Proceedings of Science and Mathematics
https://science.utm.my/procscimath
Volume 24 (2024) 47-55

Numerical Solution of Unsteady Burger's Equation

Fitrah Atikah Aminoordin, Shazirawati Mohd Puzi
Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia
Corresponding author: shazirawati@utm.my

Abstract Burger's equation is a basic partial difference equation that can model nonlinear waves and fluid dynamics. This study presents the numerical solutions of Burger's equation using explicit and implicit finite difference schemes. In these methods, the second-order central finite differences are used to discretize the equation into an ordinary differential equation. Numerical calculations are conducted using C++ and MATLAB software and the results are demonstrated using Microsoft Excel. Numerical results are validated by comparing the accuracy of the results with the analytical solution. In this work, the explicit and implicit finite difference schemes approach is used to numerically solve the nonlinear unsteady Burger's equation, which focuses on accuracy and stability to produce results that closely resemble the analytical solution.
Keywords Partial Differential Equations; Explicit Finite Difference Scheme; Implicit Finite Difference Scheme; Ordinary Differential Equations (ODE)

1. Introduction

As a simplified version of the Navier-Stokes equation, Burger's equation is frequently utilized to investigate rarefaction and shock waves. Named after Dutch mathematician Johannes Martinus Burgers, the equation incorporates the Reynold number (Re) to characterize fluid flow, which measures the relative importance of viscous versus inertial forces. This dimensionless number, introduced by Osborne Reynolds, helps predict fluid behaviours like turbulence onset.

To solve the unsteady Burger's equation, numerical methods such as the method of lines (MOL) and finite difference method (FDM) are commonly employed. This study focuses on solving the unsteady Burger's equation, a nonlinear partial differential equation significant in fluid dynamics and wave propagation studies. The research aims to explore and implement suitable numerical methods, specifically explicit and implicit finite difference schemes, to solve this equation. The study compares the accuracy of these numerical approximations with analytical solutions.

2. Literature Review

2.1. Burger's Equation

Burger's equation, a mathematical model combining linear diffusion and nonlinear wave motion, is used to study fluid dynamics and nonlinear wave evolution in a one-dimensional medium. It is an expansion of the linear Burger's equation, addressing both convection and diffusion. The nonlinear Burger's equation is often written as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2}$$

[1] introduced a numerical method combining the modified cubic B-spline differential quadrature (CN-MCDQ) technique with the Crank-Nicolson scheme to approximate solutions to Burger's equation. This method approximates derivatives using differential quadrature, resulting in a system of equations solved with the Crank-Nicolson technique, proving to be accurate, efficient, and user-friendly compared to other numerical solutions. The differential quadrature method for solving partial differential equations was first proposed by Bellman *et al*. [2].

2.2. Unsteady Burger's Equation

[3] introduces new fully implicit numerical schemes for solving one-dimensional and two-dimensional unsteady Burger's equation. This approach transforms the equation into a nonlinear system of ordinary differential equations (ODEs), using a second-order finite difference method for spatial discretization and the backward differentiation formula of order two (BDF2) for time discretization. Thomas' method linearizes the nonlinear term, converting it into a system of linear algebraic equations. The schemes, tested with Dirichlet and Neumann boundary conditions, are straightforward, precise, and efficient, even at high Reynold numbers.

[4] propose new numerical methods for solving the nonlinear unsteady Burger's equation. These methods discretize all variables except time, transforming the PDE system into a nonlinear ODE system. Stability is confirmed using Lyapunov's criteria, and implicit stiff solvers with backward differentiation formulas of orders one, two, and three are used to solve the nonlinear ODE system.

In this study, the one-dimensional nonlinear unsteady Burger's equation is considered:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} \tag{1}$$

2.3. Finite Difference Method

The finite difference method (FDM) is a set of numerical techniques used to solve differential equations by approximating derivatives with finite differences. This process, called discretization, breaks the spatial domain and period into finite steps. FDM transforms ordinary and partial differential equations, whether linear or nonlinear, into systems of linear equations that can be solved using matrix algebra techniques. Its efficiency in computation and ease of implementation makes FDM a widely used approach in modern numerical analysis.

