Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 15, 2023, page 150-157

Numerical Solution of One-Dimensional Heat Equation using Method of Lines

Nannthini Kumaragurubaran, Shazirawati Mohd Puzi
Department of Mathematical Sciences, Faculty of Science,
University Teknologi Malaysia
*Corresponding author: shazirawati@utm.my

Abstract
The one-dimensional heat equation is explored numerically in this project by applying the Method of Lines (MOL) with the iterative Euler's approach. The primary goals are to use the MOL technique to solve the heat equation, evaluate the findings' accuracy by comparing them to the exact solution, and simulate the computational processes using MATLAB software. The MOL method utilizes second-order central finite differences to discretize the equation into an ordinary differential equation. Based on stability requirements, the solutions, and errors for various time step sizes, t are evaluated using MATLAB. The results demonstrate that the MOL solution has smaller errors at $t = 1$ but is marginally lower than the analytical solution. Notably, the MOL solution closely resembles the analytical solution at $t = 0.2$. The accuracy of the numerical solution is improved by reducing the time step size t. The one-dimensional heat equation is numerically solved using the method lines approach in this work, which emphasizes stability and accuracy to generate results that closely resemble the analytical results.
Keywords: Heat equation; Method of lines (MOL); Partial Differential Equations; Euler's method; Ordinary Differential Equations (ODE)

Introduction

Partial differential equations (PDEs) involve derivatives of an unknown function with respect to multiple variables. They can be classified as hyperbolic, parabolic, or elliptic. Solving PDEs requires appropriate boundary and initial conditions. The heat equation, a parabolic PDE, describes temperature distribution over time. It can be solved analytically or numerically, with methods like the method of lines (MOL) and Euler's method. Numerical simulations are commonly implemented using software such as MATLAB. The heat equation finds applications in various scientific fields, and its solution accuracy can be evaluated by comparing with the exact solution. This research focuses on solving the one-dimensional heat equation using the MOL with Euler's method and analysing the results.

The focus of the study is to solve one-dimensional heat equation using method of lines by applying Euler's method, examine the accuracy of results obtained using numerical by comparing the exact solution and to stimulate the numerical computational of heat equation in MATLAB software. This paper is arranged as follows. In section 2 will be explained about partial differential equations, difference method, method of lines, system of ODE using Euler's method and one-dimensional heat equation and following with section 3 will present about the solving of the method of lines using Euler's method as well as stability criteria. Then the experimental analysis and the results obtained are discussed in section 4. Lastly section 5 contains conclusions and suggestions for future research work.

2. Literature Review

2.1. Partial Differential Equations (PDE)

Partial differential equations (PDEs) are mathematical equations that involve partial derivatives of a function with multiple independent variables. They can be categorized into three basic types: parabolic, hyperbolic, and elliptic. Parabolic PDEs, such as the heat diffusion equation, use first-order temporal derivatives and second-order spatial derivatives. Hyperbolic PDEs, like wave transformations and vibrations, involve second-order derivatives with opposite signs in time and space. Elliptic PDEs, such as the Laplace equation, describe steady-state problems with second-order derivatives. Subani et al.,

(2020) provides insights into these classifications. PDEs require appropriate boundary conditions to define behaviour at the domain boundary and initial conditions to determine the solution at the starting point or initial state. These conditions are crucial for establishing well-posed problems with unique solutions.

2.2 Difference Formula

Analytical solutions of partial differential equations provide us with closed-form expressions that depict the variation of the dependent variables in the domain. The values at discrete points in the domain known as grid points are provided by the numerical solutions, which are based on finite differences. A partial derivative will be replaced by an appropriate algebraic different quotient in a finite difference calculation. The majority of finite-difference derivative representations are built on Taylor's series expansions. Difference formulas can be obtained by extending the function in a Taylor series around the desired point and truncating at a specific order. For many derivatives, including higher-order derivatives, these formulas can be generated. Consider Figure 2.1 which shows a section of a discrete grid in the xy-plane. Assume that the spacing of the grid points in the x-direction is uniform and given by Δx and the spacing of the grid points in the y-direction is uniform and given by Δy.

