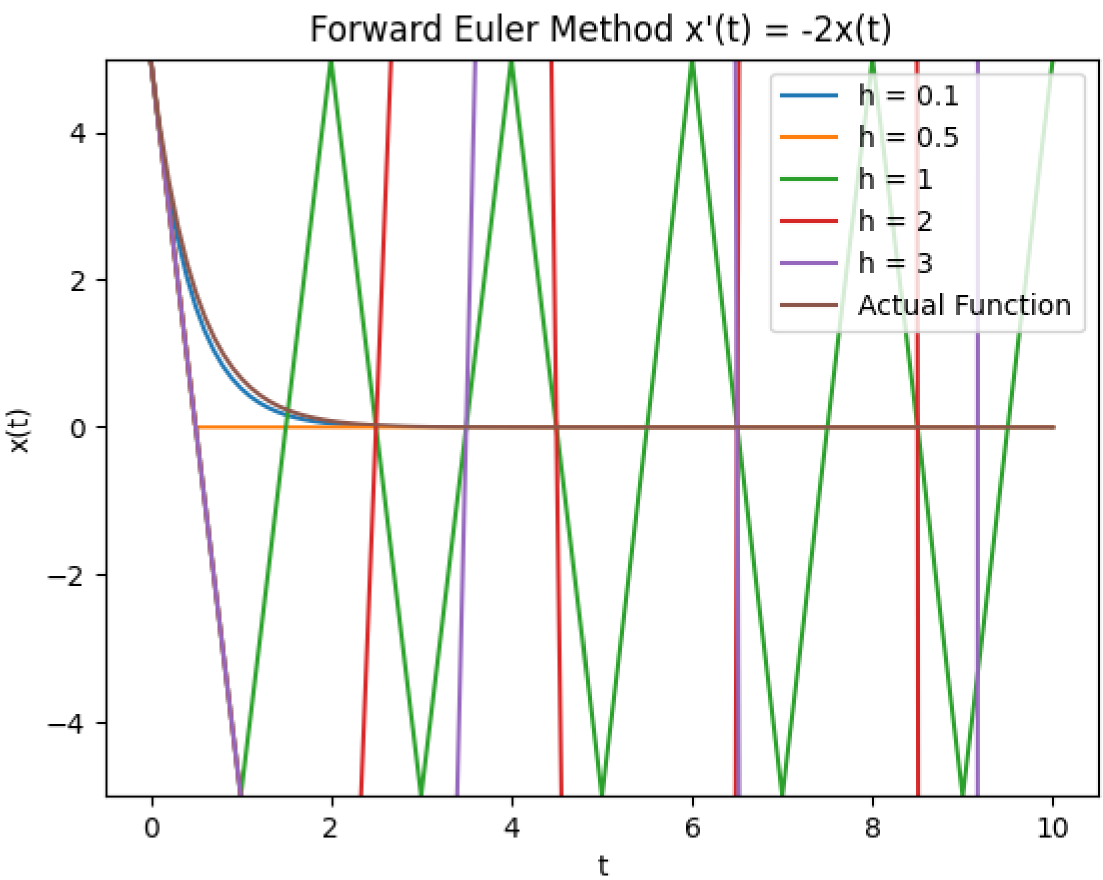
Lab Report

Q1. We took the Polynomal function from the previous assignment. In this question we made the following methods:

- *points* : this method will return the best fit fourier approximation of the input operation.
- *ODEsolve* : this method will use the forward euler method to solve the given ODE.