For solving the one-dimensional equation (1), the explicit scheme and the implicit scheme are used. For the explicit method, a forward difference and a second-order central difference for the space derivative are used. The recurrence equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &\approx \frac{u_i^{j+1} - u_i^j}{k} \\ \frac{\partial^2 u}{\partial x^2} &\approx \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} \end{aligned} \tag{2}$$

The recurrence equation for the implicit method:

$$\begin{aligned} \frac{\partial u}{\partial t} &\approx \frac{u_i^{j+1} - u_i^j}{k} \\ \frac{\partial^2 u}{\partial x^2} &\approx \frac{u_{i+1}^{j+1} - 2u_i^j + u_{i-1}^{j+1}}{h^2} \end{aligned} \tag{3}$$

The difference between the precise analytical answer and the approximation in a method's solution is known as the error. Round-off error, or the loss of precision resulting from computer rounding of decimal quantities, and truncation error, or discretization error, or the difference between the exact solution of the original differential equation and the exact quantity assuming perfect arithmetic, are the two sources of error in finite difference method.

3. Methodology

3.1. Discretization of One-Dimensional Nonlinear Unsteady Burger's Equation Using Explicit Finite Difference Scheme

An explicit finite difference scheme discretizes a one-dimensional nonlinear unsteady Burger's equation by approximating the continuous partial differential equation on a discrete grid in both space and time. The spatial domain is divided into a grid of points with a uniform spatial step size, and the time domain is separated into time steps of a specified size. This numerical approach allows for the solution of the PDE by iteratively updating the values at each grid point.

The forward difference method and the central finite difference are used to discretize equation (1) into:

$$u_i^{j+1} = u_i^j + k\left[\frac{1}{Re}\left(\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2}\right) - u_i^j\left(\frac{u_{i+1}^j - u_{i-1}^j}{2h}\right)\right] \tag{4}$$

3.2. Discretization of One-Dimensional Nonlinear Unsteady Burger's Equation Using Implicit Finite Difference Scheme

An implicit finite difference scheme discretizes the one-dimensional nonlinear unsteady Burger's equation by approximating the continuous PDE with a numerical scheme where future values of the dependent variable are determined from current values through algebraic equations. This approach is particularly effective for PDEs with stability or convergence issues that explicit schemes struggle to solve. Similar to the explicit scheme, it requires discretization in both spatial and temporal domains. After discretization, equation (1) becomes:

$$\frac{u_i^{j+1} - u_i^j}{k} + u_i^j\left(\frac{u_{i+1}^{j+1} - u_{i-1}^{j+1}}{2h}\right) = \frac{1}{Re}\left(\frac{u_{i-1}^{j+1} - 2u_i^{j+1} + u_{i+1}^{j+1}}{h^2}\right) \tag{5}$$

Rearrange equation (5),

$$u_i^{j+1} - u_i^j + \frac{k}{2h}u_i^j\left(u_{i+1}^{j+1} - u_{i-1}^{j+1}\right) = \frac{k}{h^2 Re}\left(u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}\right)$$

Let $\alpha = \frac{k}{h^2 Re}$ and $\beta = \frac{k}{2h}$,

Factoring u_{i-1}^{j+1}, u_i^{j+1} and u_{i+1}^{j+1} at the left-hand side yields :

$$\left(-\alpha - \beta u_i^j\right)u_{i-1}^{j+1} + (1 + 2\alpha)u_i^{j+1} + \left(\beta u_i^j - \alpha\right)u_{i+1}^{j+1} = u_i^j \tag{6}$$

Equation (6) implies that for each step time, it will end up with a system of linear equations with

A as a tridiagonal matrix $A\bar{u} = \bar{B}$

For illustration, for 5×5 rectangular grid size, and with known initial and boundary conditions,

then by equation (6) at $j = 0$ we obtain the linear system:

$$(-\alpha - \beta u_1^0)u_0^1 + (1 + 2\alpha)u_1^1 + (\beta u_1^0 - \alpha)u_2^1 = u_1^0$$

$$(-\alpha - \beta u_2^0)u_1^1 + (1 + 2\alpha)u_2^1 + (\beta u_2^0 - \alpha)u_3^1 = u_2^0$$

$$(-\alpha - \beta u_3^0)u_2^1 + (1 + 2\alpha)u_3^1 + (\beta u_3^0 - \alpha)u_4^1 = u_3^0$$

$$(-\alpha - \beta u_4^0)u_3^1 + (1 + 2\alpha)u_4^1 + (\beta u_4^0 - \alpha)u_5^1 = u_4^0$$

