Mathematics for the Life Sciences (MATH1134) - Tutorial Sheet 3

`This tutorial complements the material covered in Lecture 3.`

USA POPULATION DATA 1790 - 2010

YEAR	POPULATION	YEAR	POPULATION	YEAR	POPULATION
1790	3,929,214	1870	38,558,371	1950	151,325,798
1800	5,308,483	1880	50,189,209	1960	179,323,175
1810	7,239,881	1890	62,979,766	1970	203,211,926
1820	9,638,453	1900	76,212,168	1980	226,545,805
1830	12,866,020	1910	92,228,496	1990	248,709,873
1840	17,069,453	1920	106,021,537	2000	281,421,906
1850	23,191,876	1930	123,202,624	2010	308,745,538
1860	31,443,321	1940	132,164,569	2020	???

1. Consider the USA population data table above.

   (a) Which 10 year period between 1790 and 1840 has the highest per capita population growth rate and what is that rate?

   (b) Which 10 year period between 1790 and 1840 has the highest population growth and what is that growth in population?

   (c) Use the per capita population growth rate, r, from the years 1790 to 1800 and assume exponential population growth (so $N(t) = N_0 e^{r(t-t_0)}$) to predict what the population would be in 1810 *(why did I not ask you to predict the population in 1800???)*, 1820, 1830, and 1840. Given the actual population from the table, find the relative errors. NOTE for accurate results, calculate r to at least 16 decimal places - meaning, use *format long* in Matlab.

   (d) Show that if there is exponential growth of a population governed by $N(t) = N_0 e^{r(t-t_0)}$, then the *doubling time* of the population is always $\frac{\ln 2}{r}$.
   ASIDE 1: if $r < 0$ and the equation $N(t) = N_0 e^{r(t-t_0)}$ can represent radioactive decay, the equivalent notion of "doubling time" is called "half-life" and is useful for carbon dating.
   ASIDE 2: This is not too hard to do if you assume $t_0 = 0$ and you just want to see how long it takes for the population to become $2N_0$. You have to think more carefully about what the term "doubling time" means if you consider the more general case where t_0 may not be 0 and you have to show the result is always true, not just when going from a population of N_0 to $2N_0$.

(a) Since $N(t) = N_0 e^{r(t-t_0)}$, to find r for any of the 10 year periods between population listings, we would take the log of both sides of that equation and solve for r:

$$r = \frac{\ln\left(N(t)/N_0\right)}{t - t_0} = \frac{\ln\left(N(t)/N_0\right)}{10}$$

since the r is only for each 10 year period. This is easy to compute - for example, at $t = 1800$ the formula is

$$r = \frac{\ln(5308483/3929214)}{10} = 0.030087$$

If you wanted to automate the calculations of these r values in Matlab, then you could store the 6 population totals for the years 1790, 1800, 1810, 1820, 1830, and 1840 in a vector, say POP, and then write a for loop for $k = 1:5$ which printed out $\log(POP(k+1)/POP(k))/10$.

Anyway, the highest 10-yearly net per capita population growth rate in that period is 0.031030 between the years 1800 and 1810.

(b) 4203433 **from 1830 to 1840.**

(c) **Use $r = 0.0300866701176633$. Of course you would get 5308483 - so no error - if you used this r to calculate the population in 1800. Here are the other calculated populations and the relative errors:**

Year (t)	N(t)	Relative Error
1810	7, 171, 916	0.00938762013558719
1820	9, 689, 468	0.00529284873639788
1830	13, 090, 754	0.0174672825877710
1840	17, 685, 992	0.0361194329793176

(d) **Assuming we know the population at some time T_1, $N(T_1) = N_0 e^{r(T_1 - t_0)}$. We then want to find the time T_2 such that the population has doubled, *so that* $T_2 - T_1$ *is the doubling time*: i.e, we want T_2 such that**

$$N(T_2) = 2N(T_1) \quad \Rightarrow \quad N_0 e^{r(T_2 - t_0)} = 2N_0 e^{r(T_1 - t_0)}$$

$$\Rightarrow \quad e^{r(T_2 - T_1)} = 2 \quad \Rightarrow \quad T_2 - T_1 = \frac{\ln 2}{r} = \textit{ the doubling time.}$$

---

2. Solve the **Logistic differential equation**,

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right) \quad \text{with initial condition} \quad P(0) = P_0,$$

where r (the relative growth rate) and K (the carrying capacity) are constants.

Rewrite as

$$\frac{1}{P\left(1 - \frac{P}{K}\right)}\, dP = r\, dt. \tag{1}$$

- **Before, integrating, we simplify the left hand side integrand to $\frac{K}{P(K-P)}$. Now we use partial fraction decomposition:**
  - **We seek constants a and b so that $\frac{K}{P(K-P)} = \frac{a}{P} + \frac{b}{K-P} \Rightarrow K = a(K-P) + bP$.**
  - **Find a and b by matching like terms or picking appropriate values for P.**
  - **For example, $P = 0 \Rightarrow K = aK \Rightarrow \boldsymbol{a = 1}$.**
  - **And picking $P = K \Rightarrow K = bK \Rightarrow \boldsymbol{b = 1}$.**