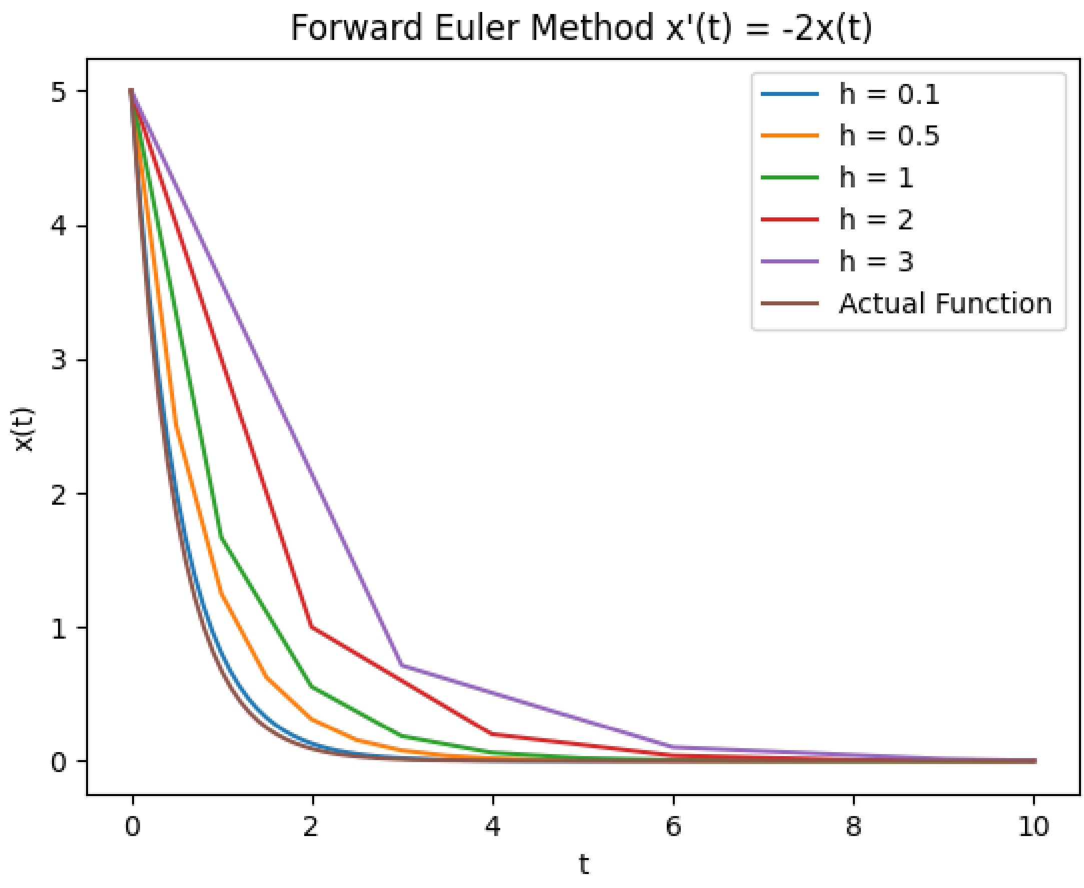
We can call the ODEsolve function and pass the step size as a parameter and it will output a polynomial that passes through the discrete solution point of ODE.



Q2. This question is the same as the above question but with a different ODE function as an input.
In this question we made the following methods:

- *points* : this method will return the best fit fourier approximation of the input operation.
- *ODEsolve* : this method will use the forward euler method to solve the given ODE.

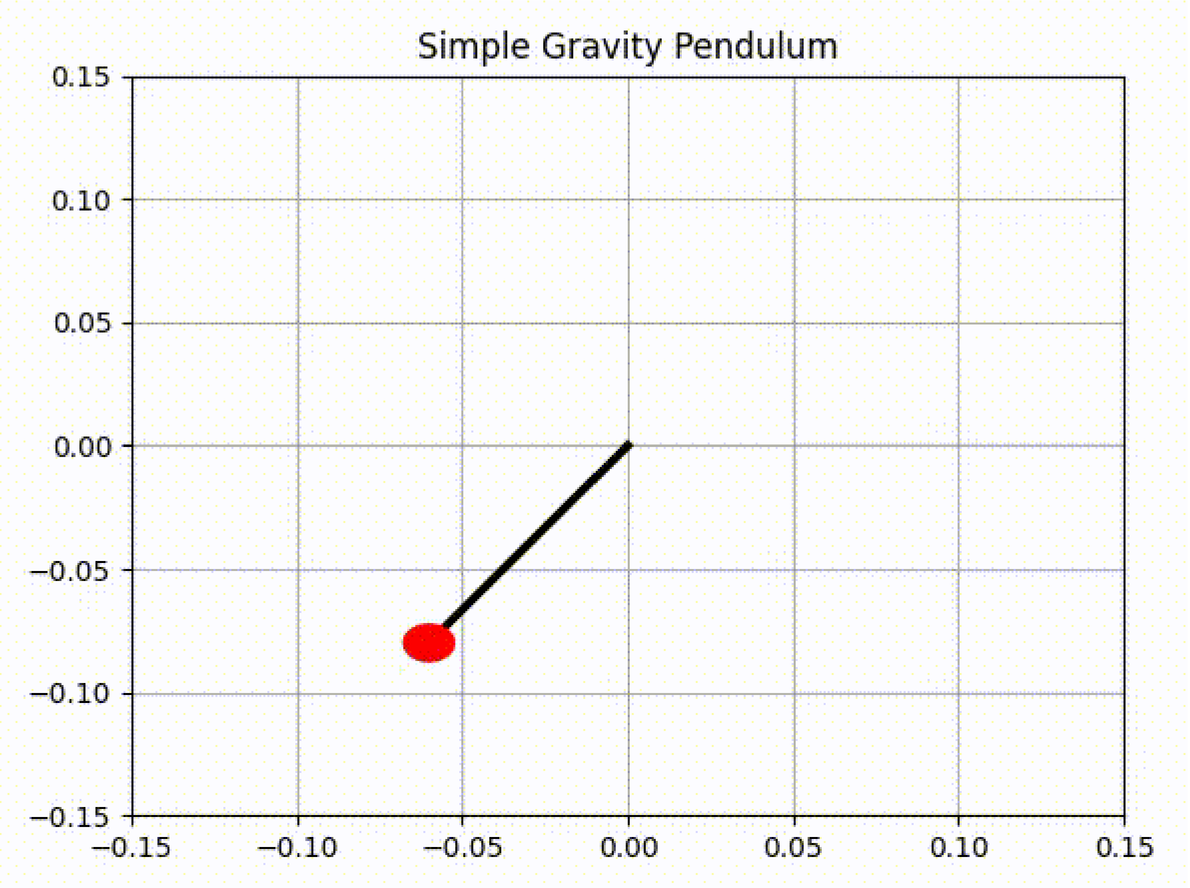
We can call the ODEsolve function and pass the step size as a parameter and it will output a polynomial that passes through the discrete solution point of ODE.