Which can be seen as a tridiagonal system :

$$\begin{pmatrix} 1 + 2\alpha & \beta u_1^0 - \alpha & 0 & 0 \\ -\alpha - \beta u_2^0 & 1 + 2\alpha & \beta u_2^0 - \alpha & 0 \\ 0 & -\alpha - \beta u_3^0 & 1 + 2\alpha & \beta u_3^0 - \alpha \\ 0 & 0 & -\alpha - \beta u_4^0 & 1 + 2\alpha \end{pmatrix}\begin{pmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ u_4^1 \end{pmatrix} = \begin{pmatrix} u_1^0 - (-\alpha - \beta u_1^0)u_0^1 \\ u_2^0 \\ u_3^0 \\ u_4^0 - (\beta u_4^0 - \alpha)u_5^1 \end{pmatrix}$$

In the same manner, the linear tridiagonal system for $j = 2, 3,$ and 4 can be obtained. For a larger grid size, a computer program is used as a computational tool in obtaining the numerical solutions.

4. Results and Discussion

4.1. Parameter Settings

Considering the nonlinear unsteady Burger's equation (1), the following parameter settings are used for experimental purposes:

$$\Delta t = 0.001, \qquad \Delta x = 0.05, \qquad Re = 10$$

In this study, C++ and MATLAB are used to get the numerical solutions and Microsoft Excel is used for diagram illustration. The accuracy of numerical solutions of explicit and implicit finite difference schemes are compared with the following analytical solutions:

$$u(x,t) = \frac{0.1e^{-A}+0.5e^{-B}+e^{-C}}{e^{-A}+e^{-B}+e^{-C}} \tag{7}$$

where

$$A = 0.05Re(x - 0.5 + 4.95t)$$
$$B = 0.25Re(x - 0.5 + 0.75t)$$
$$C = 0.5Re(x - 0.375)$$

Initial condition at $t = 0$:

$$u(x,0) = \frac{0.1e^{-A}+0.5e^{-B}+e^{-C}}{e^{-A}+e^{-B}+e^{-C}} \tag{8}$$

Boundary conditions

at $x = -4$:

$$u(-4,t) = \frac{0.1e^{-A}+0.5e^{-B}+e^{-C}}{e^{-A}+e^{-B}+e^{-C}}$$

at $x = 4$:

$$u(4,t) = \frac{0.1e^{-A}+0.5e^{-B}+e^{-C}}{e^{-A}+e^{-B}+e^{-C}} \tag{9}$$

The analytical solutions are computed by using MATLAB and the results are visualized by using Microsoft Excel.

