

From Ordinary Differential equations to Neural Networks, solving mechanics problems in a new way

Introduction.

The aim of this project is to study an oscillation problem using different methods for solving ordinary differential equations. We will focus on traditional methods as well as on solving similar equations neural networks. For solving differential equations with Neural Networks we recommend the material at the [https://compphysics.github.io/MachineLearning/doc/pub/odenn/html/Learning and Data Analysis](https://compphysics.github.io/MachineLearning/doc/pub/odenn/html/Learning%20and%20Data%20Analysis) course.

The first part is a repeat of homework 5 on an oscillating block tied to a spring.

0.1 The standard ordinary differential equation solver

The relevant material with codes etc is covered by the [https://mhjensen.github.io/Physics321/doc/pub/harmonics/Notes on Oscillations](https://mhjensen.github.io/Physics321/doc/pub/harmonics/Notes%20on%20Oscillations). Taylor's chapter 5, in particular sections 5.1 and 5.2. 5.4 is relevant for the bonus exercise. We start now with our next numerical application with simple rewrites of our equations. The system we will look at is that of a block fastened to a spring (which in turn is tied to a wall). The force acting on the block from the spring in the x -direction only is

$$m \frac{d^2 x(t)}{dt^2} = F(x) = -k(x - b),$$

where k is a material specific constant and b is the equilibrium position. The block has mass m and t is time. Define the initial time as t_0 . We will for simplicity set the equilibrium position to zero, that is $b = 0$.

- Does this force conserve energy? If so, with given initial position and velocity x_0 and v_0 , respectively, find the expression for energy conservation in terms of the potential and kinetic energies.
- Define a constant $\omega_0 = \sqrt{k/m}$ and show that you can write the acceleration as $a(t) = -\omega_0^2 x(t)$. What is the dimensionality of ω_0 ? (it is normally called a natural frequency).
- Introduce now a dimensionless time $\tau = t\omega_0$. Show that you can rewrite the equation for the acceleration in terms of two first-order differential equations

$$\frac{dv}{d\tau} = -x,$$

and

$$\frac{dx}{d\tau} = v.$$

What are the dimensionalities of these two equations?

These are the equations which we will code in the next exercise.

- Find the analytical solutions for the velocity and position as function of the initial conditions.
- Write then a program which solves the above differential equations for the sliding block using the Euler-Cromer method and the velocity Verlet method. Study the results for the position, velocity and energy conservation as function of time and the step size in time. Compare the numerical results with the analytical ones. Check that your results for position and velocity obey periodicity.

Is energy properly conserved? Discuss your results.

Now we will add to our program a first derivative of position (the velocity) which is meant to mimick the role of friction. Our differential equation is now

$$m \frac{d^2 x(t)}{dt^2} + \beta \frac{dx}{dt} + kx = ma(t) + \beta v(t) + kx(t) = 0,$$

where k is a material specific constant and β is a constant representing the resitance from the plane on which the block slides. We have thus assumed that friction depends on velocity in a linear way.

We introduce again the dimensionless time $\tau = t\omega_0$ with $\omega_0 = \sqrt{k/m}$.

- Divide by m and show that you can rewrite your equation as two first-order coupled differential equations

$$\frac{dv}{d\tau} = -2v(t)\gamma - x(t),$$

and

$$\frac{dx}{d\tau} = v(t).$$

What are the dimensionalities of these two equations? We have defined $\gamma = \beta/2m$.

- Add the role of friction to your code and study the cases where (i) $\gamma < \omega_0$ (underdamping), (ii) $\gamma = \omega_0$ (critical damping) and (iii) $\gamma > \omega_0$ (overdamping). Plot your resulting positions as functions of time and discuss the physical meaning of your results.
- Add also a sinusoidal external force that depends only on time $F(t) = F_0 \cos(\omega t)$. Scale the equations again and study your results by adding the Runge-Kutta method to fourth order.

0.2 Neural Networks

Finally, with the above equations (you could start without friction and external forces), implement a Neural Network solver and compare the results from this Machine Learning algorithm with those obtained using the standard Euler-Cromer method and the fourth-order Runge-Kutta method. Discuss your results.