Volume 8 No 2 December (2025)

e-ISSN: 2615-3270
p-ISSN: 2615-3599

Eigen Mathematics Journal

homepage: https://eigen.unram.ac.id/index.php/eigen

# Simulation of Spring Oscillations in Second-Order Differential Equations Using the Finite Difference Method

**Muhammad Imam Al Paqih[1], Rida Alkausar Hardi[1], Nuzla Af'idatur Robbaniyyah[1*]**

[1] *Department of Mathematics, Universitas Mataram, Indonesia*
[*] *Corresponding author: nuzla@unram.ac.id*

## A B S T R A C T

This study aims to simulate the motion of a damped spring oscillation, modeled by a second-order ordinary differential equation, using the Finite Difference Method (FDM). The main focus is on implementing the central finite difference scheme to discretize the equation, deriving an explicit iterative formula, and analyzing the oscillation dynamics and the accuracy of the numerical solution. The simulation was conducted with specific parameters (mass $m$ = 1.0 kg, spring constant $k$ = 10.0 N/m, damping coefficient $c$ = 0.5 Ns/m) and various time steps ($\Delta t$ = 0.5 s, 0.1 s, 0.01 s). The simulation results qualitatively show damped oscillatory behavior consistent with physical theory, where the amplitude decreases over time. The accuracy of the numerical solution, measured by the Symmetric Mean Absolute Percentage Error (SMAPE) against the analytical solution, was significantly influenced by $\Delta t$; the smallest time step (0.01 s) yielded the highest accuracy with a SMAPE of 0.4495%. The Finite Difference Method proved effective in analyzing the spring oscillation system, demonstrating that the proper selection of $\Delta t$ is crucial for balancing accuracy and computational efficiency.

**Keywords:** Discretization Schemes, Iterative Formula, Error Analysis, Computational Modelling

Received : 24-07-2025;
Revised : 16-11-2025;
Accepted : 03-12-2025;
Published : 17-12-2025;

DOI: https://doi.org/10.29303/emj.v8i2.321



## 1. Introduction

Differential equations play a fundamental role in modeling various physical phenomena and problems in engineering [1]. The ability of differential equations to describe the changes and dynamics of a system makes them an invaluable tool in modern science. One common and important physical phenomenon to be studied is oscillatory motion, which is a characteristic solution of many systems of

differential equations [2]. Oscillatory systems, such as pendulum motion [3], or vibrations in various structures like a beam on an inclined plane [4], can be found in various contexts. Complex wave phenomena also exemplify the application of differential equations in physics [5].

Specifically, spring oscillatory motion is one of the fundamental models in physics, often represented by a second-order ordinary differential equation (ODE) [6]. A thorough understanding of the dynamics of spring oscillation is crucial, not only from a theoretical standpoint but also for practical applications in the design of mechanical systems and vibration analysis.

Although the ideal spring oscillation model can be solved analytically to obtain an exact solution, many real-world oscillatory systems involve more complex factors such as the presence of damping [2]. Systems with time-varying mass also pose particular challenges for analytical solutions [7]. Nonlinear forms of differential equations also often make the search for an exact solution very difficult or even impossible [8]. In such situations, numerical approaches become a very important and popular alternative for obtaining approximate solutions with an acceptable level of accuracy [1].

The Finite Difference Method is one of the numerical techniques that has proven effective and is widely used to solve various types of differential equations [1]. The fundamental principle of the Finite Difference Method is to discretize the continuous domain into a finite number of grid points. Subsequently, the derivatives present in the differential equation are approximated using difference schemes, which are generally derived from Taylor series expansions [6]. The finite difference approach, particularly the central difference scheme, has also shown good results in approximating solutions of higher-order ODEs [8].

Various previous studies have explored the use of numerical methods for the analysis of oscillatory systems. Among them, the Runge-Kutta method has been used for oscillators with varying mass [7]. Other research employed the Adams-Bashforth Moulton method to analyze vertical and torsional oscillations in suspension bridge structures [9]. Furthermore, k-step block methods have also been applied to solve second-order Ordinary Differential Equations (ODEs) [6]. Additionally, the finite difference method has been specifically applied to solve the Korteweg-de Vries (KdV) equation [10]. Each of these numerical methods offers different levels of accuracy and computational complexity.

