Numerical Analysis of Ordinary Differential Equations Programming Exercises

Summer semester 2016

Universität Heidelberg - IWR
Prof. Dr. Guido Kanschat, Dr. Dörte Beigel,
Stefan Meggendorfer, Philipp Siehr

Exercise sheet 2
Until: 02.06.2016

Exercise P2.1 (**Adaptive Refinement**)
In this exercise we want to experience the difference between solving an IVP on an equidistant grid compared to the adaptive refinement algorithm. We will test both implementation on the Lotka-Volterra equations:

$$u_1'(t) = u_1(t)(a - bu_2(t))$$
$$u_2'(t) = u_2(t)(cu_1(t) - d)$$

with $u(0) = (1,1)^\top$, $(a,b,c,d) = (5,2,2,1)$, $I = [0,10]$.

a) Implement the function:

```
[u,uhat] = rk_dp_step(u0,t0,h,f)
```

This function will compute 1(!) step of Dormand-Prince 45:

$$u_1 = u_0 + h \sum_i b_i k_i$$
$$\hat{u}_1 = u_0 + h \sum_i b_i k_i$$

The Butcher tableau can be found here:
`https://en.wikipedia.org/wiki/Dormand-Prince_method`
Remark: If you have difficulties starting this task, have a look at the function `rk_general(A,B,C,u0,t0,tn,n,f)` on the webpage. This function creates the whole solution u (see Exercise P1.2).

b) Calculate the solution with $h = 10^{-3}$ and plot the solution with respect to I.

c) Implement the adaptive stepsize algorithm (Algorithm 2.4.2). Since Dormand-Prince 45 is an embedded Runge-Kutta method it will provide both u_1 and $\hat{u}_1$ for the estimate of the local error: $\|u_1 - \hat{u}_1\|$.

d) Use the tolerance $\varepsilon = 10^{-4}$ and start with $h = 1/10$ to calculate the solution. Plot the solution with respect to I. Keep in mind, that you have to scale the vector representing the x-axis according to the adaptive refinement.

e) Compare the effort of the adaptive solver to the equidistant solver. (How many Dormand-Prince evaluations ? How many steps needed to be recalculated? Is the effort twice as large, since you calculate u and $\hat{u}$? [...])
Plot the adaptive step size on the intervall I. Discuss the results.