Mathematics of Scientific Computing, Inc., 1991.

[4] D. Kuzmin, Introduction to computational fluid dynamics, https://www.mathematik.uni-dortmund.de/ kuzmin/cfdintro/lecture1.pdf.

[5] K. Arleemeen, W. Sanprasert, A. Chaiyasit, T. Klinthub and K. Wuttanachamsri, A second-order accurate numerical approach for Burgers equation, The 11th NPRU National Academic Conference Nakhon Pathom Rajabhat University, Vol. 82, 2019.

[6] L. Andallah, A. Sheikhl and A. Kowser, A comparative study of finite difference scheme for Burgers equation, Gazi University Journal of Science 27 (2014), 1045-1052.

[7] M. Landajuela, Burgers equation, Basque center for applied mathematics, BCAM Internship-Summer 2011, 2011.

[8] N. A. Mohamed, Fully implicit scheme for solving Burgers' equation based on finite difference method, The Egyptian International Journal of Engineering Sciences and Technology 26 (2018), 38-44.

[9] N. Uauwongsarot, S. Phairat and N. Kaewbanjak, An improvement of a numerical method for solving Burgers equation, The 22nd Annual Meeting in Mathematics (AMM 2017), Chiang Mai University, Chiang Mai, Thailand, 2017, 5.

[10] P. G. Zhang and J. P. Wang, A predictor-corrector compact finite difference scheme for Burgers equation, Applied Mathematics and Computation 219 (2012), 892-898.

[11] S. Kimc, G. Wooc, S. Xiea, S. Heob and S. Yic, Numerical solution of one-dimensional Burgers equation using reproducing kernel function, Journal of Computational and Applied Mathematics 214 (2008), 417-434.

[12] T. Bui, Explicit and Implicit Methods in Solving Differential Equations, University of Connecticut, Honors Scholar Theses, 119, 2010. http://digitalcommons. uconn.edu/srhonors_theses/119.

[13] V. Mukundan and A. Awasthi, Linearized implicit numerical method for Burgers equation, Walter de Gruyter, 2016, pp. 219-234.

Author Biography

Name	Miss Apichaya Chaiyasit
Date of Birth	23 May 1996
Address	200/14, moo 1, Watprathong Road, Thepkrasattri, Thalang, Phuket, 83110
Education (2018)	Bachelor of Science in Applied Mathematics (GPA 3.02) 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. Chaiyasit, A. and Wuttanachamsri, K. 2021. "Application of Implicit Adams-Moulton method to Burgers equation." *Far East Journal of Mathematical Sciences(FJMS).* 129(1) : 21-34.