Pen and paper

1. Consider the general form of an explicit two-stage Runge–Kutta scheme:

$$y_{n+1} = y_n + b_1 k_1 + b_2 k_2, \tag{1}$$
$$k_1 = hf(t_n, y_n), \tag{2}$$
$$k_2 = hf(t_n + ch, y_n + ak_1) \tag{3}$$

   (a) Express the truncation error in terms of a, b_1, b_2, and c.

   (b) Show that the scheme becomes consistent with the initial value problem (6) if and only if $b_1 + b_2 = 1$.

   (c) Derive the conditions that the scheme becomes a second-order method. Confirm that the explicit midpoint scheme and the explicit trapezoid scheme satisfy these conditions.

2. Consider a three-stage Runge–Kutta scheme whose Butcher table is give as follows:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & \frac{2}{3}-a & a & \\ \hline & b_1 & b_2 & b_3 \end{array} \tag{4}$$

   Express b_1, b_2, and b_3 in terms of a from the order conditions (assuming this scheme is a third-order method). *Note: Since the simplifying assumption is applied, you can obtain order conditions from $y' = f(y)$ instead of $y' = f(t, y)$.*

3. Consider a Runge–Kutta scheme whose Butcher table is give as follows:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{3} & \frac{1}{3} & & & \\ \frac{2}{3} & -\frac{1}{3} & 1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array} \tag{5}$$

   (a) Write down the scheme to solve the initial value problem:

$$y' = f(t, y), \quad y(t_0) = y_0 \tag{6}$$

   (b) Suppose we use this scheme to solve the following initial value problem:

$$y' = f(t), \quad y(0) = 0 \tag{7}$$

   Express the result of $y_1 \approx y(h)$ in terms of $f(0)$, $f(\frac{1}{3}h)$, $f(\frac{2}{3}h)$, $f(h)$.

   (c) Compute $y_1 \approx y(h)$ for (7) using the RK4 scheme. By noting $y(h) = \int_0^h f(t)dt$, the result from the RK4 scheme corresponds to Simpson's first rule, whereas the result from (5) corresponds to Simpson's second rule (or $\frac{3}{8}$ rule).

MATLAB

1. (a) Write a MATLAB function
`function [tvals,yvals] = RK4_1d(F,t0,T,y0,n)`
that computes the solution $y(t)$ $(t_0 \le t \le T)$ to the initial value problem

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0. \tag{8}$$

using the RK4 method. Note that `tvals` contains points $t_k = t_0 + k\Delta t$ $(k = 0, 1, \cdots, n)$ with $\Delta t = (T - t_0)/n$ and `yvals` contains the values of the approximate solution at these points.

(b) Validate your code by using the following initial value problem:

$$y' = \left(1 - \frac{4}{3}t\right)y, \quad y(0) = 1. \tag{9}$$

In other words, (i) plot the approximate solutions $y(t)$ $(0 \le t \le 3)$ for $n = 4$ and 8 with the exact solution $y(t) = \exp(t - \frac{2}{3}t^2)$ and (ii) draw a log-log plot of the error at the final point $t = T$ versus the number of subintervals n. Observe how the errors decrease.

2. (a) By *modifying* `RK4_1d`, write a MATLAB function
`function [tvals,uvals] = RK4(F,t0,T,u0,n)`
that solves the *vector-valued* initial value problem using the RK4 method:

$$\mathbf{u}'(t) = \mathbf{F}(t, \mathbf{u}(t)), \quad \mathbf{u}(t_0) = \mathbf{u}_0. \tag{10}$$

(b) Validate your code by using the following initial value problem:

$$\frac{d}{dt}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_2 \\ -(a^2+b^2)u_1 + 2au_2 \end{bmatrix}, \quad \begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ a \end{bmatrix}. \tag{11}$$

Choose $a = -0.1$, $b = 2$, $t_0 = 0$, $T = 5$.

3. (a) Write a *wrapper* MATLAB function
`function [tvals,yvals] = solve_2nd_ode_by_RK4(a2,a1,f,t0,T,y0,dy0,n)`
that computes the solution $y(t)$ $(t_0 \le t \le T)$ to the initial value problem

$$a_2(t)y''(t) + a_1(t)y'(t) + f(t, y(t)) = 0, \quad y'(t_0) = dy_0, \quad y(t_0) = y_0. \tag{12}$$

by *internally calling* `RK4`.

(b) Solve the following initial value problem:

$$y'' + \frac{1}{10}y' + 10\sin y = 0, \quad y(0) = 3, \quad y'(0) = 0 \tag{13}$$

Plot the solution $y(t)$ $(0 \le t \le 10)$ for $n = 50$, 100, and 200.