Based on this background, this research aims to perform a numerical simulation of the oscillatory motion of a spring, modeled by a second-order differential equation, using the finite difference method. The primary focus of this study is to implement the finite difference scheme for analyzing the dynamics of spring oscillation and to evaluate the behavior of the resulting numerical solution. The simulation results are expected to provide a better understanding of the application of the finite difference method to the spring oscillation system.

## 2. Research Methods

The methodology of this research follows the systematic steps in the application of the finite difference method to solve the second-order differential equation in the case of spring oscillation. Figure 1 shows a flowchart that represents the entire research process from the initial stage to the analysis of the results.

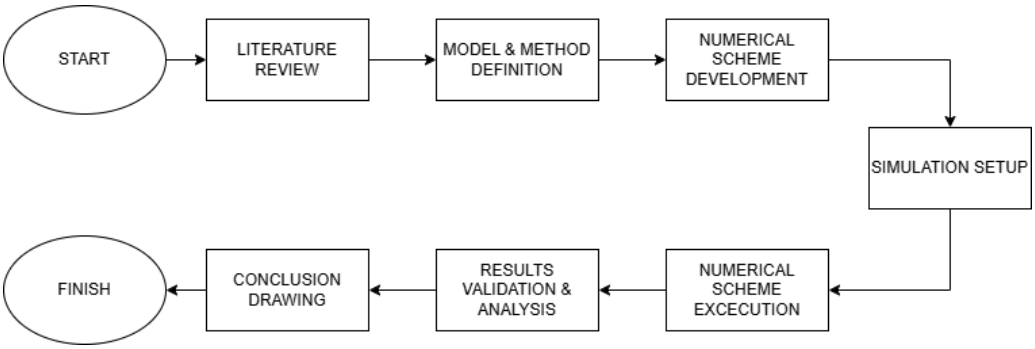


**Figure 1.** Flowchart Visualisasi Osilasi Pegas

### *2.1. Model and Method Definitions*

Focuses on homogeneous second-order ODE (without external force, $F(t) = 0$) which describes a damped harmonic oscillator with the following equation [11]:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0 \tag{1}$$

- $m$ is the mass of the object (kg)
- $c$ is the damping coefficient (Ns/m)
- $k$ is the spring constant (N/m)
- $x(t)$ is the deviation of the mass from the equilibrium position at time $t$ (m)
- $t$ is the time (s)

equation 1 above can be solved using numerical methods, one of the numerical methods that can be used is the Finite Difference Method. In this method, discretization of the time domain is required. Suppose by dividing the total time interval $T$ to $N$ equal subintervals, with a time step length of $\Delta t = T/N$. The time at each discrete step is $t_i = i\Delta t$, where $i = 0, 1, 2, \ldots, N$. Deviation in time $t_i$ is denoted as$x_i = x(t_i)$.

### *2.2. Numerical Schema Development*

Once the fundamental model and method are established, the next step is the development of a specific numerical scheme. This is the core of the process of translating continuous differential equations into a discrete form that can be processed computationally. This process begins with the discretization of the continuous time domain into a series of discrete time points with a time step interval of ($\Delta t$) which is uniform. The derivatives contained in the ODE (first derivative for velocity and second derivative for acceleration) are then approximated using a finite difference scheme, such as center difference for better accuracy. These approximated derivatives are then substituted back into the Governing Differential Equation of the model. Through a series of algebraic manipulations, the resulting equations are rearranged to derive an explicit iterative formula (as illustrated in Equation 6 in the source study) that allows the calculation of the spring position value at the next time step. ($x_{i+1}$) based on the value at the current time step ($x_i$) and before($x_{i-1}$).

### *2.3. Simulation Setup*

Before the main computational process is executed, the preparation or setup stage of the simulation plays an important role to ensure that all conditions and parameters are properly defined. This step includes determining the initial conditions of the system, i.e, the initial position value.($x_0$) and initial

velocity($v_0$) of the spring on $t = 0$. Since the commonly used iterative formulas require two prior values, the estimate for the position value at the first time step is ($x_i$) is also necessary, often using approaches such as Taylor series expansion based on initial conditions. In addition, all the physical parameters relevant to the model, such as the mass of the spring ($m$), spring constant ($k$), and damping coefficient ($c$), as well as simulation control parameters, including the total simulation time duration ($T$) and time step size ($\Delta t$), carefully inputted.