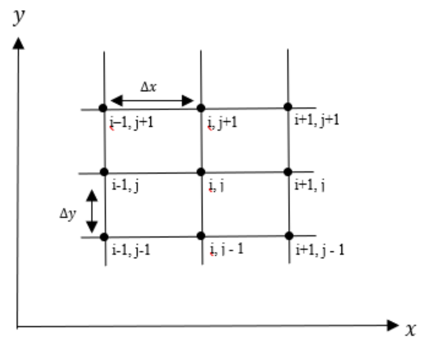

Figure 1 Finite Difference Grid

Based on Figure 1, the grid points are identified by an index i which runs in the x – direction and index j which runs in the y – direction. Therefore, the (i, j) mean index for point C in Figure 1, then the point to the right of C is labelled as $(i + 1, j)$, the point to the left is $(i - 1, j)$, the point directly above s $(i, j + 1)$ and the point directly below is $(i, j - 1)$. The finite difference method can be expressed by three ways in x direction. They are:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \begin{cases} \dfrac{u_{i,j+1} - u_{i,j}}{\Delta x} + O(\Delta x) & \text{Forward difference} \\ \dfrac{u_{i,j} - u_{i,j-1}}{\Delta x} + O(\Delta x) & \text{Backward difference} \\ \dfrac{u_{i,j+1} - u_{i,j-1}}{2\Delta x} + O(\Delta x)^2 & \text{Central difference} \end{cases} \quad (2.1)$$

In the above equation, $O(\Delta x)$ and $O(\Delta x)^2$ stand in for the truncation error. The more accurate findings will be attained the higher the order of the truncation error. In addition, the difference calculation itself includes an approximation that causes truncation error. It indicates a difference between the precise derivative and the approximative derivative.

2.3 Method of Lines (MOL)

The Method of Lines (MOL), as described by Hamdi et al., (2007), approximates partial differential equations (PDEs) by replacing the spatial derivatives with algebraic approximations. This transforms the PDE into a system of ordinary differential equations (ODEs) with only the time variable as the independent variable. Various techniques, such as finite element, finite volume, spectral, or meshless methods, can be employed to discretize the spatial dimensions in MOL, as mentioned by Kazem et al. (2017). Integration methods like Euler, Runge-Kutta, Adams-Bashforth, or Backward Difference Formula (BDF) can then be used to numerically solve the ODE system and approximate the original

PDE. MOL has been extensively utilized in various fields, including conservation laws, dispersive wave equations, biomedical sciences, and parabolic equations, as discussed in studies by Ahmad et al. (2001), Koto (2004), Kazem et al. (2017), Bratsos et al., (2007), Shakeri et al. (2008), and Hamdi et al. (2005).

2.4 System of Ordinary Differential Equation (ODE) Using Euler's Method

A system of ordinary differential equations consists of multiple ODEs that depend on the same independent variable. The variables in the system represent the unknown functions to be solved. Consider a system of n first-order differential equations needs be written as:

$$\begin{cases} x_1' = f_1(t, x_1, x_2, \ldots, x_n) \\ x_2' = f_2(t, x_1, x_2, \ldots, x_n) \\ \quad \vdots \\ x_n' = f_n(t, x_1, x_2, \ldots, x_n) \\ x_1(a) = s_1, x_2(a) = s_2, \ldots, x_n(a) = s_n, \quad \text{all given} \end{cases} \tag{2.2}$$

where t is the independent variable (often representing time) and $x_1, x_2, \ldots x_n$ are the dependent variables. Then, $x_1(a) = s_1, x_2(a) = s_2, \ldots, x_n(a) = s_n$ are the initial and boundary conditions. This form can be written in vector notation.

$$\begin{cases} X' = F(t, X) \quad \text{given} \\ X(a) = S, \end{cases} \tag{2.3}$$

Therefore, we can define the following n component vectors as,

$$\begin{cases} \boldsymbol{X} = [x_1, x_2, \ldots, x_n]^T \\ \boldsymbol{X}' = [x_1', x_2', \ldots, x_n']^T \\ \boldsymbol{F} = [f_1, f_2, \ldots, f_n]^T \\ \boldsymbol{X}(a) = [x_1(a), x_2(a), \ldots, x_n(a)]^T \end{cases} \tag{2.4}$$

The Euler's method formula can be taken from the Taylor series method of order m which is

$$\boldsymbol{X}(t+h) = \boldsymbol{X} + h\boldsymbol{X}' + \frac{h^2}{2}\boldsymbol{X}'' + \cdots + \frac{h^m}{m!}\boldsymbol{X}^{(m)} \tag{2.5}$$

Since we only need first order of Euler's method, therefore,

$$\boldsymbol{X}(t+h) = \boldsymbol{X} + h\boldsymbol{X}' \tag{2.6}$$

where, $\boldsymbol{X} = \boldsymbol{X}(t), \boldsymbol{X}' = \boldsymbol{X}'(t)$ and h is the step size.

