Numerical Methods
Exercise Sheet 6

I. Bezshyiko, A. Pattori
`http://www.physik.uzh.ch/en/teaching/PHY233/HS2016.html`

Issued: 16.11.2016
Due: 23.11.2016

Exercise 1: Explicit Runge-Kutta methods (120 Pts.)

In this exercise you will be asked to find numerically the solution to the following ODE:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y) = 8\,(1-2x)\,y\ , \qquad y(0) = e^{-2}\ . \tag{1}$$

You will do it using the so-called Bogacki–Shampine method, which makes use of the following two Butcher tableaus:

Tableau A (2nd order):

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{3}{4} & 0 & \frac{3}{4} & & \\ 1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\ \hline & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 \end{array}$$

Tableau B (3rd order):

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{3}{4} & 0 & \frac{3}{4} & & \\ 1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\ \hline & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{8} \end{array}$$

Tableau B will be used to extract $y_{n+1} \equiv y^B_{n+1}$, while tableau A will be used to estimate the error on y_{n+1} through the difference $\delta y_{n+1} = y^B_{n+1} - y^A_{n+1}$.

Please notice that tableau A and B differ only by the last row.

a) Implement the two tableaus A and B in a function

$$\Phi(x_n, y_n, h) \tag{2}$$

which returns:

- The point $\{x_{n+1}, y_{n+1}\}^B$, computed using the tableau B (3rd order).
- The value $\delta y_{n+1} = y^B_{n+1} - y^A_{n+1}$, defined as the difference in the prediction for y_{n+1} between the tableaus B and A.

(20 points)

Optional: make the ODE function $f(x,y)$ and the Butcher tableaus themselves input parameters of Φ, in order to give more flexibility to your code. (20 points)

b) As explained, δy_{n+1} gives you an estimate of the error with which y^B_{n+1} is extracted. Compute $\delta y_1(h)$ for $\{x_0, y_0\} = \{0, e^{-2}\}$ and different values of h (e.g. $h = 0.01, 0.02, \ldots, 0.1$). Use these results to show that your error indeed scales as h^3, as expected by the order of the method. (20 points)

c) *Fixed step size method.* We want to solve eq. (1) in the range $x \in [0, 2]$. We will first do it using a fixed step size. Set $h = 0.1$ and solve ODE (1) using your function $\Phi(x_n, y_n, h)$.

Plot the result and the errors δy_{n+1} of each step (you may do it in two different plots). (20 points)

d) *Adaptive step size method.* Suppose we want to impose $\delta y_n < 10^{-4}$ at every step. A possibility could be to reduce uniformly the step size h until we reach such goal. This is not an efficient solution and sometimes it is overwhelmingly time-consuming.

A more interesting solution is the adaptive step size method, in which you adjust h at each step accordingly to the estimated error δy_{n+1}.

Implement a simple adaptive step size method in this way:

1- Start with $h = 0.1$

2- At every step $n \to n + 1$, evaluate δy_{n+1}:

- If $\delta y_{n+1} \in (10^{-5}, 10^{-4})$, keep the result and go to next step
- If $\delta y_{n+1} > 10^{-4}$, reduce h accordingly:

$$h \to 0.9 \sqrt[3]{\frac{10^{-4}}{\delta y_{n+1}}}\, h$$

  then compute again step $n \to n + 1$

- If $\delta y_{n+1} < 10^{-5}$, increase h accordingly:

$$h \to 1.1 \sqrt[3]{\frac{10^{-5}}{\delta y_{n+1}}}\, h$$

  and go to the next step

Solve ODE (1) for $x \in [0, 2]$ with this method. (20 points)

Compare the result and the errors δy_n with what obtained in task (1c). (10 points)

Also, keep tracks of the adapted h_n of each step, plot and comment them. (10 points)

Maximum number of points for mandatory tasks on 23.11 : 100

Maximum possible number of points for tasks on 23.11 : 120