LAB 3: Runge-Kutta Methods

In this lab, you will extend the code for Euler's method from Lab 2 to implement a RK2 method to solve IVPs of the form

$$y(t_0) = y_0, \quad y'(t) = f(t, y) \text{ for } t > t_0.$$

1 RK2

A generic one-step method is written as

$$y_{i+1} = y_i + h\Phi(t_i, y_i; h) \text{ for } i = 1, 2, \ldots, n.$$

To get a Runge-Kutta 2 ("**RK2**") method, set

$$k_1 = f(t_i, y_i), \tag{1a}$$
$$k_2 = f(t_i + \alpha h, y_i + \beta h k_1), \tag{1b}$$
$$\Phi(t_i, y_i; h) = ak_1 + bk_2. \tag{1c}$$

In Section 2.4, we saw that if we pick any $b \neq 0$, and let

$$a = 1 - b, \qquad \alpha = \frac{1}{2b}, \qquad \beta = \alpha, \tag{2}$$

then we get a second-order method: $|\mathcal{E}_n| \leq Kh^2$.

For example, if we choose $b = 1$, we get the so-called *Modified* or *Mid-Point* Euler Method. However, *any* value of b, other than $b = 0$, should give a second-order method.

2 Your own RK2 method

In Lab 2, you wrote a MATLAB script that used the basic Euler method to solve an IVP.

You will now adapt this to implement an RK2 method.

1. Take b in (1c) to be the last digit of your ID number, unless that is 0, in which case take $b = -3/2$. (For example, if your ID number is 01234567, take $b = 7$. If your ID number is 76543210, take $b = -3/2$).

   Compute the values of a, α and β according to (2).

2. Choose an initial value problem to solve, and for which you know the exact solution. To avoid having a problem that is too simple,

   - your solution should involve trigonometric, logarithmic or nth-root functions.
   - f should depend explicitly on both t and y.

   (Hint: decide on the solution first, and then differentiate that to get f). You also need to choose an initial time, t_0, and a final time for the simulation, t_n.

3. The MATLAB program should approximate the solution to this IVP using your RK2 method for $n = 2$, $n = 4$, $n = 8$, $\ldots$, $n = 512$ (at least). For each n it should output the estimate for $y(t_n)$ and the error $|\mathcal{E}_n| = |y(t_n) - y_n|$.

4. The program should produce a figure displaying a log-log plot of these errors against the corresponding values of n, as well as n^{-2} against n. If your method is second-order, then these two lines should be parallel.

3 What to upload

In `Assignments and Labs` section of the MA385 Blackboard module, click on "Lab 3". Upload your RK2 solver.

Add appropriate comments to the top of your file(s) indicating

- your name, ID number and email address;
- date: when you wrote it the code
- what it does: what problem does it solve?
- how it does it

("*Who, When, What, How?*").

Make sure your program runs as-is before uploading. If you don't, you might have given it an invalid name, such as one containing spaces or mathematical symbols.

The deadline for uploading your code is **5pm, Friday, 15 November**.