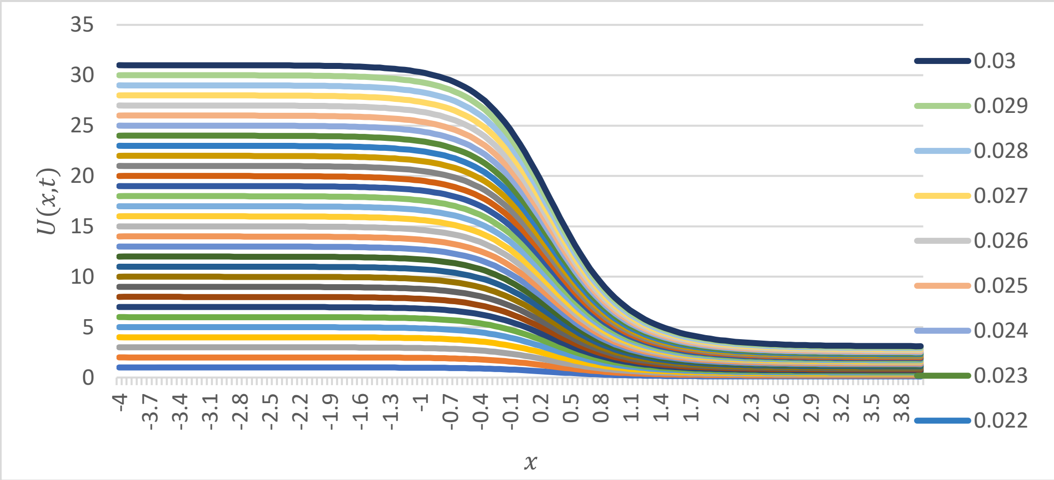

Figure 1 Analytical solutions for $\Delta x = 0.05$ and $\Delta t = 0.001$

4.2. Numerical Results of Explicit Finite Difference Scheme

The numerical results of Burger's equation in equation(1) are calculated explicitly by using C++ software with the initial condition and boundary conditions in equations (8) and (9). The results are visualized using Microsoft Excel and compared with the analytical solution in equation (7).

Table 1: Abbreviated explicit solutions for $\Delta x = 0.05$ and Δt=0.001

t/x	-4	-3.95	-3.9	...	3.9	3.95	4
0	0.999988	0.999986	0.999984	...	0.100445	0.100403	0.100364
0.001	0.999988	0.999986	0.999984	...	0.100445	0.100403	0.100365
0.002	0.999988	0.999986	0.999984	...	0.100446	0.100403	0.100364
0.003	0.999988	0.999986	0.999984	...	0.100446	0.100404	0.100364
0.004	0.999988	0.999986	0.999985	...	0.100446	0.100403	0.100364
0.005	0.999988	0.999986	0.999985	...	0.100446	0.100403	0.100366
...	...	...	...	...	...	...	...
0.026	0.999988	0.999987	0.999985	...	0.100452	0.100409	0.10037
0.027	0.999988	0.999987	0.999985	...	0.100452	0.100409	0.10037
0.028	0.999988	0.999987	0.999985	...	0.100453	0.10041	0.10037
0.029	0.999988	0.999987	0.999985	...	0.100453	0.10041	0.100371
0.03	0.999989	0.999987	0.999985	...	0.100453	0.10041	0.100371

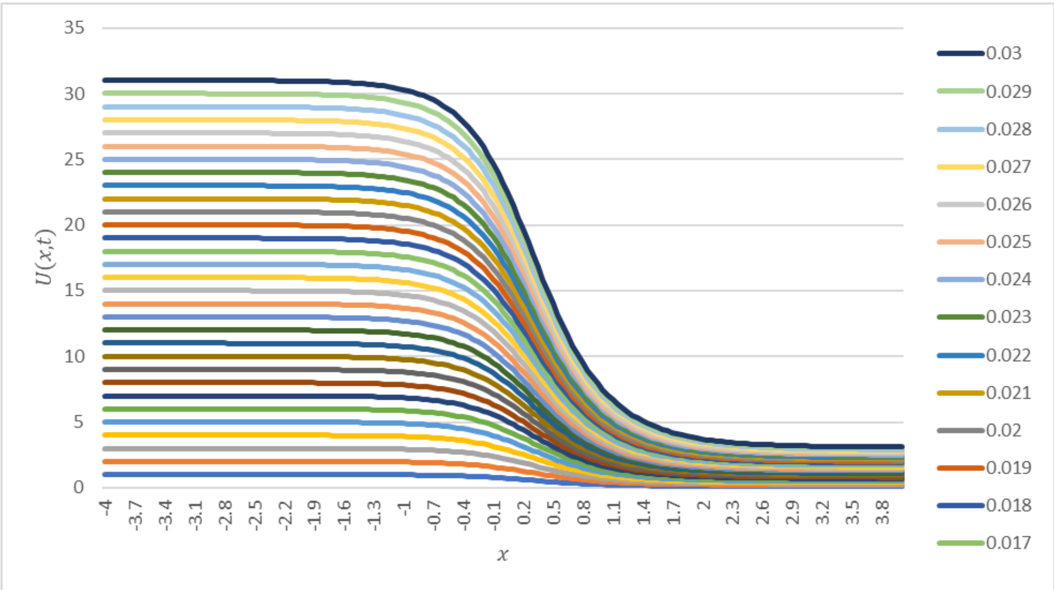

Figure 2 Explicit solutions for $\Delta x = 0.05$ and $\Delta t = 0.001$

Based on the above results, the explicit and analytical solutions are compared to evaluate the accuracy of the results at $t = 0.015$. The results are shown in Table 2.