2.5 One-Dimensional Heat Equation

The heat equation is a fundamental tool utilized in various scientific fields, as highlighted by Mamun et al. (2018). It is considered the standard parabolic partial differential equation in mathematics and finds applications in fields such as physics, engineering, probability theory, and financial mathematics. The Fokker-Planck equation connects the study of Brownian motion to the heat equation, while in financial mathematics, it is employed to solve the Black-Scholes partial differential equation. Additionally, the heat equation is crucial for investigating thermal phenomena, including heat transport in solid objects, fluid behaviour, temperature distribution in electronic devices, phase change, and solidification. Understanding the heat equation enables researchers to predict and optimize temperature profiles and heat transfer rates in various systems (Mamun et al., 2018).

Despite claims by Subani et al. (2020) that the heat equation propagates energy at an unlimited speed, which is impossible, the validity of the heat equation as a model for temperature evolution remains strong for classical applications in engineering and physics. Kazem et al. (2017) and Mamun et al. (2018) emphasize that the heat equation, a parabolic partial differential equation, describes how heat or temperature variation is distributed throughout regions over time. It captures the diffusive process where heat spreads and evens out within a system, governing the flow of thermal energy from regions of higher temperature to lower temperature until reaching an equilibrium state. Derived from the principles of conservation of energy and Fourier's law of heat conduction, the heat equation is also known as the diffusion equation. The standard form of the one-dimensional heat equation provides insights into how temperature distribution changes over time in a system due to heat conduction (Kazem et al., 2017; Mamun et al., 2018). The standard form of the one-dimensional heat equation is described how the temperature distribution within a system change over time due to heat conduction:

$$\frac{\partial u}{\partial t} - K\frac{\partial^2 u}{\partial x^2} = 0 \quad , \quad a \le x \le b \ , \ t \ge 0 \tag{2.7}$$

where, the temperature distribution, $u(x,t)$, is represented as a function of position (x) and time (t) by the partial differential equation provided. The equation illustrates how the temperature changes over time in a one-dimensional domain that is enclosed by the boundaries of a and b. The term $\frac{\partial u}{\partial t}$ represents the rate of change of temperature with respect to time, while $\frac{\partial^2 u}{\partial x^2}$ represents the curvature of the temperature distribution with respect to position. The material properties are defined by the coefficient K, sometimes referred to as thermal diffusivity, which also controls how heat diffuses inside the system. According to the equation, the second derivative of temperature with respect to position and the proportionality constant, K, determine the rate of change of temperature over time.

3. Methodology

3.1 Discretization of Linear Equation into Finite Difference Form

One-dimensional heat equation as stated in equation (2.2) can be solved by using MOL. The first step is to discretize the linear equation into algebraic form by using central finite difference. The derivative in (2.2) can be computed by finite difference scheme which is second order central finite difference.

3.1.1 Derivation of First Order Derivative with Second Order Solution

The Finite difference method is used to replace the partial derivative with a suitable algebraic different quotient. The most common finite representations of derivatives are based on Taylor's series expansions. The equation (3.1) below is Taylor's series expansion at point (i,j) based on Figure 2.1:

$$u_{i+1,j} = u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{2!} + \cdots \tag{3.1}$$

The finite difference expansion in equation (3.1) comes from the information in the right of grid point (i,j), based on Figure 2.1 by using $u_{i+1,j}$ and $u_{i,j}$. As a result, the finite difference in equation (3.1) is called forward difference method. According to Figure 2.1, information from the left of grid point (i,j) is applied to construct the backward finite difference using $u_{i-1,j}$ and $u_{i,j}$ as written in equation (3.2).

$$u_{i-1,j} = u_{i,j} - \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{2!} - \cdots \tag{3.2}$$

The first order accuracy is insufficient for the majority of applications in real problems. Therefore, the central difference form is generated by simply subtracted the equation (3.1) and equation (3.2):

$$u_{i+1,j} - u_{i-1,j} = 2\left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x + \cdots \tag{3.3}$$

This equation (3.3) can be simplified into:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + O(\Delta x)^2 \tag{3.4}$$

where, $O(\Delta x)^2$ is represent truncation error. Hence equation (3.4) is called second order solutions of central finite difference.

3.1.2 Derivation of Second order Derivative with Second Order Solutions

The central finite difference form is generated by simply added equation of (3.1) and (3.2):

$$u_{i+1,j} + u_{i-1,j} = 2u_{i,j} + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} (\Delta x)^2 + \cdots \tag{3.5}$$

This equation (3.5) can be simplified into:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 \tag{3.6}$$

The term $O(\Delta x)^2$ in equation (3.6) is represent truncation error. Therefore, equation (3.6) is called second order solutions of central finite difference.