### *2.4. Numerical Simulation Execution*

With all parameters and initial conditions prepared, the numerical simulation execution phase can begin. In this phase, the iterative formula that has been developed in the numerical scheme development stage is applied iteratively. This computational process runs step by step, calculating displacement of the spring ($x_i$) at each discrete time point ($t_i$) from the beginning to the end of the predefined simulation duration. Each iteration uses the results from one or two previous time steps to generate a solution at the current time step, effectively tracking the evolution of the spring oscillation movement over time based on the given mathematical model and parameters.

### *2.5. Validation and Analysis of Results*

Once a numerical solution for the oscillatory motion of a spring is obtained, the next crucial step is the validation of the accuracy and in-depth analysis of the results. For cases where an analytical or exact solution of the ODE can be determined (for example, in simple linear systems), this solution is used as a benchmark standard. A comparison between the numerical solution and the analytical solution is performed, and the error rate or error is calculated using an appropriate metric, such as Symmetric Mean Absolute Percentage Error (SMAPE), to quantify the deviation. The simulation results, both numerical and analytical, are then visualized in the form of graphical plots of position against time. This visualization facilitates qualitative interpretation of the system behavior and direct comparison between the two solutions. Further analysis focused on understanding the dynamics of the resulting oscillations, including investigation of the influence of parameters such as the size of the time step.($\Delta t$) on the accuracy and stability of numerical solutions.

Absolute error is defined as the difference between the analytic value (or true value) $a$ and numerical values $b$. SMAPE is used to avoid potential miscalculations when the actual value approaches zero or is even equal to zero. The SMAPE value is calculated using the following formula [12]:

$$SMAPE = \frac{1}{n}\Sigma_{t=1}^{n}\frac{|a-b|}{(a-b)/2} \tag{2}$$

SMAPE is divided into 4 criteria, which can be seen in the Table 1 [13].

**Table 1.** Criteria of SMAPE

| No | SMAPE Value | Description |
|---|---|---|
| 1 | < 10% | Excellent |
| 2 | 10% – 20% | Good |
| 3 | 21% – 50% | Fair |
| 4 | > 50% | Poor |

### *2.6. Inference*

The final stage of this research is drawing conclusions. This section presents the main results of the numerical simulations, validation of the analytical solution, and a brief summary of the system analysis. The effectiveness of the finite difference method in simulating spring oscillations and its accuracy are also covered in this section. In addition, suggestions for further development or future research are outlined here.

## 3. Result and Discussion

### *3.1. Discretization by Finite Difference Method*

In this spring oscillation simulation, the ordinary differential equation governing the system is solved by approximating the derivatives in the differential equation using the finite difference method, which can theoretically be derived from the Taylor series expansion. By replacing the exact derivative with this finite difference approximation, the value of the spring position ($x$) at discrete points in time can be estimated. ($t_i$) iteratively throughout the simulation [14].

With the Finite Difference Method, the derivative in the ODE is replaced with a finite difference approximation. For second-order ODE, the central finite difference approximation is used because it provides a better order of accuracy (usually of the order of $O(\Delta t^2)$) versus forward or backward difference for the second derivative.

First Derivative Centered finite difference approximation for the first derivative $x$ with respect to $t$ at the point $t_i$ is:

$$x_{i+1} = x + \Delta t \frac{dx}{dt} + \frac{(\Delta t)^2}{2!}\frac{d^2x}{dt^2}$$

$$x_{i-1} = x - \Delta t \frac{dx}{dt} + \frac{(\Delta t)^2}{2!}\frac{d^2x}{dt^2}$$

then the two equations above are subtracted.

$$x_{i+1} - x_{i-1} = 2\Delta t \frac{dx}{dt}$$

$$\frac{dx}{dt} = \frac{x_{i+1} - x_{i-1}}{2\Delta t}$$

so the first derivative can be written as.

$$\left.\frac{dx}{dt}\right|_{t=t_i} = \frac{x_{i+1} - x_{i-1}}{2\Delta t} \tag{3}$$

Second Derivative The center finite difference approximation for the second derivative of $x$ with respect to $t$ at point $t_i$ is:

$$x_{i+1} = x + \Delta t \frac{dx}{dt} + \frac{(\Delta t)^2}{2!}\frac{d^2x}{dt}$$

$$x_{i-1} = x - \Delta t \frac{dx}{dt} + \frac{(\Delta t)^2}{2!}\frac{d^2x}{dt}$$

then the two equations above are summed up.