Table 2: Abbreviated comparison of explicit and analytical solutions at $t = 0.015$

x	Explicit Numerical	Analytical	Absolute Error
-4	0.999988	0.999988	0
-3.95	0.999987	0.999987	0
-3.9	0.999985	0.999985	0
-3.85	0.999983	0.999983	0
...	...	...	...
3.8	0.100548	0.100548	0
3.85	0.100496	0.100496	0
3.9	0.100449	0.100449	0
3.95	0.100406	0.100406	0
4	0.100368	0.100368	0
	Average		0.000019

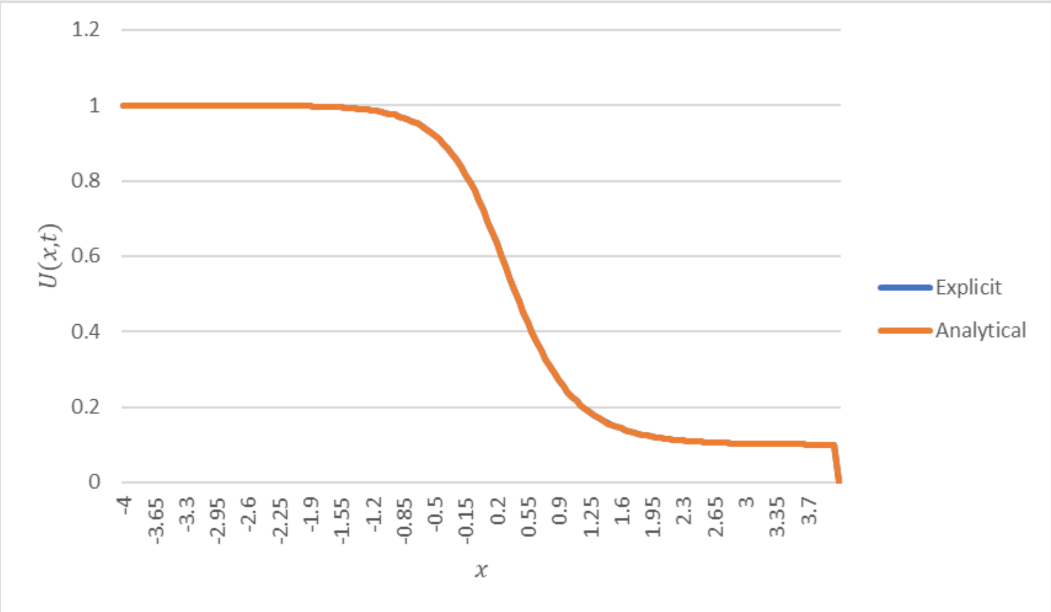

Figure 3 Comparison of explicit and analytical solutions at $t = 0.015$

Based on Table 2, the solutions of explicit are slightly different compared to analytical solutions with an average error of 0.000019. From the graph, the numerical curves have similar patterns to the analytical curve at $t = 0.015$. Therefore, it can be concluded that an explicit finite difference scheme is a suitable method to solve Burger's equation.

4.3. Numerical Results of Implicit Finite Difference Scheme

Using C++ software and the initial condition from equation (8) and the boundary conditions from equation (9), the numerical results of Burger's equation in equation (1) are computed implicitly.

