EEE4119F – practical 3

Student names: Student numbers:

Question 1 – Euler

Given the differential equation

$$\frac{dy}{dt} = -y$$

with analytical solution $y(t) = Ce^{-t}$,

1. Write a `function dy = deriv(time, y)` which returns the derivative of y at that point in time. This essentially characterises the differential equation in a function. In this case, the function will just return `dy = -y` (ie don't take the symbolic derivative)
2. Write a `function [t, y] = euler1stOrder(y0, f, dt, t_final)` to numerically integrate the given ODE. `y0` is the initial value of y, `f` is the `deriv` function written previously (passed as `@deriv`), `dt` is the simulation timestep and `t_final` is the final time.
3. Plot the analytical solution against the numerical solution for `dt = [0.1, 0.2, 0.4, 0.8]`.

Hint: if you want to put all functions in one script, make sure the function definitions come AFTER you use them in your code (ie, at the end of your script)

Question 2 – Runge Kutta 4 (RK4)

1. Write a `function [t, y] = runge_kutta_4(y0, f, dt, t_final)` to numerically integrate the previous ODE for the same set of timesteps. The algorithm is given below.
2. Plot the Runge Kutta solution against the analytical solution and Euler method.

$$k1 = \Delta t f(t^N, y^N)$$
$$k2 = \Delta t f(t^N + \Delta t/2, y^N + k1/2)$$
$$k3 = \Delta t f(t^N + \Delta t/2, y^N + k2/2)$$
$$k4 = \Delta t f(t^N + \Delta t, y^N + k3)$$
$$y^{N+1} = y^N + \frac{k1}{6} + \frac{k2}{3} + \frac{k3}{3} + \frac{k4}{6}$$

Question 3 – built-in solver

In this question, you will use one of matlab's built in solvers to solve an ODE.

1. Using the options = `odeset( )` command, set the absolute tolerance option of the solver to 1e-6. (Use `help` to find out how)

2. Use `[t, y] = ode45(ODEFUN, TSPAN, Y0, OPTIONS)` to solve the Lorenz equation given below. Note that it should still have the form `[dy] = Lorenz(time, y)`, except y and dy are vectors. Let `Y0 = [1, 1, 1]` and simulate over 30 seconds.

$$\frac{dx}{dt} = 10\,(y-x)\,,\ \ \frac{dy}{dt} = x\cdot\,(27-z)-y,\ \ \frac{dz}{dt} = x\cdot y-\frac{8}{3}z$$

3. Produce a 3D plot of the output using `plot3`

Question 4 – basic symbolic simplification

Use MATLAB's symbolic toolbox to `simplify`/prove the following identities:

1. $sin\,(x)^{\,2} + cos\,(x)^{\,2} = 1$
2. $sin\,(x)\ cos\,(y)\ + cos\,(x)\ sin\,(y)\ = sin\,(x+y)$
3. $cosh\,(x)^{\,2} - sinh\,(x)^{\,2} = 1$

Hint: start with the line `syms x y;`

Question 5 – solve a simple equation

The law of cosines for a triangle states that $a^2 = b^2 + c^2 - 2\,b\,c\,cos(A)$, where a is the length of the side opposite the angle A, and b and c are the lengths of the other sides.

1. Use the symbolic toolbox to solve for b, using the `solve` command
2. Given that $A = 60°$, $a = 5m$ and $c = 2m$, determine b using the `subs` command. Don't forget that MATLAB trig functions take radians as input

Question 6 – symbolic pole placement

Given state space matrices $A = [0\ \ -1;\ 5\ 2]$ and $b = [0;\ 1]$, use the characteristic equation

$$\det\left(sI - A + bK^T\right)\ = 0$$

to find controller gains $K = \left[k_1;\ k_2\right]$ to place the resulting system's poles at $s = -5,\ -10$.

A possible approach:

1. Calculate your characteristic equation using the information given above, and leave it as a polynomial by not setting it equal to zero
2. Do the same for your pole placement equation: $eq2 = (s+p_1)(s+p_2)$
3. Just as you would by hand, get the coefficients of each equation using `coeffs(equation, s)` and set each coefficient equal to each other (thus eliminating s from the equations).
4. Solve for K. You should get $k_1 = -45$, $k_2 = 17$