$$x_{i+1} + x_{i-1} = 2x + 2\frac{(\Delta t)^2}{2!}\frac{d^2x}{dt}$$

$$x_{i+1} + x_{i-1} = 2x + (\Delta t)^2\frac{d^2x}{dt}$$

$$\frac{d^2x}{dt^2} = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2}$$

so the second derivative can be written as.

$$\left.\frac{d^2x}{dt^2}\right|_{t=t_i} = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2} \tag{4}$$

### 3.2. Numerical Scheme

Substitute the finite difference approximation (4) and (3) into the ODE model (1):

$$m\left[\frac{x_{i+1}-2x_i+x_{i-1}}{(\Delta t)^2}\right]+c\left[\frac{x_{i+1}-x_{i-1}}{2\Delta t}\right]+kx_i=0 \tag{5}$$

It then finds $x_{i+1}$ (the position at the next time step) based on the values at the current ($x_i$) and previous ($x_{i-1}$) time steps. he above equation is rearranged to isolate $x_{i+1}$. First, multiply the entire equation by $(\Delta t)^2$:

$$m(x_{i+1}-2x_i+x_{i-1})+\frac{c\Delta t}{2}(x_{i+1}-x_{i-1})+k(\Delta t)^2x_i=0$$

Group the terms that contain $x_{i+1}$, $x_i$, and $x_{i-1}$:

$$\left(m+\frac{c\Delta t}{2}\right)x_{i+1}+\left(k(\Delta t)^2-2m\right)x_i+\left(m-\frac{c\Delta t}{2}\right)x_{i-1}=0$$

Move tribe $x_i$ and $x_{i-1}$ to the right segment:

$$\left(m+\frac{c\Delta t}{2}\right)x_{i+1}=-\left(k(\Delta t)^2-2m\right)x_i-\left(m-\frac{c\Delta t}{2}\right)x_{i-1}$$

$$\left(m+\frac{c\Delta t}{2}\right)x_{i+1}=\left(2m-k(\Delta t)^2\right)x_i+\left(\frac{c\Delta t}{2}-m\right)x_{i-1}$$

Thus, the explicit iterative formula for $x_{i+1}$ is:

$$x_{i+1}=\frac{\left(2m-k(\Delta t)^2\right)x_i+\left(\frac{c\Delta t}{2}-m\right)x_{i-1}}{m+\frac{c\Delta t}{2}} \tag{6}$$

This formula applies to $i=1,2,\ldots,N-1$.

### 3.3. Initial Condition

To start the iteration, two initial values are required: $x_0$ and $x_1$. ODE initial conditions are usually given in the form of initial positions $x(0)=x_0$ and initial velocity $v(0)=dx/dt(0)=v_0$. $x_0$ already known. To get $x_1$, special treatment is required. A common approach is to use Taylor expansions or information from ODE itself on the $t=0$. Using the second-order Taylor expansion around $t=0$:

$$x(\Delta t)\approx x(0)+\left.\frac{dx}{dt}\right|_{t=0}\Delta t+\frac{1}{2}\left.\frac{d^2x}{dt^2}\right|_{t=0}(\Delta t)^2$$

Suppose $x(0)=x_0$ and $dx/dt(0)=v_0$. The initial acceleration $a_0=d^2x/dt^2(0)$ can be calculated from the ODE (1) at $t=0$:

$$ma_0+cv_0+kx_0=0\implies a_0=-\frac{cv_0+kx_0}{m}$$

Then, the approximation for $x_1$ is:

$$x_1\approx x_0+v_0\Delta t-\frac{1}{2}\left(\frac{cv_0+kx_0}{m}\right)(\Delta t)^2 \tag{7}$$

### *3.4. Visualization*

For homogeneous second-order linear ODE with constant coefficients, analytic solutions can be found. For the case of *underdamped*, the solution is in the form $x(t) = e^{-\gamma t}(A\cos(\omega_d t) + B\sin(\omega_d t))$, where $\gamma = c/(2m)$ and $\omega_d = \sqrt{k/m - (c/2m)^2}$. Constant $A$ and $B$ determined by initial conditions [15, 16].

To demonstrate this method, the simulation was conducted with the following parameters: Mass $m = 1.0$ kg, Spring Constant $k = 10.0$ N/m, Damping Coefficient $c = 0.5$ Ns/m (the *underdamped* case), initial position $x_0 = 1.0$ m, initial velocity $v_0 = 0.0$ m/s, total time $T = 10.0$ s, and time steps $\Delta t = 0.5, 0.1, 0.01$ s.

