Mathematical Modelling - Assignments week 4:

Numerical Methods for Differential Equations

Assignment 1: Differential equations

Given is the following differential equation (DE) and boundary condition (BC):

$$\frac{dy}{dt} = y - 3t,\ y(0) = 5$$

Calculate the analytical solution of this DE and BC.

Write a Matlab live script `assignments_week4.mlx`, in which you plot this solution for the time interval [0,1].

Assignment 2: Euler with Matlab

Given is the following header the function Euler:

```
function [t,y] = Euler(deriv_func, t_end,dt,t0,y0)
% Function to approximate a first order DE by the Euler method
% The DE has the form
%
%               dy/dt = f(t,y)
%
% Inputs:
% deriv_func: a Matlab function in which the right hand side
%             of the DE has been programmed.
% t_end: time until which the DE has to be solved
% dt: step size
% t0: start time
% y0: initial value of the solution
%
% Outputs:
% t: array with times at which the solution has been calculated
% y: array with the values of the numerical approximation.
```

One of the inputs needed by the Euler function is the expression f(t, y).
This expression needs to be programmed in a separation function file.
An example of such a file looks like:

```
function dy = righthandside(t,y)
   dy = t.*y;
end
```

NB: this is just an example of the structure of such a file and not a correct answer!

Make a Matlab script in which you call the Euler function to solve the DE.
Calling the Euler function goes like:

```
t0 = 0;
t_end = 1;
dt = 0.1;
y0 = 1; % NB: this is just an example value, not the correct one
[t,y] = Euler(@righthandside, t_end,dt,t0,y0);
```

NB: this is just an example, you might need to change the values of the variables

We are now going to solve the DE of the previous exercise by means of Matlab.
(a) Write a Matlab function `righthandside` to implement the righthand side of the DE.
(b) Write the Euler function itself in the matlab file `Euler.m`.
(c) Extend the Matlab live script `assignments_week4.mlx` with code for solving the DE for the end time t = 1.

- In this script you should start time t = 0 and step size dt = 0.1.
- Plot the solution using markers (e.g. circles). Don't draw a connection line through the points of the solution.
- The analytical solution has been given in the previous assignment. Evaluate this solution at the same time values as the Euler solution.
- Plot the analytical solution in the same figure, use this time a line and no markers.
- Determine now the Euler solution for the step sizes 0.05 and 0.025, and plot these in the same figure.
- Calculate the error ratios $\frac{E_{dt=0.05}}{E_{dt=0.1}}$ and $\frac{E_{dt=0.025}}{E_{dt=0.05}}$ for t=1 by means of matlab code.
  Don't use a semicolon in the final result, so that the output will be printed

Assignment 3: Heun with Matlab

Answer the same questions as in assignment 2, but now with the Heun method (so you should use the step sizes 0.1, 0.05 and 0.025. Of course you can reuse the Matlab function `righthandside`.

So you have to implement a new function Heun in the file `Heun.m`, which has the same parameters as the Euler function, but of course a different implementation.

To compare the Heun method with the Euler method, we will solve again the DE of the first assignment

Continue to work in the Matlab live script `assignments_week4.mlx` which you have written in the precious assignment. Add an extra line in your Matlab live script in which you solve the DE by means of the Heun method.

Plot the solutions (with different markers) in the same figure, together with the analytical solution.

Don't forget to calculate the error ratios too.

Assignment 4: The Matlab command ode45

In practice the Euler method and even the Heun method are often not precise enough. To obtain a sufficiently high precision, the step size has to be taken very small, which is most of the times not very practical. Matlab has very advanced functions available to solve DE which are much more precise. The working horse of these routines is the function ode45. This is a fourth order method (hence the 4 in its name). This means that halving the time step, makes the error $2^4 = 16$ times smaller.
The ode45 function solves first order DE of the form

$$\frac{dy}{dt} = f(t, y)$$

The syntax to use this function is as follows:

```
[t,y] = ode45(@righthandside, timedomain, initialvalues, options)
```

where

- **`righthandside`** is again a Matlab function file, in which the expression f(t, y) has been programmed.
- **`timedomein`**. This input variable can be used in two ways:
    1. an array with two values; the first is the starting time and the second the end time. So e.g. timedomain = [0 ,1] defines the start time as t = 0 and the end time as t = 1.
    2. An array with times at which we like to know the solution. e.g. timedomain = [0:0.1:1].
- **`initialvalues`** is an array with initial values. A first order DE needs only one initial value.
- **`options`**: this is not needed, but here some extra options can be indicated like the wished precision.

In this assignment we are going to use this built-in Matlab function.
Add a line in your Matlab script file in which you solve the DE of the first assignment, now using the ode45 method. Take again as end time 1. Use as time domain = [t0 t_end].
Plot the solution (using markers) in the same figure as where the Euler and Heun solution have been plotted.

You should submit the Matlab live script as well as Matlab files for the functions you wrote at the blackboard submit point, as well as the generated pdf for the Matlab live script.