Numerical Methods
MATH 417 - 501
A. Bonito

April 18
Spring 2018

First Name: ____________________ **Last Name:** ____________________

Homework 10

Exercise 1 **25% (MATLAB)**

Denote by $[0,T]$ the time interval we are interested in solving the first order equation

$$\frac{d}{dt}y = f(y,t), \qquad y(0) = y_0.$$

Any numerical method start with a partition of $[0,T]$ into sub-intervals. For this, let N be a positive integer and define $h = T/N$ and $t_n = nh$ for $n = 0, ..., N$. This creates a *uniform* partition $0 = t_0 < t_1 < ... < t_N = T$ of $[0,T]$. The goal of numerical methods is to provide approximations Y_n of the exact solution $y(t_n)$, $n = 0, ..., N$.
The performance of each particular numerical algorithm is measured by computing (for instance)

$$\max_{n=0,\dots,N} |y(t_n) - Y_n|$$

and in particular how the above quantity behaves with N, the number of steps in the method.
For the Euler method, there exists a constant C independent of N and the solution such that

$$\max_{n=0,\dots,N} |y(t_n) - Y_n| \leqslant CN^{-1} \sup_x \left| \frac{d^2}{dx^2} y \right|,$$

provided the exact solution y is twice differentiable. This means that if the number of subintervals is doubled, then the error is divided by a factor 2. In general, a method is said to be of order r if

$$\max_{n=0,\dots,N} |y(t_n) - Y_n| \leqslant CN^{-r} \sup_x \left| \frac{d^{r+1}}{dx^{r+1}} y \right|,$$

provided y is $r+1$ times differentiable.
In practice, the validation of the implementation is performed as follow:

- Manufacture a non trivial exact solution $y(t)$, for instance, $y(t) = \cos(\pi t)e^{-3t}$.
- Deduce what should be $f(y,t)$ and y_0. In the example above, $f(y,t) = \frac{d}{dt}y = -(\pi \sin(\pi t) + 3\cos(\pi t))e^{-3t}$ and $y_0 = y(0) = 1$.
- Run your code with this $f(y,t)$ and y_0 and record e_N for different values of N (e.g. $N = 100, 200, 400, 800, ...$).
- Plot e_N versus N in a $log - log$ scale and check you have a straight line (for large N). The slope of such line is the order of the method (why?). To check the performance of your implementation against a method of order r, the matlab code for this last point would be something like:

```
loglog(arrayN,arrayErrors,arrayN,arrayN.^(-r));
legend('Errors','Expected decay');
```

where *arrayN* is an array with all the values of N considered and *arrayErrors* is an array with the corresponding errors e_N.

1. Implement the backward and forward Euler methods and check your implementations following procedure described above.
2. Take the specific case of $T = 1$, $y_0 = 1$, $f(y,t) = -21y$ and run both implementations with $N = 10$. Provide a graph of the exact solution together with the backward and forward Euler approximations.
3. Now perform the same simulations but with $N = 100$. Briefly discuss your observations.

Exercise 2 **25% (MATLAB)**

Ducks swim in the direction of the position they want to reach. The position $(y_1(t), y_2(t))$ of a duck crossing a river is given by

$$\frac{d}{dt}y_1 = \frac{v_D(L-y_1)}{\sqrt{(L-y_1)^2+y_2^2}},$$
$$\frac{d}{dt}y_2 = \frac{-v_D y_2}{\sqrt{(L-y_1)^2+y_2^2}} - v_W,$$

where v_D and v_W are the given duck and water velocities and L is the river width. Refer to the provided figure for a sketch of the situation.

