ME2 Computing- Session 5: Numerical solution of differential equations: initial value problems

Learning outcomes:

- Being able to solve first order ODEs with explicit methods
- Being able to solve first order ODEs with implicit methods
- Being able to solve a system of first order ODEs and Higher order ODEs

Before you start:

In your H drive create a folder *H:\ME2MCP\Session5* and work within it.

We will be testing Tasks A and B with the ODE:

$$\frac{dy}{dt} = -2yt - 2t^3$$

whose analytical solution is:

$$y(t) = 1 - t^2 + ce^{-t^2}$$

Task A: Explicit methods: Forward Euler and RK4

1. Write a function, *FwEuler()*, to solve a general ODE:
$$\frac{dy}{dt} = F(t, y)$$
by adopting a forward Euler numerical scheme (slide 225).
The function receives the initial condition, t_0 and y_0, the time step h and the desired final computational time t_{end} (all these input arguments are scalars). The function outputs two arrays, t and y, describing the solution $y(t)$, both of dimensions $1 \times N_t$, where N_t is the number of temporal nodes computed.
Within *FwEuler()*, the mathematical function $F(t, y)$ can be evaluated by invoking a separate Python function *func()*.
Explicit methods are subject to instabilities: consider this when choosing the value for h.

2. Write a function *ODERK4()* to perform as the function at point 1, but implementing the Runge-Kutta method instead (slide 232).

Task B: Implicit methods: Backward Euler

1. Write a function, *BwEuler()*, to solve the above ODE, by adopting a backward Euler numerical scheme.
The function receives the initial condition, t_0 and y_0, the time step h and the desired final computational time t_{end} (all these input arguments are scalars).
The function outputs two arrays, t and y, describing the solution $y(t)$, both of dimensions $1 \times N_t$, where N_t is the number of temporal nodes computed.

2. Plot, on the same graph, the solutions obtained from Task A1, Task A2, Task B1 and the analytical solution, vs time.

Answer Quizzes 1 and 2

Task C: System of ODEs, with explicit methods

Modify the function *FwEuler()*, into a new function *FwEulerN()*, to solve a set of N_v given ODEs:

$$\begin{cases} \dfrac{dy_1}{dt} = F_1\left(t, y_1, y_2, \dots y_{N_v}\right) \\ \dfrac{dy_2}{dt} = F_2\left(t, y_1, y_2, \dots y_{N_v}\right) \\ \dots \\ \dfrac{dy_{N_v}}{dt} = F_{N_v}\left(t, y_1, y_2, \dots y_{N_v}\right) \end{cases}$$

with initial conditions:

$$\begin{cases} y_1(t_0) = y_1^0 \\ y_2(t_0) = y_2^0 \\ \dots \\ y_{N_v}(t_0) = y_{N_v}^0 \end{cases}$$

The function *FwEulerN()* receives as input: the initial and final computational time, t_0 and t_{end}, and the time step h (all these input arguments are scalars); the vector Y_0 with the initial values of the solution at $t = t_0$ (Y_0 has dimensions $1 \times N_v$, where N_v is the number of equations of the system (i.e., the number of variables solved).

The function *FwEulerN()* outputs an array t of dimensions $1 \times N_t$ and an array $Y = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_{N_v}(t) \end{bmatrix}$, of dimensions $N_v \times N_t$, with the solutions. N_t is the number of temporal nodes computed.

Task C1: Covid-19 model (SIR model)

The spread of the virus can be modelled considering three classes of population:

i) The number of susceptible individuals, S. These can be infected when exposed to the virus.

ii) The number of infected individuals, I, growing with a rate a.

iii) The number of recovered individuals, R. These have been infected and therefore become immune. The recovery rate is b.