Table 3: Abbreviated implicit solutions for $\Delta x = 0.05$ and $t = 0.001$

t/x	-4	-3.95	-3.9	...	3.9	3.95	4
0	0.999988	0.999986	0.999984	...	0.100445	0.100403	0.100364
0.001	0.999988	0.953631	0.997836	...	0.096815	0.100403	0.100365
0.002	0.999988	0.911026	0.993868	...	0.093442	0.100403	0.100365
0.003	0.999988	0.871811	0.988369	...	0.090302	0.100403	0.100365
0.004	0.999988	0.835664	0.981593	...	0.087376	0.100403	0.100365
0.005	0.999988	0.802294	0.973759	...	0.084645	0.100403	0.100366
...	...	...	...	...	...	...	...
0.026	0.999988	0.445493	0.753606	...	0.053104	0.100403	0.10037
0.027	0.999988	0.436988	0.74422	...	0.052288	0.100403	0.10037
0.028	0.999988	0.428868	0.735035	...	0.051507	0.100403	0.10037
0.029	0.999988	0.421107	0.72605	...	0.050757	0.100403	0.100371
0.03	0.999989	0.413683	0.717264	...	0.050038	0.100403	0.100371

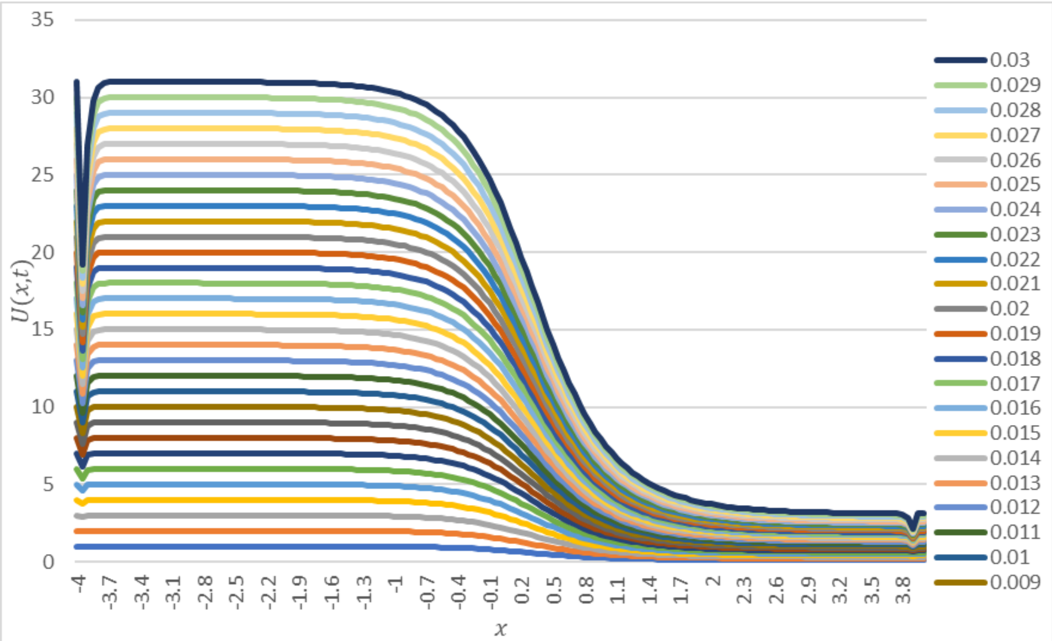

Figure 4 Implicit solutions for $\Delta x = 0.05$ and $\Delta t = 0.001$

The implicit solutions' accuracy is evaluated by comparing them with the analytical solutions at $t = 0.015$. The results are displayed in Table 4 and Figure 5.

Table 4: Abbreviated comparison of implicit and analytical solutions at $t = 0.015$

x	Implicit Numerical	Analytical	Absolute Error
-4	0.999988	0.999988	0
-3.95	0.575008	0.999987	0.424979
-3.9	0.868844	0.999985	0.131141
-3.85	0.968485	0.999983	0.031498
...	...	...	...
3.8	0.098969	0.100548	0.001579

3.85	0.091979	0.100496	0.008517
3.9	0.065126	0.100449	0.035323
3.95	0.100403	0.100406	0.000003
4	0.100368	0.100368	0
	Average		0.004008

