MATH 336: Homework #3

Due: Tuesday, October 4, 2016

Solve the below problems concerning differential equations. A (possibly improper) subset of them will be graded. All calculations should be done analytically, unless marked with an (M). (M) problems require the use of MATLAB. ES denotes the online lecture notes.

1. (20 points) (ES, p.127-128, #3) Problem 3 in the ODE1 section in the notes (end of chapter 2).

2. (20 points) (M) Using MATLAB, solve the logistic equation

$$\frac{dN}{dt} = 2N(1 - \frac{N}{3})$$

numerically, for the 3 different initial conditions $N(0) = 1, 2.5, 4$ on the time interval $t \in [0, 10]$. Plot all 3 solutions on the same set of axes, and use a legend *in MATLAB* to label each one. Note that you are now solving ODEs, and not difference equations, so your fundamental command will be **ode45**. Please read up on **ode45**, and/or come talk to me about it if you have questions. The basic skeleton for the code is below:

```
tI=0;
tF=10;
IC1=1;
[T1,N1]=ode45(@rhsHw3No2,[tI tF],IC1);

%Add two other initial conditions and plotting below here
```

This will be the main source code that runs your program, saved as say *hw3No2.m*. However, you must create an additional **function** *m-file*, where the right-hand side of the equation is defined. This function should be called *rhsHw3No2.m* (note the same name in the code above), and must be saved in the same directory as *hw3No2.m*. Inside this function, define the right-hand side of the logistic equation, **exactly** as below:

```
function dNdt=rhsHw3No2(t,N)
    r=2;
    K=3;
```

```
    dNdt=r*N.*(1-N/K);
end
```

Note the keyword *function*, which distinguishes this from a script (like *hw3No2.m*). Here I am giving you the vector field (ODE part) precisely, but in the future you will have to define it yourself. When you want to actually solve the equation, you run *hw3No2.m* and **NOT** *rhsHw3No2.m*. I also did not include the plotting and solving for the other initial conditions in *hw3No2.m*, and I expect you to add these features to the code.

3. (10 points) Suppose a population of cells grows exponentially (in **continuous time**) with a rate constant k. Assume that after τ minutes, the population doubles. Find an expression for k as a function of τ, that is, $k = k(\tau)$.

4. (20 points) (partly taken from ES, p.128-129, #4) Consider the continuous Ricker model

$$\frac{dN}{dt} = rNe^{-\beta N}$$

where $\beta > 0$.

   (a) Find constants $\bar{t}, \bar{N}$ such that under the non-dimensionalization $N_*(t_*) := \frac{1}{\bar{N}}N(\bar{t}t_*)$, $N_*(t_*)$ satisfies the differential equation

$$\frac{dN_*}{dt_*} = N_* e^{-N_*}. \tag{1}$$

   (b) Find the values of N_* that are inflection points of solutions of the equation (1). What values of the original Ricker model do these correspond to? State the concavity on the different population domains ($N_* > 0$ only of course).

5. (15 points) (ES, p.130, #1, parts (a) and (b)) Problem 1 in the ODE2 section in the notes (end of chapter 2), **parts (a) and (b) only**.

6. (15 points)

   (a) Consider a bacterial population whose growth rate is

$$\frac{dN(t)}{dt} = K(t)N(t) \tag{2}$$

Here $K(t)$ is the *non-constant* growth rate. Show that $N(t)$ satisfies

$$N(t) = N_0 \exp\left(\int_0^t K(s)\,\mathrm{d}x\right),$$

where $N_0 = N(0)$.

(b) Show that if $K(t)$ decreases exponentially (that is, $K(t) = e^{-\alpha t}$ for some $\alpha > 0$), then $N(t)$ solving (2) must remain bounded (i.e. there exists a constant M such that $N(t) \leq M$ for all times t). Such growth is termed *Gompertzian*, and is often the growth rate utilized to model solid tumor growth. In fact, despite the model's simplicity, it has found a great deal of success in fitting clinical data. See "Tumor growth and chemotherapy: mathematical methods, computer simulations, and experimental foundations" in *Math. Biosc.* (Aroesty *et al.*) for a specific example.