The dynamics of the three classes are described by the set of ODEs:

$$\begin{cases} \dfrac{dS}{dt} = -aSI \\ \dfrac{dI}{dt} = aSI - bI \\ \dfrac{dR}{dt} = bI \end{cases}$$

Apply the function *FwEulerN()* to predict the number of S, I and R, with conditions:

a) Low infection rate: $a = 0.001$, $b = 0.05$, $S(0) = 500$, $I(0) = 10$, $R(0) = 0$, within the time window $t_0 = 0 - t_{end} = 100$ with $h = 0.05$.
b) High infection rate: $a = 0.01$, $b = 0.05$, $S(0) = 500$, $I(0) = 10$, $R(0) = 0$, within the time window $t_0 = 0 - t_{end} = 100$ with $h = 0.05$.
c) Large time step: $a = 0.01$, $b = 0.05$, $S(0) = 500$, $I(0) = 10$, $R(0) = 0$, within the time window $t_0 = 0 - t_{end} = 100$ with $h = 0.5$.
d) Zero infection rate: $a = 0$, $b = 0.05$, $S(0) = 500$, $I(0) = 10$, $R(0) = 0$, within the time window $t_0 = 0 - t_{end} = 100$ with $h = 0.05$.

Task C2: Financial model of the house market in London (Lotka-Volterra)

The house market exhibits a periodic trend, where the number of houses sold, N, is interdependent with the average house prices, £.

The set of ODEs describing the cycle is:

$$\begin{cases} \dfrac{d£}{dt} = 0.3£N - 0.8£ \\ \dfrac{dN}{dt} = 1.1N - N£ \end{cases}$$

where $£(t)$ is the average house price (in $100k$ pounds) and $N(t)$ is the number of houses sold (in thousands).

Apply the function *FwEulerN()* to predict the trend of $£(t)$ and $N(t)$, with initial conditions:

$$\begin{cases} £(t = 0) = 0.8 \\ N(t = 0) = 7 \end{cases}$$

over a period of 40 months, with a weekly step (i.e., $h = 0.005$).

Plot, on the same graph, $£(t)$ and $N(t)$ vs time. Plot also, in a different figure, $£(t)$ vs $N(t)$.

Answer Quiz 3

Task D: Higher order ODEs

An ODEs of higher order n^{th} can be decomposed into a set of n first order ODEs.

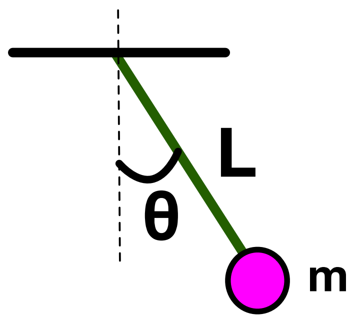
$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \ \dots + a_2\frac{d^2 y}{dt^2} + a_1\frac{dy}{dt} + a_0 y = b(t)$$

This can be achieved by introducing artificial variables:

$$\begin{cases} \dfrac{dy}{dt} = w_1 \\ \dfrac{d^2y}{dt^2} = \dfrac{dw_1}{dt} = w_2 \\ \dots \\ \dfrac{d^{n-1}y}{dt^{n-1}} = \dfrac{dw_{n-2}}{dt} = w_{n-1} \\ \dfrac{d^n y}{dt^n} = \dfrac{dw_{n-1}}{dt} = -a_{n-1}w_{n-1} - \ \dots - a_2 w_2 - a_1 w_1 - a_0 y + b(t) \end{cases}$$

The set of first order ODEs, is then made of n equations, with variables $y, w_1, w_2, \dots, w_{n-2}, w_{n-1}$.

Task D1: Damped non-linear motion of a pendulum



The oscillation of a pendulum of mass m, attached to a weightless string, is described by the second order ODE:

$$\frac{d^2\theta}{dt^2} + \frac{c}{m}\frac{d\theta}{dt} + \frac{g}{L}sin\theta = 0$$

where c is the damping coefficient, g the gravitational acceleration and L the length of the string.
The pendulum initially is at rest, displaced at an angle θ_0

Determine the motion of the pendulum, $\theta(t)$, for the first initial 15 seconds, with initial condition $\theta(t=0) = \pi/4$. (Use *FwEulerN* with $\Delta t = 0.005s$).
Plot into two subplots: the displacement $y(t)$ vs time t, and, the velocity $w_1(t)$ vs time t. Use a mass of 0.5Kg and a string $L = 1m$. Observe the difference between the swinging within a dry place ($c = 0.05Ns/m$) and within a humid viscous environment ($c = 0.18Ns/m$).