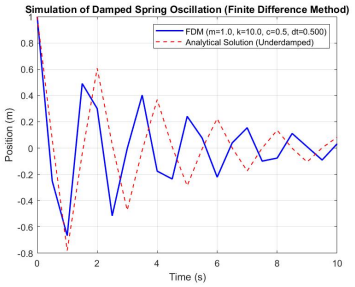


**Figure 2.** Simulation with a time step size of 0.5 s

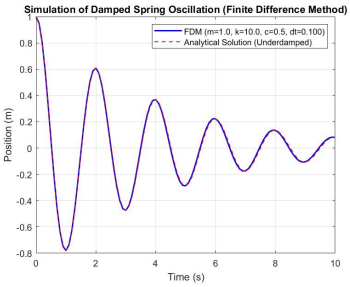


**Figure 3.** Simulation with a time step size of 0.1 s

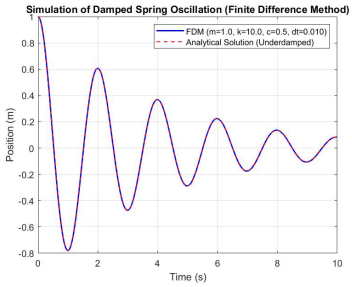


**Figure 4.** Simulation with a time step size of 0.01 s

The error values for each time step can be seen in Table 2.

**Table 2.** Comparison of Errors for each Time Step

| Step Time | SMAPE |
|---|---|
| 0.5 | 168.9687% |
| 0.1 | 22.5830% |
| 0.01 | 0.4495% |

The simulation results (Figure 4) are qualitatively consistent with the expected physical behavior of a damped harmonic oscillator. Oscillation occurs due to the spring's restoring force ($-kx$), and damping occurs due to the frictional force ($-c\,dx/dt$), which always opposes the direction of velocity, thereby removing energy from the system and causing the amplitude to decrease.

According to Table 2, the accuracy of the Finite Difference Method depends on $\Delta t$. Smaller time steps yield more accurate solutions. However, an excessively small $\Delta t$ will increase computation time. Conversely, an excessively large $\Delta t$ can lead to inaccuracies or even numerical instability.

If $c$ is increased, the amplitude will decrease more rapidly. Simulations with different values of $c$ will show the transition from *underdamped* ($c < \sqrt{4mk}$) to *critically damped* ($c = \sqrt{4mk}$) and *overdamped* ($c > \sqrt{4mk}$). If $c = 0$, the simulation will produce simple harmonic oscillation with a constant amplitude.

## 4. Conclusions

This study successfully demonstrated the numerical simulation of spring oscillation using the Finite Difference Method, which qualitatively captured the physical behavior of the dynamics of damped oscillation, wherein oscillation is caused by the restoring force and damping reduces the amplitude. The accuracy of the numerical solution, validated against the analytical solution and measured using

SMAPE, was found to be significantly influenced by the time step size ($\Delta t$), with smaller $\Delta t$ values yielding higher accuracy but with the potential for increased computational load, while large $\Delta t$ values can lead to inaccuracies. This study also confirmed the transition in the behavior of the oscillating system due to variations in the damping coefficient. Overall, the Finite Difference Method proved effective for analyzing spring oscillation systems, providing valuable insights for theoretical and practical applications in mechanical systems and vibration analysis, as well as opening avenues for further research on more complex numerical schemes or non-linear oscillation models.

## REFERENCES


[1] P. P. Darajat, "Simulasi numerik metode beda hingga non uniform gird dalam menyelesaikan masalah nilai batas persamaan diferensial biasa," *G-Tech: Jurnal Teknologi Terapan*, vol. 4, no. 2, pp. 309–314, 2021. https://doi.org/10.33379/gtech.v4i2.633.

[2] M. Malik and S. Mardiyati, "Syarat cukup osilasi persamaan diferensial linier homogen orde dua dengan redaman," *Teorema*, vol. 1, p. 5.

[3] M. R. Ghifari, R. A. Pramono, and K. N. Aziz, "Aplikasi gaya magnet pada fenomena osilasi teredam dalam sistem gerak harmonik sederhana," *Jurnal Penelitian Fisika Dan Terapannya (JUPITER)*, vol. 5, no. 1, pp. 16–22, 2023. https://doi.org/10.31851/jupiter.v5i1.11839.

