

# Runge-Kutta Methods

July 2022

## 1 Introduction

Runge-Kutta (RK) methods are one of the most often employed approaches for solving ODEs. Recall that the Euler approach approximates numerical integration using the first two terms of the Taylor series. We can significantly enhance the accuracy of numerical integration by retaining more terms of the series, which demands additional effort. Using the RK methods, which are based on truncated Taylor series but do not involve computation of these higher derivatives, this additional labor may be avoided.

Following is a definition of Runge-Kutta schemes in general:

$$\begin{aligned}y^{n+1} &= y^n + dt(b_1 k_1 + \dots + b_s k_s) \\k_1 &= f(t^n, y^n) \\k_2 &= f(t^n + c_2 dt, y^n + dt a_{21} k_1) \\k_s &= f(t^n + c_s dt, y^n + dt a_{s1} k_1 + \dots + a_{s,s-1} k_{s-1}).\end{aligned}\tag{1}$$

Some restrictions are therefore applied to all coefficients in order to attain a particular level of accuracy,  $O(dtp)$ . Runge- $s$ -stage Kutta's approach is said to be of order  $p$ .

## 2 Second-order Runge Kutta Scheme

First, let us develop the second-order RK method. If we assume that  $dy/dt = f(t, y)$ , we may use the following integration formula

$$y^{n+1} = y + y' = y^n + \frac{1}{2}(k_1 + k_2)h,\tag{2}$$

where  $y_{n+1}$  is the value of  $y$  at  $t = N + 1$   $-0 \leq N \leq \frac{t-t_o}{h}$ , the  $h$  is the time step and thus  $t_{n+1} = t_o + h$ , and  $y_n$  is the value of  $y$  at  $t = t + N$ .

In essence, the following value,  $y_{n+1}$ , is calculated using the current value,  $y_n$ , and the weighted average of two increments:

- $K_1$  is the increment calculated using  $y$  and the slope at the start of the interval.
- Using  $(y + h k_1/2)$ ,  $K_2$  is the increment depending on the slope at the interval's midpoint.

This method is a second-order one, which means that the total cumulative (global) error is of order  $O(h^4)$ , while the local truncation error is on the order of  $O(h^3)$ .

### 3 Fourth-order Runge Kutta Scheme

$$y^{n+1} = y + y' = y^n + \frac{1}{6}(k_1 + k_2 + k_3 + k_4)h, \quad (3)$$

The following value,  $y_{n+1}$ , is calculated using the current value,  $y_n$ , and the weighted average of four increments:

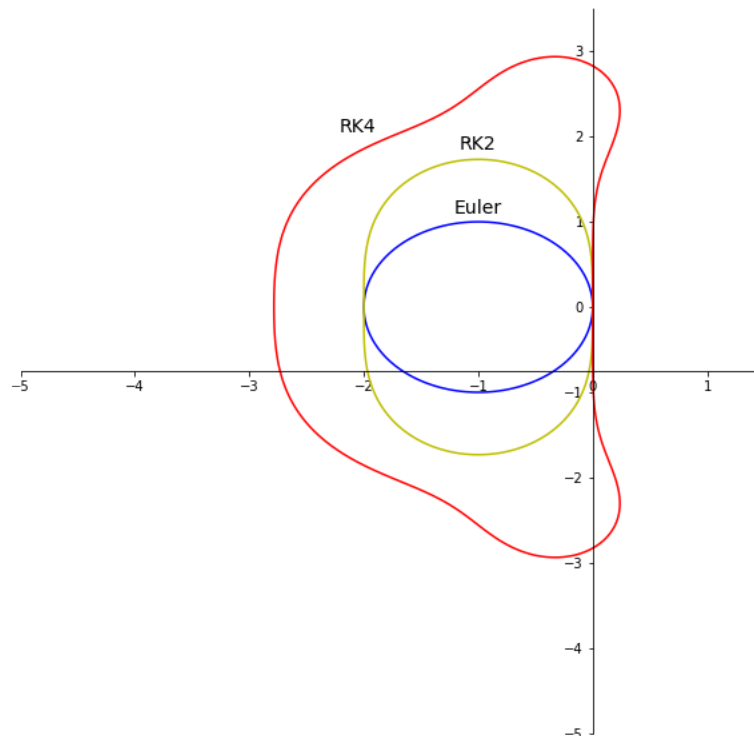
- $K_1$  is the increment calculated using  $y$  and the slope at the start of the interval.
- Using  $(y + hk_1/2)$ ,  $K_2$  is the increment depending on the slope at the interval's midpoint.
- Using  $y + hk_2/2$ ,  $k_3$  is the increment based on the midpoint slope once more.
- Using  $y + hk_3$ ,  $k_4$  is the increment depending on the slope at the end of the period.

#### Coding

An implementation of both methods can be found [here](#).

### 4 Stability

Below is a comparison graph that shows the areas of stability for the several methods that I have previously covered.



Notes: the forward Euler method is unstable if one of the system's eigenvalues is completely imaginary. Even though the stability domain of the RK2 scheme is wider, the system possesses the same feature. RK4 has the widest stability domain of all the schemes examined to yet, and more crucially, it encompasses a portion of the imaginary axis, therefore it can handle issues with completely imaginary eigenvalues.

## 5 Explicit and Implicit RK

For explicit RK schemes,  $h = s - 1$ . This means that any  $k_i$  may be explicitly calculated using the previously calculated  $k_j$ , where  $j < i$ . For implicit RK schemes,  $h = i$  exists. It means that we must solve the implicit equation  $k_i = f(k_1, k_2, \dots, k_i)$  for each  $k_i$ . The complexity is the cost we must pay, yet we must find a way to solve the preceding equation. However, implicit RK approaches are more stable than explicit ones and may thus be applied to a broader range of issues.