Q3. In this question we made the following methods:

- *coordinate* : this method will return the coordinate for an input theta.
- *Euler* : this method will apply the Euler method.
- *ODEsolve* : this method uses the forward euler method to solve the given ODE.
- *init* : this method will initialise the elements of the necessary graph elements.
- *animate* : this method will animate the graph.

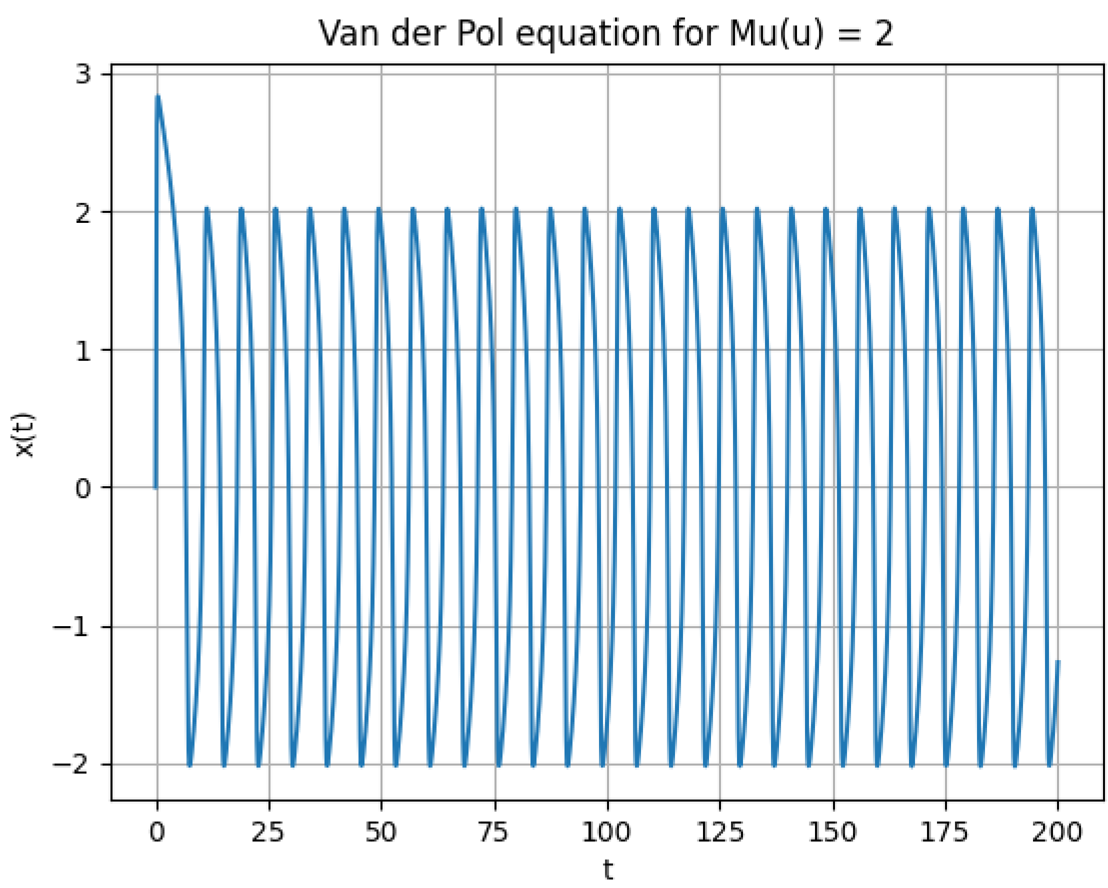
We can call the ODEsolve method and pass the ODE and it will return the animation of the simple pendulum.



Q4. In this question we made the following methods:

- *plot* : this method will plot the graph.
- *derv* : this method will return the derivative of the system of differential equations.
- *ODEsolve* : this method will compute the period of the limit cycle.

We can call and ODEsolve function and pass the initial condition and value of μ it will plot the graph for solution for the period of cycle.



μ(=2) = 7.641

Q5. In this question we made the following methods:

- *normal* : this method will return the normal of the vector.
- *derv2* : this method will return the 2nd derivative of the system of ode.
- *body* : this method will initial all the bodies.
- *ODEsolve* : this method will solve ODE using the euler method.
- *derv* : this method will return the derivative of the system of differential equations.
- *init* : this method will initialise the elements of the necessary graph elements.
- *animate* : this method will animate the graph.

We can call the ODEsolve method and give the position and velocity vector as input and it will return the animation for a three body problem.