3.2 Discretization of One-Dimensional Heat Equation using MOL

In MOL, the spatial derivative, $\frac{\partial^2 u}{\partial x^2}$ stated in (2.2) must be discretized using finite difference formula. In section 3.2, second order solutions of central finite difference are derived. Thus, the result is substituted into equation (2.2) which produce the following equations:

$$\frac{\partial u_i}{\partial t} = K\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}\right) = 0 \ , \qquad i = 1,2,3, \dots N. \tag{3.7}$$

Let $\Delta x = h$, thus we can write (3.7) as

$$\frac{\partial u_i}{\partial t} = K\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right) = 0 \ , \qquad i = 1,2,3, \dots N. \tag{3.8}$$

Based on the equation (3.8), the system of difference equations of one independent variable t is formed. This system will be solved by using iterative method which is Euler's method.

3.3 The Euler's Method

The Euler's method is used to solve the linear ODE produced in equation (3.8). The Euler's method is a basic numerical method for solving ODEs. The Euler's method formula is given by:

$$U^{k+1} = U^k + \Delta t\, F(U^k) \tag{3.9}$$

where $F(U^k)$ is the expression at right hand side of the equation (3.8) and

$$\Delta t = \frac{(t_n - t_0)}{number\ of\ grid\ size} \tag{3.10}$$

3.4 Stability Criteria

3.4.1 Courant-Friedrichs-Lewy (CFL) condition

The Courant-Friedrichs-Lewy (CFL) condition is a rule used in numerical methods for solving equations that change over time. It prevents numerical errors and instability by making sure that the time step size used in the calculations is small enough. The condition says that the time step should be shorter than the time it takes for information to travel across a small distance in the problem being solved. This distance is determined by the grid spacing and the properties of the material. By following the CFL condition, we ensure that our numerical solution remains stable and accurate. In this study, Euler method is used to numerically solve the resulting ODE system, the CFL condition can be expressed as:

$$\Delta t = \frac{\Delta x}{2 \propto} \tag{3.11}$$

where Δx is the grid spacing in the x direction, α is the thermal diffusivity of the material, and Δt is the time step.

3.5 Performance

a) Relative Error, RE:

The relative error is a measure of the difference between the numerical solution and the exact solution, expressed as a percentage or a decimal. It quantifies the accuracy of the numerical method by comparing it to the known exact solution. The formula for calculating the relative error is:

$$RE = \left|\frac{u_{numerical} - u_{exact}}{u_{exact}}\right| \times 100\% \tag{3.12}$$

b) Euclidean Norm (L_2Norm):

The Euclidean norm, also known as the L_2 norm, is a measure of the magnitude or length of a vector. In the context of comparing numerical solutions, the Euclidean norm is used to compute the difference between the numerical solution and the exact solution. It provides a measure of the overall error or discrepancy between the two solutions. The formula for calculating the Euclidean norm is:

$$L_2 = \sqrt{\sum (u_{numerical} - u_{exact})^2} \tag{3.13}$$

c) Maximum norm (L_∞ norm):

The Maximum norm is also known as the infinity norm or supremum norm, is the absolute difference is taken at each data point, and the maximum value is selected. In other words, it represents the largest value among the entries of data. The formula for calculating Maximum norm is:

$$L_\infty = \max\left(|u_{numerical} - u_{exact}|\right) \tag{3.14}$$

4. Result and Discussions

4.1 Experimental Setting

Let consider, the thermal diffusivity $K = 0.5$ with the initial condition $u(x, 0) = \sin \pi x$ where the interval of x is assumed as $0 \le x \le 1$ and the interval of t is assumed as $0 \le t \le 1$ and the Dirichlet boundary conditions are $u(0, t) = 0$ and $u(1, t) = 0$.This experiment is conducted using MATLAB to get the numerical solutions while Microsoft Excel is used for diagram illustration. By taking the spatial grid size, $N = 20$ and the temporal grid size, $T = 400$, therefore the step size, $\Delta x = \frac{b-a}{N} = \frac{1-0}{20} = 0.05$ and the step size $\Delta t = \frac{c-d}{T} = \frac{1-0}{400} = 0.0025$.