Answer Quiz 4

Task D2: Coupled spring-mass systems

The system in Figure D2 consists of three masses and four springs, fixed between two rigid walls. The masses are m_1, m_2 and m_3; the springs have Young modulus K_1, K_2, K_3 and K_4; the relaxed length of each spring is L_1, L_2, L_3 and L_4.

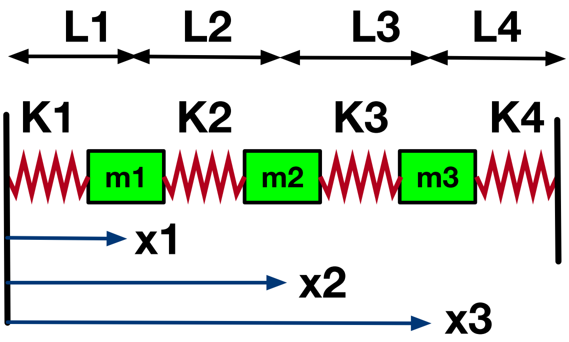


Figure D2

The displacement of the three masses is described by the set of second order ODEs:

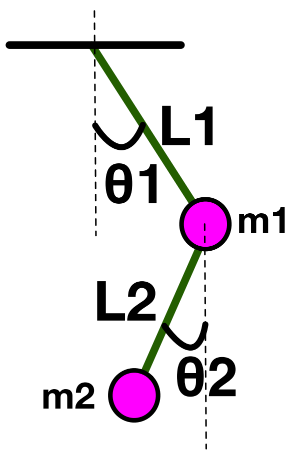
$$\begin{cases} m_1 \dfrac{d^2x_1}{dt^2} = -K_1(x_1 - L_1) + K_2(x_2 - x_1 - L_2) \\ m_2 \dfrac{d^2x_2}{dt^2} = -K_2(x_2 - x_1 - L_2) + K_3(x_3 - x_2 - L_3) \\ m_3 \dfrac{d^2x_3}{dt^2} = -K_3(x_3 - x_2 - L_3) + K_4(L_1 + L_2 + L_3 - x_3) \end{cases}$$

Calculate and plot the displacement of the three masses $x_1(t)$, $x_2(t)$ and $x_3(t)$, for various values of the system parameters and initial conditions of your choice.

Task D3: Motion of a double pendulum

The dynamics of the double pendulum, in an ideal viscous free environment, is described by the set of ODEs (slide 299):

$$\begin{cases} \ddot{\theta}_1 = \dfrac{m_2 g s\theta_2 c(\theta_1 - \theta_2) - m_2 s(\theta_1 - \theta_2)\left[L_1\dot{\theta}_1^2 c(\theta_1 - \theta_2) + L_2\dot{\theta}_2^2\right] - (m_1 + m_2) g s\theta_1}{L_1\left[m_1 + m_2 s^2(\theta_1 - \theta_2)\right]} \\ \ddot{\theta}_2 = \dfrac{(m_1 + m_2)\left[L_1\dot{\theta}_1^2 s(\theta_1 - \theta_2) + g s\theta_1 c(\theta_1 - \theta_2) - g s\theta_2\right] + m_2 L_2 \dot{\theta}_2^2 s(\theta_1 - \theta_2) c(\theta_1 - \theta_2)}{L_2\left[m_1 + m_2 s^2(\theta_1 - \theta_2)\right]} \end{cases}$$



Calculate and plot the displacement of the two masses $\theta_1(t)$ and $\theta_2(t)$ for:

$L_1 = 1m$, $L_2 = 0.5m$
$m_1 = 1Kg$ and $2Kg$, $m_2 = 1Kg$

$\theta_1(0) = \pi/4$, $\theta_2(0) = -\pi/4$
Zero initial velocities
Time step $h = 0.002$
$t_{end} = 40s$

Answer Quiz 5