[4] R. M. Muspa and Y. Pramudya, "Studi numerik gerak osilasi teredam balok pada bidang miring," *JRST (Jurnal Riset Sains dan Teknologi)*, vol. 7, no. 1, pp. 59–64, 2023. https://doi.org/10.30595/jrst.v7i1.15315.

[5] N. A. Robbaniyyah, A. S. Muliyanti, D. A. Malasso, and D. H. Pajri, "Simulasi dan akurasi numerik persamaan gelombang satu dimensi menggunakan aproksimasi metode beda hingga," *Semeton Mathematics Journal*, vol. 1, no. 1, pp. 8–14, 2024. https://doi.org/10.29303/semeton.v1i1.204.

[6] M. A. Septianda, M. Kiftiah, and Y. Yudhi, "Penyelesaian persamaan diferensial biasa orde dua dengan metode blok k-langkah," *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, vol. 10, no. 4. https://jurnal.untan.ac.id/index.php/jbmstr/article/view/7126.

[7] Y. Acu, B. P. Lapanporo, and A. A. Kushadiwijayanto, "Model sederhana gerak osilator dengan massa berubah terhadap waktu menggunakan metode runge kutta," *POSITRON*, vol. 7, no. 2, pp. 42–47, 2017. https://jurnal.untan.ac.id/index.php/jpositron/article/view/23276.

[8] H. A. Sitompul and E. W. Siahaan, "Solusi numerik persamaan diferensial biasa orde dua dengan sistem persamaan nonlinier," *Jurnal Ilmiah Teknik Sipil*, vol. 11, no. 2, pp. 379–389, 2022. https://jurnal.universitasdarmaagung.ac.id/tekniksipil/article/view/2841.

[9] H. W. Permata, A. Kusumastuti, and J. Juhari, "Solusi numerik model gerak osilasi vertikal dan torsional pada jembatan gantung," *Jurnal Riset Mahasiswa Matematika*, vol. 1, no. 1, pp. 1–13, 2021. https://repository.uin-malang.ac.id/9469/.

[10] M. R. Haizar, M. Rizki, N. A. Robbaniyyah, B. N. Syechah, S. Salwa, and L. Awalushaumi, "Numerical solution of the korteweg-de vries equation using finite difference method," *EIGEN MATHEMATICS JOURNAL*, vol. 7, no. 1, pp. 1–7, 2024. https://doi.org/10.29303/emj.v7i1.190.

[11] A. F. Rozanni, *Pemodelan Sistem Pegas dengan Dua Beban Massa*. PhD thesis, Universitas Islam Negeri Maulana Malik Ibrahim, 2020. http://etheses.uin-malang.ac.id/24742/.

[12] S. Rahmatullah, Y. Arman, and A. Apriansyah, "Simulasi gerak osilasi model pegas bergandeng menggunakan metode runge-kutta," *Prisma Fisika*, vol. 8, no. 3, pp. 180–184, 2020. https://doi.org/10.26418/pf.v8i3.43681.

[13] A. N. Rais, R. Rousyati, I. J. Thira, D. N. Kholifah, N. Purwati, and Y. M. Kristania, "Evaluasi metode forecasting pada data kunjungan wisatawan mancanegara ke indonesia," *EVOLUSI: Jur-*

*nal Sains Dan Manajemen*, vol. 8, no. 2, 2020. https://ejurnal.ung.ac.id/index.php/jjom/article/view/10068.

[14] A. Z. Arifin, "Simulasi dampak penghalang pada gelombang tsunami menggunakan persamaan air dangkal dengan metode beda hingga," *Jambura Journal of Mathematics*, vol. 3, no. 2, pp. 93–102, 2021. https://doi.org/10.34312/jjom.v3i2.10068.

[15] O. Soegihardjo, "Simulasi komputer untuk analisis karakteristik model sistem pegas-peredam kejut-massa," *Jurnal Teknik Mesin*, vol. 3, no. 1, pp. 29–34, 2001.

[16] M. Cahyadi and S. N. Khotimah, "Visualisasi solusi analitik dari osilasi pegas dengan menggunakan visual basic for application (vba)," 2015. https://ifory.id/proceedings/2015/z4pZjcJkq/snips_2015_melisa_cahyadi_4b2be883495be3878e3f02662b89b460.pdf.