ENGR20005
Numerical Methods in Engineering

Workshop 12

Part B: MATLAB Livescripts

12.1 The livescript *ENGR20005_Workshop12p1.mlx* runs through the solution of systems of differential equations in MATLAB.

(a) Read through the livescript and make sure you understand what each line of code does.

(b) Modify the livescript to solve the initial value problem

$$\frac{dy_1}{dt} = -\exp y_2 \tag{12.1a}$$

$$\frac{dy_2}{dt} = -\exp y_1 \tag{12.1b}$$

with the initial conditions $y_1(0) = 1$ and $y_2(0) = 2$.

12.2 The livescript *ENGR20005_Workshop12p2.mlx* covers the solution of a common class of ordinary differential equations called *stiff problems*.

(a) Read through the livescript and understand how to solve these problems.

(b) Using what you have learnt in this livescript, solve the system of differential equations

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -100 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{12.2}$$

with the initial conditions $y_1(0) = 1$ and $y_2(0) = 0$.

Part C: Applications

12.3 * The evolution of N vortices in an inviscid fluid is governed by the following set of differential equations

$$\frac{dx_j}{dt} = -\frac{1}{2\pi} \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{\omega_i (y_j - y_i)}{r_{ij}^2} \tag{12.3a}$$

$$\frac{dy_j}{dt} = \frac{1}{2\pi} \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{\omega_i (x_j - x_i)}{r_{ij}^2} \tag{12.3b}$$

where (x_j, y_j) is the position of the j^{th} vortex, $|\omega_j|$ is the strength of the vortex, the sign of ω_j denotes its direction (positive ω_j corresponds to an anti-clockwise direction of rotation), and r_{ij} is the distance between the i^{th} and j^{th} vortices.

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

(a) Consider a system with 2 counter-clockwise rotating vortices ($N = 2$) of unit strength. Find the differential equations that govern the position of these vortices.

(b) Apply the Explicit Euler method to the system of equations you derived in part (a).

(c) Write a MATLAB function (`.m` file) that uses your answer to part (b) to solve the system of differential equations you derived in part (a) with the initial conditions $(x_1(0) = 1, y_1(0) = 0)$ and $(x_2(0) = -1, y_2(0) = 0)$. Use a time step of $\Delta t = 0.01$ over the interval $0 \leq t \leq 200$.

(d) Now consider a system with 4 counter-clockwise rotating vortices placed at the points $(x, y) \in \{(1, 1), (1, -1), (-1, -1), (-1, 1)\}$. Derive the equations of motion from Eq. (12.3).

(e) Apply the RK4 method to the system of differential equations you derived in part (d).

(f) Write a MATLAB function (`.m` file) that uses the equations you derived in part (e) to solve the system of differential equations. Use a time step of $\Delta t = 0.01$ over the interval $0 \leq t \leq 200$.

12.4 In the livescript *ENGR20005_Workshop9p3.mlx*, we saw an example of solving the *convection equation*

$$\frac{\partial \phi}{\partial t} + U\frac{\partial \phi}{\partial x} = 0 \tag{12.4}$$

with periodic boundary conditions using a backwards difference scheme in space and the explicit Euler method in time.

We'll now solve Eq. (12.4) with periodic boundary conditions and the initial condition

$$\phi(x, 0) = \sin x$$

over the domain $x \in [0, 2\pi]$. We'll also assume an evenly spaced mesh with the spacing $\Delta x = 2\pi/(N-1)$, where N is the number of nodes.

In order to solve this problem numerically, we need approximate the problem in both space and time.

(a) We'll start by approximating the spatial derivatives of Eq. (12.4). In this case well apply a finite difference method to the term $\frac{\partial}{\partial x}$.

i. Use the backwards difference formula and show that

$$\frac{d\phi_i}{dt} = -\frac{U}{\Delta x}(\phi_i - \phi_{i-1}) \tag{12.5}$$

for the interior nodes $i = 1, \ldots, N-1$.

ii. Apply the boundary condition to the end nodes and show that the relations at the boundaries are given by

$$\begin{aligned} \frac{d\phi_0}{dt} &= -\frac{U}{\Delta x}(\phi_0 - \phi_N) \\ \frac{d\phi_N}{dt} &= -\frac{U}{\Delta x}(\phi_N - \phi_{N-1}) \end{aligned}$$

iii. Hence show that the the discretised problem may be written as

$$\frac{d}{dt}\{\phi\} = \frac{U}{\Delta x}[C]\{\phi\} \tag{12.6}$$

where $[C]$ is the *convection matrix* given by

$$[C] = \begin{bmatrix} -1 & 0 & \cdots & 0 & 1 \\ 1 & -1 & & & \vdots \\ \vdots & & \ddots & & \\ 0 & & 1 & -1 & 0 \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix}$$

(b) Hopefully you can see that by applying a finite difference approximation to Eq. (12.4), we've reduced a partial differential equation to a system of first order ordinary differential equations, which we can then solve using any of the methods covered in Lecture 21.

i. Apply the explicit Euler method to Eq. (12.6) and show that the iterative formula is given by

$$\{\phi\}_{n+1} = \{\phi\}_n + \frac{U\Delta t}{\Delta x}[C]\{\phi\}_n \tag{12.7}$$

ii. Modify the code snippets given in *ENGR20005_Workshop12p1.mlx* in order to apply Eq. (12.7). Use your script to solve Eq. (12.4) with a convection velocity of $U = 1$, a time step of $\Delta t = 0.005$, and 1000 nodes.