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 1
Until: 19.05.2016

We have prepared an example written in C++ and MATLAB. You can download these from the webpage. The example solves the equation:

$$u'(t) = \lambda u(t), t \in [0, 1]$$
$$u(0) = u_0$$

using the explicit Euler method.

The C++ file can be compiled by:

```
g++ -std=c++11 -o ex01_example ex01_example.cc
```

After that, execute the program with `./ex01_example`
We used the concept of `std::function` instead of classes, which would be better but more complex. Further examples with the `std::function` can be found here:
`http://de.cppreference.com/w/cpp/utility/functional/function`

The MATLAB Code runs with the command `main`, using either MATLAB or Octave.

If you have problems solving the exercises, start with the following tasks using the example code:

- Change the interval $[0, 1]$ to $[0, 4]$.
- Change the code, such that the explicit Euler is called with h instead of n.
- Change the right-hand-side $f(t, u) = \lambda u$ to something more difficult, such as $f(t, u) = t \cdot \sin(u)$?

You can solve the following exercises by modifying the example code or by writing your own code.
The deadline for this exercise sheet is the **19.05.2016**.

Exercise P1.1 (**Lotka-Volterra equations**)
The Lotka-Volterra equations are of the following form:

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

with the initial value $u(0) = (u_{1,0}, u_{2,0})^\top$ and parameters $a, b, c, d \in \mathbb{R}$.

a) What do these equations model ? Inform yourself with the help of Wikipedia.

b) Write a new function

`erg = rhs(t,(u1,u2))`

that represents the right-hand-side of these equations. The input (u_1, u_2) and the return value should be vector-valued. The parameters a, b, c and d should be stored within that function.
Remark: In C++ you can use the `std::vector` container to store (u_1, u_2).
`http://www.cplusplus.com/reference/vector/vector/`

c) Use the parameters: $u(0) = (1, 1)^\top$, $(a, b, c, d) = (5, 2, 2, 1)$, $I = [0, 10]$, $n = 100$. Solve the Lotka-Volterra equations with the explicit Euler method. What is the value of $u_h(t_n)$?

d) Modify the (or your) `expl_euler` function, so that it returns the numerical solution u_h instead of just the value $u_h(t_n)$. Plot the solution u_h with the parameters of c).

Remark (Plots): If you are using Matlab or Octave the following link will explain you how to plot data:
`http://de.mathworks.com/help/matlab/ref/plot.html`
If you are using C++ then Gnuplot is a good tool to plot functions. You need to create a file with the structure:

$$\begin{array}{ccc} t_0 & u_{1,h}(t_0) & u_{2,h}(t_0) \\ t_1 & u_{1,h}(t_1) & u_{2,h}(t_1) \\ \vdots & \vdots & \vdots \\ t_n & u_{1,h}(t_n) & u_{2,h}(t_n) \end{array}$$

After starting Gnuplot with `gnuplot` you can print the data from this file with:

```
set style data lines
plot ''filename'' using 1:2
plot ''filename'' using 1:3
```

How can you plot both functions in a single plot ?

Exercise P1.2 (**Runge-Kutta Solver**)
Now we want to solve the Lotka-Volterra Equations with a different explicit method.

a) Implement the general Runge-Kutta method:

```
u_h = runge_kutta(A,b,c,u0,t0,tn,n,f)
```

This function gets the same arguments as `expl_euler` in the example code. Additionally the method needs to get a matrix `A` and two vectors `b`, `c` which define the Runge-Kutta method. (See Defintion 2.3.5 Butcher tableau.)

Remark: The general form of an explicit Runge-Kutta method is given in Definiton 2.3.2 of the lecture notes.
Remark: In C++ you can store `A` with a vector of vectors:

```
std::vector< std::vector<double > > A
```

b) Solve the Lotka-Volterra equations from Exercise P1.1c) with the classical Runge-Kutta method of fourth order (Example 2.3.18). And plot the solution.