

2.8 LAB 3: RK2 and RK3 methods

In this lab, you will extend the code for Euler's method from Lab 2 to implement higher-order methods to solve IVPs of the form

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

In particular, you will write programs to implement certain RK2 and RK3 methods.

2.8.1 RK2

A generic one-step method is written as

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

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

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + \alpha h, y_i + \beta h k_1), \\ \Phi(t_i, y_i; h) &= \alpha k_1 + \beta k_2. \end{aligned}$$

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

$$\alpha = 1 - b, \quad \alpha = \frac{1}{2b}, \quad \beta = \alpha, \quad (2.8.1)$$

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 from Section 2.4. However, *any* value of b , other than $b = 0$ should give a second-order method.

Download the MATLAB script `Euler_Solution.m` and run it. Make sure you understand how it works. Next, adapt its to preform the following tasks.

1. Choose the value of b to be the last digit of your ID number, unless that is 0, in which case take $b = -1$. (For example, if your ID number is 01234567, take $b = 7$. If your ID number is 76543210, take $b = -1$).
2. Compute the values of α , α and β according to (2.8.1).
3. 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 n th-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 .

4. The MATLAB program should approximate the solution to this IVP using your RK2 method for $n = 2$, $n = 4$, $n = 8$, ..., $n = 512$ (at least). For each n it should output the estimate for $y(t_n)$ and the error $|\mathcal{E}_n| = |y(t_n) - y_n|$.
5. 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.

2.8.2 RK3

Next write a program that implements the RK3-1 method given in Section 2.6.3:

$$\begin{array}{c|cc} \alpha_1 & 0 & \\ \alpha_2 & \beta_{21} & \\ \alpha_3 & \beta_{31} & \beta_{32} \end{array} = \begin{array}{c|cc} 2/3 & 2/3 & \\ 2/3 & 1/3 & 1/3 \end{array}$$

$$\begin{array}{c|ccc} & b_1 & b_2 & b_3 \end{array} \quad \begin{array}{c|ccc} & 1/4 & 0 & 3/4 \end{array}$$

The method is then

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + \alpha_2 h, y_i + \beta_{21} h k_1), \\ k_3 &= f(t_i + \alpha_3 h, y_i + \beta_{31} h k_1 + \beta_{32} h k_2), \end{aligned}$$

and

$$\Phi(t_i, y_i; h) = b_1 k_1 + b_2 k_2 + b_3 k_3.$$

As with the RK2 method, the MATLAB program should approximate the solution to this IVP using this RK3 method for $n = 2$, $n = 4$, $n = 8$, For each n it should output the estimate for $y(t_n)$ and the error $|\mathcal{E}_n| = |y(t_n) - y_n|$.

The program should also produce a figure displaying a log-log plot of these errors against the corresponding values of n , as well as n^{-3} against n .

2.8.3 What to upload

When your RK2 and RK3 programs are complete, upload them to the "Lab 3", under the "Assignments and Labs" tab on Blackboard. This can be done in two separate files, but it is okay to combine them to a single file if you prefer. (After all, every RK2 method is an example of an RK3 method).

Add appropriate comments to the top of your file(s) indicating who wrote it, when they wrote it, what it does, and how it does it ("Who/When/What/How?"). Include your ID number, and give the program a sensible name, which includes something distinctive like you name and ID number (so that I don't end up with 50 programmes all called `RK2.m`).

Make sure your programmes run 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 **TBD**.