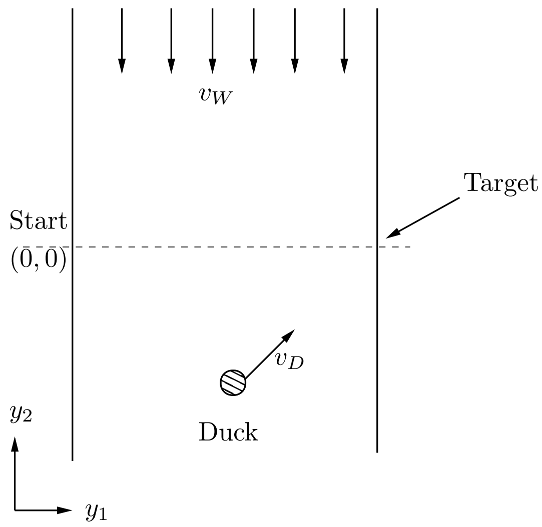

Figure 1: Duck crossing a river

1. Extend your forward Euler implementation for a system of two linear first order ODEs

$$\frac{d}{dt}y_1(t) = f_1(y_1, y_2, t)$$
$$\frac{d}{dt}y_2(t) = f_2(y_1, y_2, t).$$

2. approximate the solution to the “duck” problem with

$$x_0 = 0, \quad x_f = 1.4, \quad y_1(0) = y_2(0) = 0, \quad L = 1, \quad v_D = 1, \qquad v_W = 0.5.$$

   Plot the trajectory $(y_1(x), y_2(x))$, $x \in (0, 1.4)$ of the duck. Run your code again with the same parameter except for $v_W = 1.5$. Discuss your results.

Exercise 3 **25% (MATLAB)**

Let $y_1(t)$ denotes the population of prey and $y_2(t)$ denotes the population of predators at time t. We assume that (i) in absence of the predator, the prey grows at a rate proportional population; (ii) in absence of the prey, the predator dies out; (iii) the number of encounters between predator and prey is proportional to the product of their population. We model this system by the following system of ODEs

$$\begin{aligned}\frac{d}{dt}y_1(t) &= y_1(1-0.5y_2)\\ \frac{d}{dt}y_2(t) &= y_2(-0.75+0.25y_1).\end{aligned}$$

Given the initial population $y_1(0)=5$ and $y_2(0)=1$, chose the number of time-steps (N) appropriately and use your algorithm to solve the system. Plot an approximation of the evolution of $y_1(t)$ and $y_2(t)$ for $t \in (0,25)$. Plot this trajectory in the phase portrait, i.e. the curve joining the points $(y_1(t_i), y_2(t_i))$, $i=1,...,N$. What happen if $y_1(0)=3$ and $y_2(0)=2$? Explain your results.

Exercise 4 **25% (BY HAND)**

Let $\beta > 0$ and consider the ODE

$$\begin{cases} \frac{d}{dt}y(t) = -\beta y(t), & t>0,\\ y(0)=y_0, \end{cases}$$

where y_0 is a given real number. For $h>0$ and $t_n = nh$ for $n=0,1,2,...$, the *Heun* method reads:

- *Initialization:* set $Y_0 = y_0$.
- *Main loop:* For $n \geqslant 0$ define recursively $Y_{n+1} \approx y(t_{n+1})$ as follows

$$p_1 = -\beta Y_n, \qquad p_2 = -\beta(Y_n + hp_1), \qquad Y_{n+1} = Y_n + \frac{h}{2}(p_1+p_2).$$

1. Show that
$$Y_{n+1} = \left(1 - \beta h + \frac{\beta^2 h^2}{2}\right) Y_n$$
2. Determine the largest possible h (depending on β) such that
$$\lim_{n\to\infty} Y_n = 0.$$
3. Show that
$$e^{-\beta h} = 1 - \beta h + \frac{\beta^2 h^2}{2} + O(h^3).$$
4. Let $T>0$ be a final time and for $N \in \mathbb{N}$ define $h = T/N$ and $t_i = ih$, $i=0,...,N$. Show that there exists a constant independent of N such that
$$|y(T) - Y_N| \leqslant Ch^2 = C\frac{T^2}{N^2}.$$