# Mathematics - Numerical Analysis

Problem Sheet 1: Euler's Method

Xin Wang

September 14, 2020

## 1 Introduction

Here is the Matlab code used to evaluate Euler's method, seen in class.

```
x=0; % set initial value of x0
y=2; % set y initial condition y at x0
h=0.03; % set step-size
xf=1; % set final value of x
N=round((xf-x)/h); % nr of steps:  (interval size)/(step size)
plot(x,y,'*'); % plot initial condition
hold on; % figure open for more data

for i=1:N % loop for N steps
    y=y + h*(x+y); % next value of y
    x=x+h; % increase x by stepsize
    plot(x,y, 'b*') % plot now values of x,y
end

xx=0:h:xf; % same interval, stepsize
yy=3*exp(xx)-xx-1; % calculate exact solution
plot(xx,yy,'b'); % plot exact solution
```

## 2 Exercise 1

Repeat the procedure of the trapeze examples with the Euler method for the differential equation:

$$\frac{dy}{dx} = x + y \text{ with initial condition } y(0) = 2$$

on $[0, 1]$, whose solution is in the notes. Experiment first with the above code, see if you can come up with improvements.

- Consider the output you want. In the trapeze example we compared two single numbers, the exact and approximate values of the integral. In this case we are approximating the value of the function $y(x)$ for $x \in [0, 1]$. In fact, we want to plot $y(x)$ on $[0, 1]$, as the very simple code above does. Similarly, for error analysis, we have to compare the estimated values to the exact solution at the correct value of $x$ in the interval $[0, 1]$, as the simple code above does.

  You will need to compare two arrays of values, obtaining an array of errors. For this function we expect the maximum error at the end, at $x = 1$ because $y(x)$ is monotone increasing on the interval considered. This cannot be assumed: The error we seek needs to be the maximal error found.

1. Declare arrays of values $y$ and $x$ of length $N$ (see above) with entries $x_i$ for $i = 0...(N-1)$ and $x_0$, $y_0$ is the initial condition. The entries for the x-array should have spacing $h$. Use the Euler method to fill in the values of $y$ for each $x_i > x_0$.
2. Declare a function handle $f(x, y)$ so you can use the function in the loop. (This will make it easier in $(b)$ to pass a function as an argument in the function call to the matlab function *euler.m*) The syntax for *feval* extends logically from functions of one variable to functions of two variables.
3. Calculate an array of exact solutions using $y = 3e^x - x - 1$, ensuring that it matches the entries in the x-array.
4. Now you can subtract the two solution arrays and obtain the error as a function of $x$. Try plotting this against $x$.
5. Calculate the maximal error. Decrease $h$ and repeat. Confirm that the error is $O(h)$.

- Now write the *euler.m* function, similar to *trapez.m*. It's output needs to be two array of values $x$ and $y$. This must include the first line: $function[x, y] = euler(f, ?, ?, ...)$. This should implement the Euler method for the function $y' = f(x, y)$, given an initial condition.

  You must first decide what arguments the function euler requires. One of these will have to be the function $f$ from the differential equation, hence try $function[x, y] = euler(f, ?, ?, ...)$ The initial condition might be useful: one or two arguments? Also consider, step-size $h$, and the final value of $x$.

- Now write a second file, a script which has lines of code calling *euler.m*, and some other lines, like the script you see in *trapez-second*.

  In the script you should alsocreate an array of exact solutions *exact(x)* corresponding to the same values of $x$ which the *euler.m* function returns. This needs to be done as in *(a)*.

- Now do an error analysis similar to that done in *trapez-second*. We know to expect an error of $O(h)$ so you know what the gradient of the log/log plot should be. Think about:

  1. How you would decrease the step-size in a sensible way.
  2. For each step-siz to do an error analysis as in (c) and obtain the log/log plot.

# 3 Exercise 2

Repeat all of the above with another differential equation of the form $y' = f(x, y)$ wit initial condition $y(x_0) = y_0$.

1. Find your own, or try $y' + y = e^x$ with initial condition $y(0) = 0$.

   This is linear, first order, easy to solve!

2. Apply Euler's method, compare with the exact solution, carry out the error analysis.