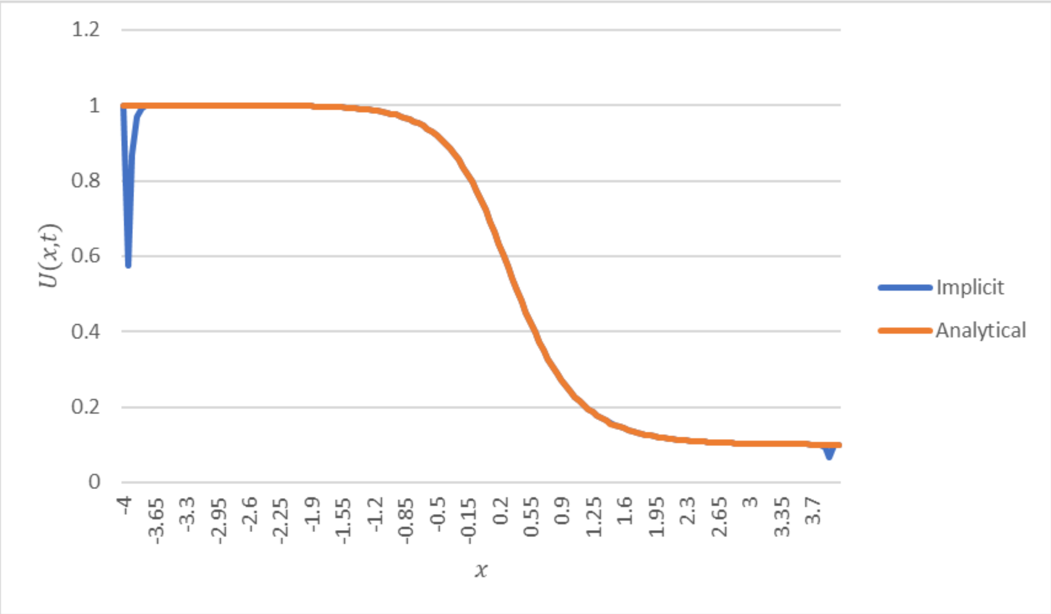

Figure 5 Comparison of implicit and analytical solutions at $t = 0.015$

Based on Table 4 and Figure 5, implicit solutions are slightly different from analytical solutions. The average error is 0.004008, suggesting a near correspondence with the analytical solutions. It is also clear that the patterns of the numerical and analytical curves are nearly identical. Consequently, it may be said that a good way to solve Burger's equation is to use an implicit finite difference scheme.

Conclusion

The study focuses on solving Burger's equation using explicit and implicit finite difference schemes, a method commonly used in studying nonlinear waves and fluid dynamics. The accuracy of the numerical results is evaluated by comparing them with analytical solutions. The results show that explicit and implicit finite difference schemes are suitable for solving Burger's equation due to their small absolute error. To achieve more accurate results, the grid size must be smaller.

Acknowledgment

The researcher would like to thank all the people who have supported this study, especially Dr. Shazirawati Bt Mohd Puzi for her guidance and constant supervision. A special gratitude to parents, for their kind cooperation and encouragement. Extend appreciation to friends who have encouraged and motivated me throughout this process. Lastly, I thank myself for not giving up throughout the process.

References

[1] Geeta, A., Shubham, M., Homan, E., & Masoumeh, K. (2023). "Numerical simulation and dynamics of Burger's equation using the modified cubic B-Spline differential quadrature method", *Discrete Dynamics in Nature and Society* volume 2023. https://www.hindawi.com/journals/ddns/2023/5102374/#B19.

[2] Bellman, R., Kashef, B.G., & Casti, J. (1972). "Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations", Journal of Computational Physics, pp. 40-52. https://www.sciencedirect.com/science/article/pii/0021991729087.

[3] Mohamed, N.A. (2019). "Solving one- and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes", *Arab Journal of Basic and Applied Science*, pp. 254-268. https://doi.org./10.1080/25765299.2019.1613746.

[4] Vijitha, M., & Ashish, A. (2018). "Numerical techniques for unsteady nonlinear Burger's equation based on backward differentiation formulas", Journal of Nonlinear Engineering. https://www.degruyter.com/document/doi/10.1515/nleng-20170068/html?lang=en#:~:text=with%20initial%20condition%20u(x,form%20of%20Navier%2DStoke's%20equation.