4.2 Comparison between MOL and Analytical Solution at various time

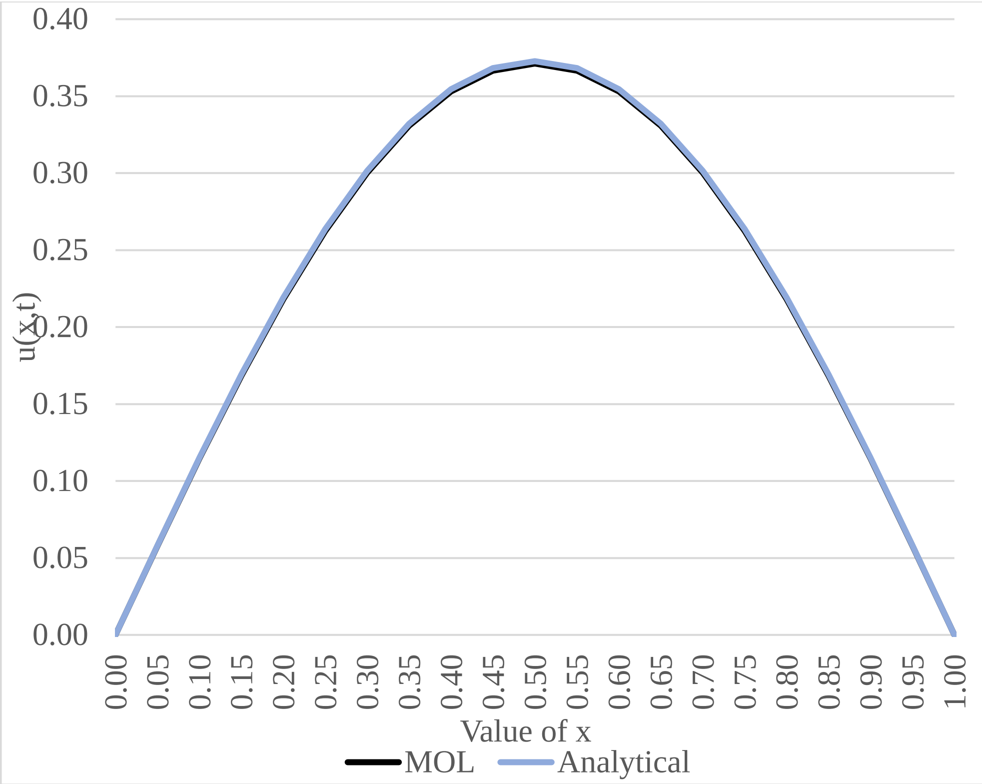

Figure 2 Comparison between MOL and Analytical Solution at $t = 0.2$

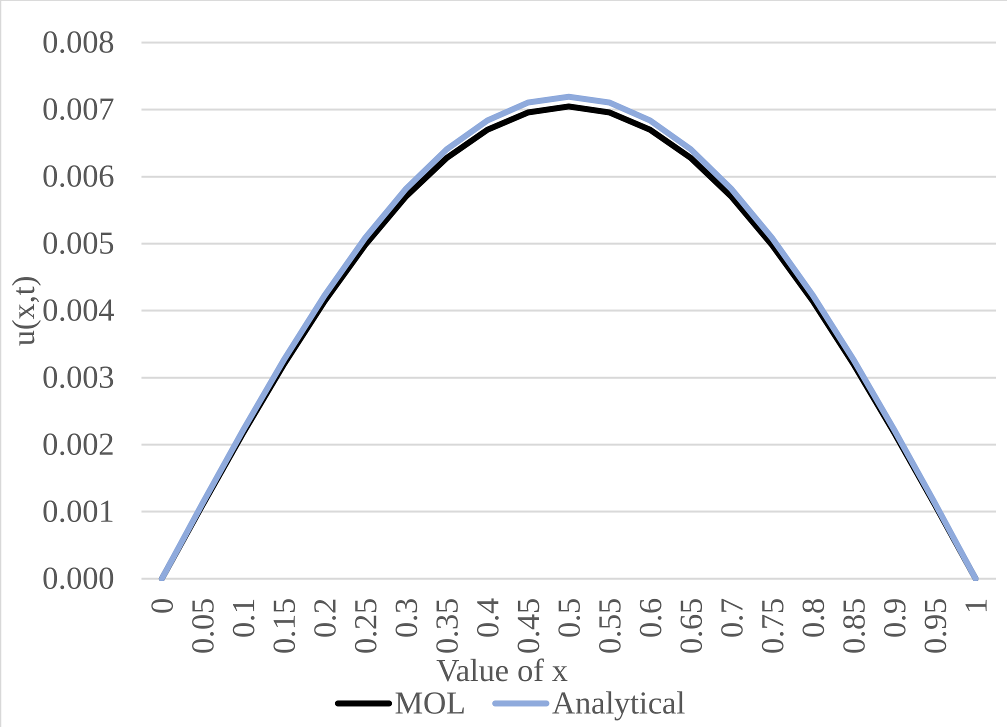

Figure 3 Comparison between MOL and Analytical Solution at $t = 1$

Table 1: Absolute error at various time, t

t	Mean Absolute Error
0.2	0.00091947
0.4	0.00068399
0.6	0.00038161
0.8	0.00018925
1.0	0.00008799

