A Simple Mathematical Model and Simulation of a Seismograph

Sanjana Gupta *

ABSTRACT This project delves into a Python-based simulation of seismograph behavior using a mathematical model. By employing a mass-spring-damper system and numerical integration, seismic responses are visualized and Richter scale intensities calculated. The simulation showcases the interplay of forces and aids in comprehending seismograph mechanics.

1 Introduction

Seismographs are instruments used to detect and record the vibrations produced by earthquakes. They play a crucial role in monitoring and analyzing seismic activities. In this project, a mathematical model of a seismograph is developed and simulated using Python. The objective is to understand how a seismograph functions and to visualize the displacement caused by an earthquake. The simulation employs a mass-spring-damper system subjected to an earthquake force profile.

2 Methods and Simulations

2.1 Mathematical Model

he mathematical model used in the simulation involves a second-order linear ordinary differential equation (ODE) that represents the behavior of the mass-spring-damper system. Here is the differential equation that governs the system:

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

The system consists of three main parameters:
Mass (m): Represents the mass of the seismograph's mass block (kg).
Damping Coefficient (c): Combines both mechanical and electrical damping effects (Ns/m).
Spring Constant (k): Indicates the stiffness of the spring (N/m).
Earthquake Force (F_t): The Force generated by the ground movement as a function of time.

This equation represents the balance between the mass's inertia, the damping forces, and the spring forces.

To solve this second-order ODE numerically, it is typically converted into a system of first-order ODEs. Introducing a new variable $v(t)$ for the velocity ($v = \frac{dx}{dt}$), the system can be rewritten as a first-order system:

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = \frac{F_t - cv - kx}{m}$$

These equations can be integrated using numerical methods like the Runge-Kutta method, as demonstrated in the provided code.

2.2 Earthquake Force Generation

The earthquake force is generated as a function of time using the `earthquake_force` function. The function produces a force signal by adding P-wave, S-wave and Surface wave components along with random noise. These signals decay over time according to exponential decay functions, mimicking the behavior of real earthquake signals. As the S-waves generally have a higher amplitude, higher frequency and lower velocity compared to P-waves so the force functions have been manipulated accordingly to make them visually look like a real Earthquake force. The exponential damping and random noise are added to replicate real conditions.

2.3 Numerical Integration: Runge-Kutta Method

To simulate the dynamic behavior of the mass-spring-damper system, the Runge-Kutta method is employed for numerical integration. The Runge-Kutta method involves breaking down the problem into a series of steps. At each step, it estimates the rate of change of the function at differ-

* Department of Physics, Ashoka University, Sonepat, 131029 India, sanjana.gupta207@gmail.com

ent points, then uses these estimates to calculate the next value of the function. The most commonly used variant of the Runge-Kutta method is the fourth-order Runge-Kutta method (RK4), which uses four estimates to improve accuracy. The `runge_kutta_step` function calculates the new displacement and velocity of the system at each time step. This iterative process provides a trajectory of the mass block's movement over time in response to the earthquake force.

2.4 Simulation Parameters

The simulation's total time span is set to 50 seconds, and the time step for numerical integration (dt) is 0.01 seconds. The initial conditions for displacement (x) and velocity (v) are both initialized to zero.

2.5 Visualization

The visualization aspect of the simulation is realized through the `matplotlib` library. The simulation results are displayed as a dynamic animation of the mass block's displacement over time. Additionally, the Richter scale value is calculated based on the maximum displacement amplitude (A) achieved during the simulation. The amplitude is normalized with respect to a reference amplitude (A_0) of 1 micron (10^{-6} meters). In the animation, the Richter Scale value is calculated in real time but it remains fixed until a higher amplitude is reached.

3 Summary and Discussions

This project successfully demonstrates the development of a mathematical model for simulating a seismograph using Python. Through the implementation of a mass-spring-damper system and the Runge-Kutta numerical integration method, the simulation provides insights into the displacement behavior of the seismograph during an earthquake. The visualization component presents an animated display of the system's response to the earthquake force. The calculated Richter scale value offers a quantitative measure of the simulated seismic activity's intensity. Overall, this project enhances understanding of seismograph mechanics and their role in earthquake monitoring.

For a more detailed understanding of the code and its execution, kindly refer to the provided Python script.

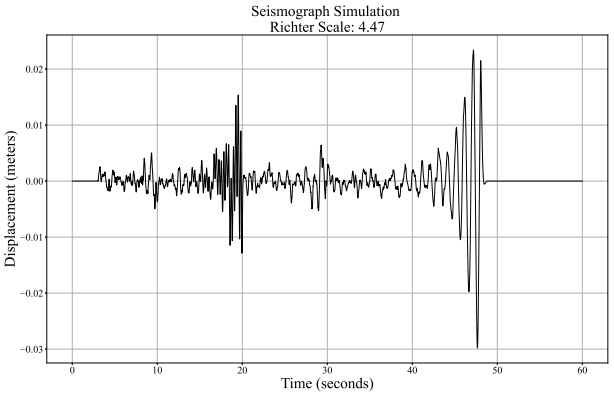

Fig. 1. Displacement of the Seismograph Needle with time. The P-waves, S-waves and surface waves can be clearly noticed in the figure

Acknowledgements

I extend my heartfelt gratitude to the organizers of The Remote Experience for Young Engineers and Scientists (REYES) for providing me with the invaluable opportunity to participate in this project. I am also thankful to Flavio Cesar Nieto Ruiz of Instituto Tecnológico de Celaya for their assistance in selecting the project.

References

[1] Soto, S., Becerra, Y., Suarez, C., Gamboa, E. (2019). Seismograph Design for Landslide Monitoring in the Andean Region Using Automatic Control Techniques and Mathematical Modeling. In: Tan, Y., Shi, Y., Niu, B. (eds) Advances in Swarm Intelligence. ICSI 2019. Lecture Notes in Computer Science, vol 11656. Springer, Cham. `https://doi.org/10.1007/978-3-030-26354-6_23`.

[2] Simmons, G. F. (1991). *Differential Equations*. McGraw-Hill.