

# Numerical Analysis II

## Homework Sheet 3

### Exercises Tutorial on May 5

**1. Problem**

Give all Runge-Kutta-Methods with consistency order  $p \geq 2$  that have the following Butcher array:

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & c_2 & & \\ c_3 & 0 & c_3 & \\ \hline & 0 & 0 & 1 \end{array} .$$

**2. Problem**

Explain the ideas of Richardson extrapolation and embedded Runge-Kutta-Methods. Moreover, apply these two methods on the test equation  $\dot{y}(t) = y(t)$ ,  $y(0) = 1$ , with step size  $h = 0.1$ . Use the forward Euler scheme for the Richardson extrapolation and both forward Euler and Heun's method in the embedded Runge-Kutta-Method.

**3. Problem**

Show that Collatz' method can be interpreted as two Euler steps with step size  $h/2$ , followed by a Richardson extrapolation.

### Theoretical Homework Due: May 13, during the lecture

**1. Problem**

(10 Points)

Consider the two test problems

(a)

$$\dot{y}(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} y(t), \quad y(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{with exact solution } y(t) = \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix},$$

(b)

$$\dot{y}(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} y(t), \quad y(0) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{with exact solution } y(t) = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos(t) - \sin(t) \\ \cos(t) + \sin(t) \end{bmatrix}.$$

For each test problem, compute two steps using the Richardson extrapolation from the lecture, i.e. compute  $u_h, u_{h/2}, u_{2h/2}$ , the estimated truncation error  $\tilde{\tau}$ , and the extrapolation  $w$  for two steps of each test problem. To start the second step, use the previously computed  $u_{2h/2}$  for each test problem.

For your computations, use the forward Euler method, the step size 0.1 (thus,  $h/2 = 0.05$ ), and 4-digit precision arithmetic i.e. use the numbers  $0.0000, \pm 0.0001, \pm 0.0002, \dots, \pm 0.9999, \pm 1.0000$ .

Compare the values you calculated to the exact solutions of the test problems. Is the  $\tilde{\tau}$  you received an accurate estimate for the error made in each step? Explain the difference in the components of the estimated truncation error! Does the size of the estimated truncation error justify different step sizes for the two test problems?

**2. Problem**

(5 Points)

Consider again the test problems (a) and (b). For each test problem, estimate the truncation error of  $u_h$  (the same  $u_h$  that you previously computed using forward Euler).

This time, use Heun's method (consistent of order 2) to get the  $\hat{u}_h$  defined in the lecture to compute the estimate  $\tilde{\tau}$ . Don't forget to do two steps for each test problem and compare your results to the Richardson extrapolation.

# Programming Homework

Due: May 12 (first chance) or May 19 (second chance)

Attention: You can only submit your program on May 19 if you presented a programming approach on May 12!

Write a program that solves an ordinary differential equation  $\dot{y}(t) = f(t, y(t))$ ,  $y(t_0) = y_0$  on an interval  $[t_0, t_0 + a]$  using an embedded Runge-Kutta method with step size control. The method should be defined by a general Butcher array of the form

$$\begin{array}{c|ccc}
 0 & & & \\
 \alpha_2 & \beta_{2,1} & & \\
 \vdots & \vdots & \ddots & \\
 \alpha_s & \beta_{s,1} & \cdots & \beta_{s,s-1} \\
 \hline
 & \gamma_1^{(1)} & \cdots & \gamma_{s-1}^{(1)} & \gamma_s^{(1)} \\
 & \gamma_1^{(2)} & \cdots & \gamma_{s-1}^{(2)} & \gamma_s^{(2)}
 \end{array}$$

and be called with the line

$$[\mathbf{h}, \mathbf{t}, \mathbf{u}] = \text{embrk}(\text{fun}, \text{but}, \mathbf{t0}, \mathbf{y0}, \mathbf{h0}, \mathbf{a}, \mathbf{p}, \text{eps}).$$

Here, **fun** should be a MATLAB function handle corresponding to the right hand side  $f(t, y)$  of the differential equation. It should also be possible for **y** and **f** to be vectors of  $\mathbb{R}^n$ .

Furthermore, **but** should be a Butcher array generated by the routine

$$[\text{but}] = \text{butcher}(\mathbf{i}),$$

where the parameter **i** determines the size and shape of the array. For **i=1**, you should get the Butcher array of the Runge-Kutta-Fehlberg 2(3) method

$$\begin{array}{c|cc}
 0 & & \\
 1 & 1 & \\
 1/2 & 1/4 & 1/4 \\
 \hline
 & 1/2 & 1/2 \\
 \hline
 & 1/6 & 1/6 & 4/6
 \end{array}$$

and **i=2** should give the Runge-Kutta-Fehlberg 4(5) method printed on the next page. It should also be possible to call your program with a Butcher array of arbitrary size.

The parameter  $\mathbf{t0} = t_0$  is the lower interval bound,  $\mathbf{y0} = y_0 \in \mathbb{R}^n$  is the initial value,  $\mathbf{h0}$  is the starting step size, and  $\mathbf{a} = a$  is the interval length. Further, **p** is the order of consistency of the lower order method and **eps** is the tolerance for the relative and absolute error in each step.

Hence, for your computations use an appropriate formula for the error  $\text{err}(\tilde{\tau}, \text{tol})$ , where

$$\text{tol}_i = \text{eps} \left[ 1 + |(u_m)_i| \right].$$

The routine should return the vector of step sizes  $\mathbf{h} = [h_0, h_1, \dots, h_{N-1}]$ , the vector of grid points  $\mathbf{t} = [t_0, t_1, \dots, t_N]$ , and the corresponding approximated solution  $\mathbf{u} = [u_0, u_1, \dots, u_N]$ .

Compare the two methods using the test problems

1.  $\dot{y}(t) = 2y(t) - e^t$ ,  $y(0) = 2$ ,  $t \in [0, 1]$ ,  
with exact solution  $y(t) = e^t + e^{2t}$ ,
2.  $\dot{y}(t) = -\tan(t)y(t)$ ,  $y(0) = 1$ ,  $t \in [0, 6]$ ,  
with exact solution  $y(t) = \cos(t)$ ,
3.  $\dot{y}(t) = -\tan\left(\frac{1}{1.05-t}\right) \frac{y(t)}{(1.05-t)^2}$ ,  $y(0) = \cos(1)$ ,  $t \in [0, 1]$ ,  
with exact solution  $y(t) = \cos\left(\frac{1}{1.05-t}\right)$ .

Run both methods on each test problem and experiment with varying the parameters  $\mathbf{h0}, \text{eps} \in \{10^{-k} \mid k = 1, 2, 3, 4, 5\}$  and plot the computed approximations in graphs alongside the exact solutions in a way that makes the number of time steps / grid points visible. Explain the accumulation of grid points in some parts of the graphs! How can the behavior for  $\mathbf{h0} = 0.001$  and  $\text{eps} = 0.1$  be explained?

Runge-Kutta-Fehlberg 4(5):

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	-7200/2197	7296/2197			
1	439/216	-8	3680/513	-845/4104		
1/2	-8/27	2	-3544/2565	1859/4104	-11/40	
	25/216	0	1408/2565	2197/4104	-1/5	0
	16/135	0	6656/12825	28561/56430	-9/50	2/55