Based on the Table 1 and Figure 2 and 3, the mean absolute error at $t = 0.2$ is 0.00091947 while $t = 1$ is 0.00008799 which indicates that it has 1×10^{-4} and 1×10^{-5} error value respectively. Other than that, at $t = 0.2$ the MOL solution is closely matched to the analytical solution which means that the MOL method accurately captures the behavior of the system at the time point while at $t = 1$, the MOL solution is lower than the analytical solution, indicating a small deviation, however the absolute error is decreasing. This means that although the MOL solution may not exactly match with the analytical solution at $t = 1$ but it is getting closer to it. From those observations, we can conclude that the MOL method shows good agreement with the analytical solution at $t = 0.1$ and exhibits a decreasing absolute error as time progress. Based on the graph, it clearly shows that the numerical curve looks like the analytical curve. Therefore, MOL is a suitable method to solve one-dimensional heat equation.

4.3 Effect of Step Size of t to MOL Solutions

Several experiments have been conducted to obtain the numerical and analytical results for different value of step size t. There are three values of step size are used which are,

$$\Delta t = 0.002, \quad \Delta t = 0.0015, \quad \text{and} \quad \Delta t = 0.001$$

The step size of x is fixed at $\Delta x = 0.05$. Then, the analytical, numerical solutions, L_2 norm and L_∞ norm are generated using MATLAB.

