Numerical Solutions to Higher Order Ordinary Differential Equations using the Euler's method

Zain Ul Abideen[1*]

[1]Department of Space Science, Institute of Space Technology.
[*]Corresponding author. Email: dominuszain@gmail.com
[*]www.github.com/dominuszain/NumericalMethods/tree/main

Abstract

This study details the procedure for solving first order ordinary differential equations using the euler's method. The method was generalized to higher order ODEs by making substitutions to break them down into multiple first order ones. The particular example that was used was that of the damped harmonic oscillator. The solution was implemented in the Fortran language.

1 Introduction

Ordinary Differential Equations provide a conveient way of modelling physical phenomenon. Analytical solutions provide the most general results, but often times these analytical solutions are either an overkill or just not suited for the applications. For those cases, numerical methods are employed. The numerical method used in this study is known as the Euler's method. The language of choice for this study was Fortran95. This particular choice was made due to the compiled nature of the language, making it exceptionally fast for numerical calaulations.

2 Results and Discussions

The Euler's method can solve any first order ordinary differential equation of the form:

$$\frac{dy}{dx} = f(x, y) \tag{1}$$

The first step in the numerical solution would be to replace the infinitesimals with finite differences. The substitution would produce pretty accurate results as long as we keep those finite differences very small. We can use these finite differences in combination with some initial conditions to get an expression for a solution. Note that for higher derivatives, a direct finite substitution would require inital conditions that are not known for most physical systems. For that we will use substitutions.

$$\frac{\Delta y}{\Delta x} = f(x, y) \tag{2}$$

$$\Delta y = f(x, y)\Delta x \tag{3}$$

$$y_2 - y_1 = f(x_1, y_1)\Delta x \tag{4}$$

$$y_2 = y_1 + f(x_1, y_1)\Delta x \tag{5}$$

Provided we know the initial conditions (x_1, y_1), we can use the above iterative form to solve any first order ordinary differential equation. The generalized version of the above form would be:

$$y_{i+1} = y_i + f(x_i, y_i)\Delta x \tag{6}$$

Let's say we have the following second order ordinary differential equation:

$$\frac{d^2y}{dx^2} = f(x, y) \tag{7}$$

Since the Euler's method is only directly applicable to first order ODE's, we would have to break our differential equation into multiple first order ones. We can do that by using a simple substitution:

$$\frac{dy}{dx} = V \tag{8}$$

$$\frac{dV}{dx} = f(x, y) \tag{9}$$

Now, solving both of these first order ODEs simulatneously would be the same as solving the second order ODE. An n^{th} order differential equation can be broken down into n number of first order differential equations that need to be solved simultaneously.

$$\frac{\Delta y}{\Delta x} = V \tag{10}$$

$$y_2 = y_1 + V_1\Delta x \tag{11}$$

$$\frac{\Delta V}{\Delta x} = f(x, y) \tag{12}$$

$$V_2 = V_1 + f(x_1, y_1)\Delta x \tag{13}$$

$$y_{i+1} = y_i + V_i\Delta x \tag{14}$$

$$V_{i+1} = V_i + f(x_i, y_i)\Delta x \tag{15}$$

2.1 The Damped Harmonic Oscillator Problem

The differential equation that governs the Damped Harmonic Oscillator is given by the expression:

$$\frac{d^2x}{dt^2} = -\frac{b}{m}\frac{dx}{dt} - \frac{k}{m}x \tag{16}$$

We will use a substitution to disintegrate this differential equation into two simpler ones.

$$\frac{dx}{dt} = V \tag{17}$$

$$\frac{dV}{dt} = -\frac{b}{m}V - \frac{k}{m}x \tag{18}$$

$$\frac{\Delta x}{\Delta t} = V \tag{19}$$

$$x_2 = x_1 + V\Delta t \tag{20}$$

$$\frac{\Delta V}{\Delta t} = -\frac{b}{m}V - \frac{k}{m}x \tag{21}$$

$$V_2 = V_1 - (\frac{b}{m}V + \frac{k}{m}x)\Delta t \tag{22}$$

$$x_{i+1} = x_i + V_i\Delta t \tag{23}$$

$$V_{i+1} = V_i - (\frac{b}{m}V_i + \frac{k}{m}x_i)\Delta t \tag{24}$$

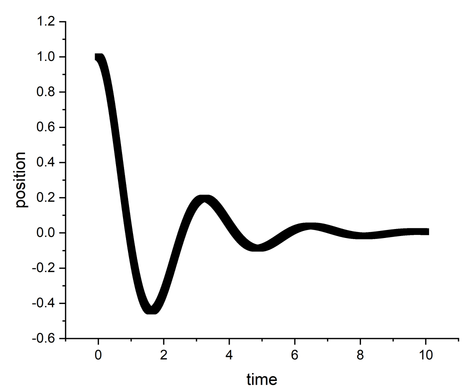

Figure 1: Solution of the above two numerical equations implimented in Fortran, and plotted in OriginLab. The constants b, m, k, and the initial conditions were set to trivial values.