RUNGE-KUTTA METHODS

http://www.phys.uconn.edu/~rozman/Courses/m3510_19f/

Last modified: December 2, 2019

Introduction

We are interested in the numerical solution of the initial value problem (IVP)

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y), \quad a \le t \le b, \quad y(a) = y_0. \tag{1}$$

We denote the time at the nth time step by t_n, the computed solution at the nth time step by y_n,

$$y_n \equiv y(t_n), \tag{2}$$

and the value of the right hand side of Eq. (1) at the nth time step by f_n,

$$f_n \equiv f(t_n, y_n). \tag{3}$$

The step size h (assumed to be constant for the sake of simplicity) is

$$h = t_n - t_{n-1}. \tag{4}$$

The error that is induced at every time-step, ϵ, is referred to as the *local truncation error* (LTE) of the method. The local truncation error is different from the *global error* g_n, which is defined as the absolute value of the difference between the true solution and the computed solution,

$$g_n = \left| y_{\text{exact}}(t_n) - y_n \right|. \tag{5}$$

In most cases, we do not know the exact solution and hence cannot evaluate the global error. However, if we neglect roundoff errors, it is reasonable to assume that the global error at the nth time step is n times the LTE. Since n is proportional to $\frac{1}{h}$, g_n should be proportional to $\frac{\epsilon}{h}$. A method with $\epsilon \sim h^{k+1}$ is said to be of kth order. This implies that for a kth order method, the global error scales as h^k.

Second order Runge-Kutta method

Runge-Kutta (RK) methods is a class of methods that uses the information on the slope at more than one point to find the solution at the future time step. Let's derive the second order RK method where the local truncation error $\epsilon \sim h^3$.

Given the IVP of Eq. (1), the time step h, and the solution y_n at the nth time step, we wish to compute y_{n+1} in the following form:

$$y_{n+1} = y_n + a\,hf(t_n, y_n) + b\,hf(t_n + \alpha h, y_n + \beta h), \tag{6}$$

where the constants α, β, a, and b have to be selected so that the resulting method has a local truncation error $O(h^3)$.

The Taylor series expansion of $y(t_{n+1})$ about t_n correct up to the h^2 term is as following,

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + h\,\frac{\mathrm{d}y}{\mathrm{d}t}\bigg|_{t_n} + \frac{h^2}{2}\,\frac{\mathrm{d}^2y}{\mathrm{d}t^2}\bigg|_{t_n} + O(h^3). \tag{7}$$

From Eq. (1),

$$\frac{\mathrm{d}^2y}{\mathrm{d}t^2} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}f(t,y) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\,\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\partial f}{\partial t} + f\,\frac{\partial f}{\partial y}. \tag{8}$$

From Eqs. (1), (7), and (8)

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{1}{2}h^2\,\frac{\partial f}{\partial t}\bigg|_{t_n,y_n} + \frac{1}{2}h^2 f(t_n, y_n)\,\frac{\partial f}{\partial y}\bigg|_{t_n,y_n} + O(h^3). \tag{9}$$

On the other hand, the Taylor series expansion of the expression Eq. (6) about t_n correct up to the h^2 term is as following,

$$y_{n+1} = y_n + (a+b)\,hf(t_n, y_n) + b\alpha\,h^2\,\frac{\partial f}{\partial t}\bigg|_{t_n,y_n} + b\beta\,h^2\,\frac{\partial f}{\partial y}\bigg|_{t_n,y_n} + O(h^3). \tag{10}$$

Comparing the terms with identical powers of h in Eqs. (9) and (10), gives us the following system of equations to determine the constants:

$$\begin{aligned} a + b &= 1, \\ \alpha\, b &= \frac{1}{2}, \\ \beta\, b &= \frac{1}{2}f(t_n, y_n). \end{aligned} \tag{11}$$

There are infinitely many choices of a, b, α and β which satisfy Eq. (11). If we choose $a = b = \frac{1}{2}$, $\alpha = 1$, and $\beta = f(t_n, y_n)$ we get the classical second order accurate Runge-Kutta method (RK2) which is summarized as follows:

$$\begin{aligned} k_1 &= hf(t_n, y_n), \\ k_2 &= hf(t_n + h, y_n + k_1), \\ y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2). \end{aligned} \tag{12}$$

If we choose $a = 0$, $b = 1$, $\alpha = \frac{1}{2}$, and $\beta = \frac{1}{2} f(t_n, y_n)$ we get the second order accurate Runge-Kutta method known as midpoint method:

$$\begin{aligned} k_1 &= hf(t_n, y_n), \\ k_2 &= hf(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}), \\ y_{n+1} &= y_n + k_2. \end{aligned} \tag{13}$$

Higher order Runge-Kutta methods

Runge-Kutta methods of higher order can be derived in a similar manner.

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i, \tag{14}$$

where k_i are given by

$$\begin{aligned} &k_1 = f(t_n, y_n), \\ &k_2 = f(t_n + c_2 h, y_n + a_{21} k_1), \\ &k_3 = f(t_n + c_3 h, y_n + a_{31} k_1 + a_{32} k_2), \\ &\quad \vdots \\ &k_s = f(t_n + c_s h, y_n + \sum_{j=1}^{s-1} a_{sj} k_j), \end{aligned} \tag{15}$$

and

$$c_i = \sum_{j=1}^{s-1} a_{ij}. \tag{16}$$

The choice of the constants c_i, a_{ij} and b_i uniquely determines a specific Runge-Kutta (RK) method. A systematical way of presenting those coefficient is called the Butcher's tableau (See Table 1).

For example, the two-stage, second-order classical Runge-Kutta methods Eq. (12) is represented as following:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

Table 1: The Butcher tableau for the explicit Runge–Kutta method.

whereas the midpoint method (13) is represented as:

$$
\begin{array}{c|cc}
0 & & \\
\frac{1}{2} & \frac{1}{2} & \\
\hline
 & 0 & 1
\end{array}
$$

For the reference, the fourth order Runge-Kutta method (RK4) is as following:

$$
\begin{aligned}
k_1 &= hf(t_n, y_n), \\
k_2 &= hf(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}), \\
k_3 &= hf(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}), \\
k_4 &= hf(t_n + h, y_n + k_3), \\
y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).
\end{aligned} \tag{17}
$$

Its Butcher tableau is:

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}
$$