4.3.1 Comparison of MOL and Analytical Solutions in various Δt what $t = 0.6$

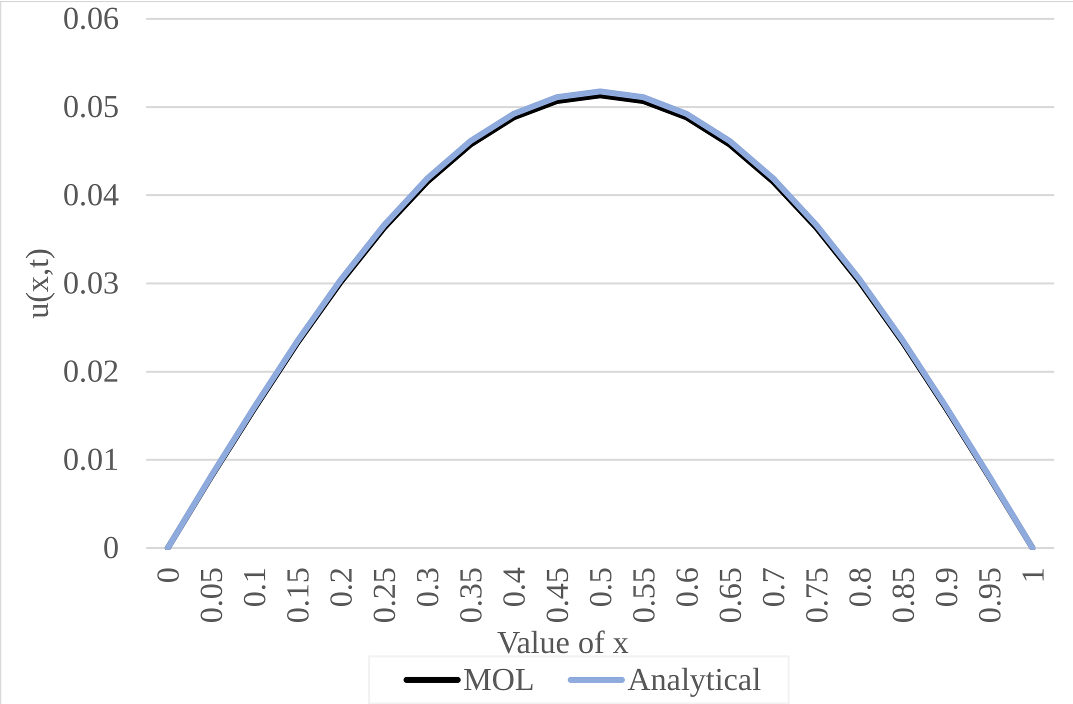

Figure 4 Comparison of Numerical and Analytical when $\Delta t = 0.002$

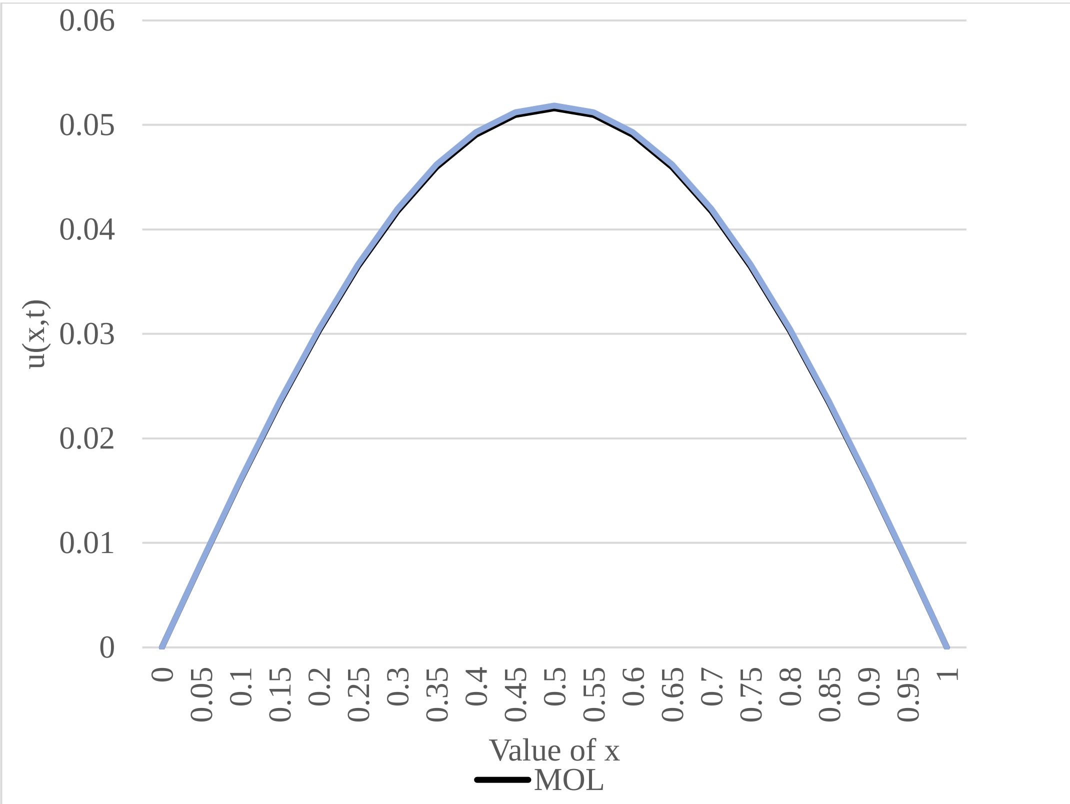

Figure 5 Comparison of Numerical and Analytical when $\Delta t = 0.0015$

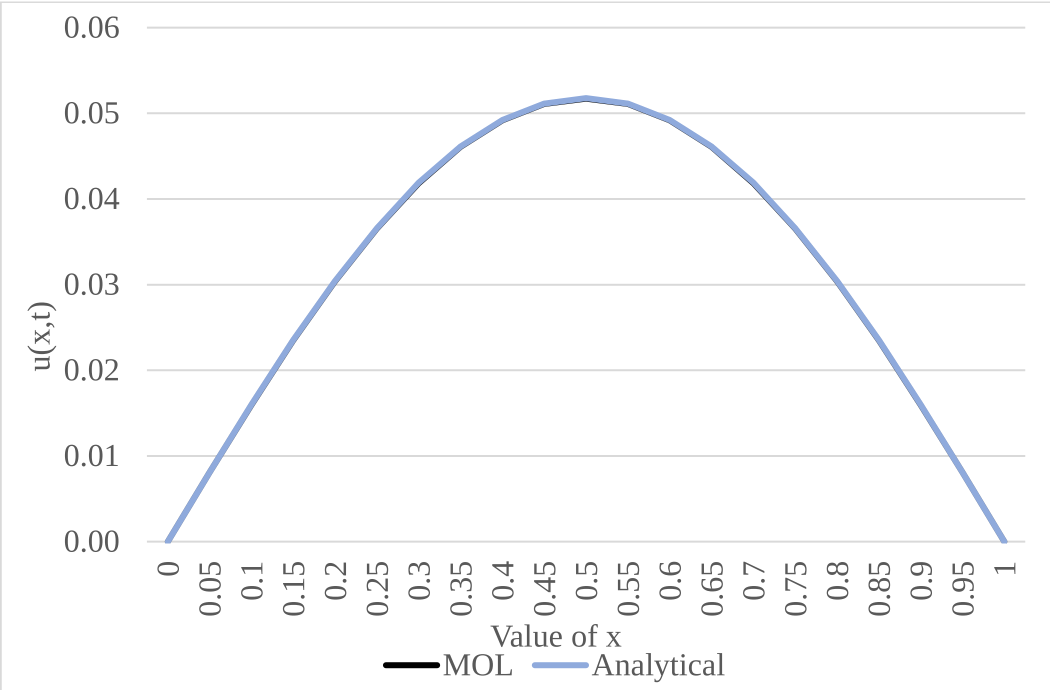

Figure 6 Comparison of Numerical and Analytical when $\Delta t = 0.001$

Table 2 L_2 and L_∞ for various Δt

Δt	L_2	L_∞
0.002	0.00044700	0.00106300
0.0015	0.00025500	0.00060600
0.001	0.00006400	0.00015100

Based on the tables and figures above, the purpose of comparison of MOL and analytical solutions is to provide the accuracy of both simulations in MATLAB. According to the Table 2 shows the error is decreasing when Δt is getting smaller. In terms of graph representations in Figure 4 – Figure 6 indicates that when the Δt getting smaller at $t = 0.6$, the MOL solution will getting closer to the analytical solution and finally it will converge to the analytical solution. Therefore, we conclude that the smaller the time step size of t, the more accurate solution will be generated.

A drawback we will face when working with simulations that is need a lot of processing. For instance, by decreasing the time step size (Δt) to achieve higher accuracy comes at the cost of increased computational resources. This means that smaller time step sizes may require longer computation times and more memory usage, especially for complex systems or larger problem domains.

Conclusion

This research focused on studying partial differential equations (PDEs), specifically the one-dimensional heat equation, which has applications in various linear problems. The objective was to develop a numerical method using the method of lines (MOL) to solve the heat equation. The study assessed the stability and accuracy of the MOL by comparing numerical and analytical solutions and analysing the effect of time step size. The results showed that the MOL was effective, with small absolute errors and improved stability with smaller time step sizes. Future research recommendations include exploring other linearization methods, extending the study to higher dimensions, investigating nonlinear heat equations, exploring different boundary conditions, and using alternative software like C++ or Python. Overall, the research achieved its objectives and provided insights into solving the heat equation using the MOL.

Acknowledgements
The researcher would like to thank all 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 me up throughout the process.

References
Ahmad, I., & Berzins, M. (2001, May). MOL solvers for hyperbolic PDEs with source terms. *Mathematics and Computers in Simulation*, *56*(2), 115–125.
Bratsos, A. G. (2007). The solution of the two-dimensional sine-Gordon equation using the method of lines. *Journal of Computational and Applied Mathematics*, *206*(1), 251- 277.
Hamdi, S., Enright, W. H., Ouellet, Y., & Schiesser, W. E. (2005). Method of lines solutions of the extended Boussinesq equations. *Journal of Computational and Applied Mathematics*, *183*(2), 327-342.
Hamdi, S., Schiesser, W., & Griffiths, G. (2007). Method of lines. *Scholarpedia*, *2*(7), 2859.
Kazem, S., & Dehghan, M. (2017, October 25). Application of finite difference method of lines on the heat equation. *Numerical Methods for Partial Differential Equations*, *34*(2), 626–660.
Koto, T. (2004). Method of lines approximations of delay differential equations. *Computers & Mathematics with Applications*, *48*(1-2), 45-59.
Mamun, A. A., Ali, M. S., & Miah, M. M. (2018). A study on an analytic solution 1D heat equation of a parabolic partial differential equation and implement in computer programming. *Int. J. Sci. Eng. Res*, *9*(9), 913-921.
Subani, N., Jamaluddin, F., Mohamed, M. A., & Badrolhisam, A. D. (2020). Analytical solution of homogeneous one-dimensional heat equation with Neumann boundary conditions. *Journal of Physics: Conference Series*, *1551*(1), 012002.
Shakeri, F., & Dehghan, M. (2008). The method of lines for solution of the one-dimensional wave equation subject to an integral conservation condition. *Computers & Mathematics with Applications*, *56*(9), 2175-2188.