- **So Equation 1 becomes** $\left(\frac{1}{P} + \frac{1}{K-P}\right) dP = r\, dt.$
- **Integrating both sides gives**

$$\ln|P| - \ln|K-P| = rt + C \Rightarrow \ln\left|\frac{P}{K-P}\right| = rt + C \Rightarrow$$

$$\ln\left|\frac{K-P}{P}\right| = -rt - C \Rightarrow \left|\frac{K-P}{P}\right| = e^{-rt-C} = e^{-C}e^{-rt} \Rightarrow$$

$$\frac{K-P}{P} = Ae^{-rt}$$

where $A = \pm e^{-C}$ **is an abritrary constant.**
Solving for P **we get**

$$\frac{K}{P} - 1 \quad = \quad Ae^{-rt} \quad \Rightarrow \quad \frac{P}{K} \quad = \quad \frac{1}{1+Ae^{-rt}} \quad \Rightarrow$$

$$\boldsymbol{P(t) = \frac{K}{1+Ae^{-rt}}}$$

- **So if we add the initial condition** $\boldsymbol{P(0) = P_0}$**, the** $\boldsymbol{A = \frac{K}{P_0} - 1}$ **so that**

$$\boldsymbol{P(t) = \frac{K}{1+\left(\frac{K}{P_0}-1\right)e^{-rt}}}$$

---

3. For this problem, use the USA population data at the start of this tutorial. Use the intrinsic growth rate from **Example 2** of the **Lecture 3** notes, $r = 0.0315482567314$ and either the Heun's method or fourth order Runge-Kutta method programs to explore the behaviour of numerical solutions to the Logistic differential equation starting in 1790, 1800, 1810, 1820, 1830, and 1840. In each case, use the given initial population at those times and an appropriate Δt to see how well the approximation method matches the actual yearly population data for each 10 year period until 2010.

---

4. Repeat **Example 2** in the **Lecture 2** notes but basing the parameters on another set of years: Use the data from the USA population table in the years 1900, 1940, and 1980 to determine a **Logistic** growth population model $N(t)$ which is exact for those three years. Use this model to predict the populations in 1960, 2000, and 2010 and comment on the accuracy. You can use a numerical approach with Heun's or the fourth order Runge-Kutta method if you prefer (provided, of course, you have found the intrinsic growth rate r and the carrying capacity K).

---

5. *This is essentially taken from* `Essential Mathematical Biology` *by Nicholas Britton:*

   Here is a quick introduction to the *ecosystems* or *resource-based* approach to modelling and its application to an alternative derivation of the Logistic differential equation.

Assume the per capita growth rate of a population depends on some **resource**. Furthermore, assume this resource exists in two states: **free** *(i.e., available for use by members of the population)* or **bound** *(i.e., already in use)*. If we let R be the density of the free resource then the rate of change of the population

$$\frac{dN}{dt} = G(R)N$$

for some function G of the free resource density.

Let the resource be *abiotic* (non-biological), and therefore cannot be born or die. (The archetypal example in the ecosystems approach is a mineral resource, but another possibility is something like nest sites.) Next assume the *conservation law* that *the total amount of resource, free and bound, is a constant* $C > 0$, and let the amount of bound resources depend on the population. Thus

$$R = C - H(N).$$

In modelling $G(R)$ and $H(N)$, note that they both must increase with their arguments. And since logically if there is no density of free resource the growth rate must be negative, then

$$G(0) < 0.$$

So a simple linear form for G would be

$$G(R) = \alpha R - \beta$$

with $\alpha > 0$ and $\beta > 0$. Similarly, if the population is 0 there can be no bound resource so $H(0) = 0$ and so a simple linear form for H is

$$H(N) = \gamma N$$

with $\gamma > 0$.

(a) Show that this ecosystems modelling approach still leads to the Logistic differential equation.

(b) Give expressions for the Malthusian parameter r and the carrying capacity K in terms of the parameters in the ecosystems modelling approach.

(c) What happens if the total amount of resource C is insufficient?

(d) Give an advantage of the ecosystems approach and the empirical approach used in the **Lecture 3** notes to modelling limited growth.

(a) A straightforward substitution shows this.

(b) $r = \alpha C$ **and** $K = \frac{\alpha C - \beta}{\alpha\gamma}$**.**

(c) If $r < 0$ **or equivalently** $C < \frac{\beta}{\alpha}$ **then** $N(t) \to 0$ **as** $t \to \infty$**.**

(d) The empirical approach from the Lecture 3 notes is probably more intutitive and simpler than the ecosystems approach. The second approach lends some useful insight and allows us to test some theories fairly easily such as the effect of a reduction in C **on both the growth rate and carrying capacity.**

---

6. Consider the differential equation which models *metapopulations*,

$$\frac{dp}{dt} = cp(t)(1 - p(t)) - ep(t),$$

where $p(t) =$ the fraction of sites occupied at time t, e is a local extinction rate, and c is a colonisation rate. Show that this is a Logistic differential equation with intrinsic growth rate

$$r = c - e$$

and carrying capacity

$$K = 1 - \frac{e}{c}.$$

This is a fairly routine calculation so I will leave it to you.

---

7. *(Essentially a slightly modified version of Exercise 4.1 of* ***`A Primer on Mathematical Models in Biology`*** *by Lee Segel and Leah Edelstein-Keshet)*

Consider the Malthusian population growth initial value problem

$$\frac{dN}{dt} = rN, \quad N(0) = N_0, \quad r > 0.$$

(a) Let $y(t) = \frac{N(t)}{N_0}$ and rewrite the differential equation and initial conditions in terms of this dimensionless $y(t)$.

(b) What are the units of r?

(c) What is the *doubling time* of the population?

(d) Define a dimensionless time τ such that this initial value problem is transformed into

$$\frac{dy}{d\tau} = y, \quad y(0) = 1.$$

Quite similar to the section on Nondimensionalisation of the Logistic DE in the Lecture 3 notes.

(a) If $y(t) = \frac{1}{N_0}N(t)$ then $\frac{dy}{dt} = \frac{1}{N_0}\frac{dN}{dt} \Rightarrow \frac{dN}{dt} = N_0\frac{dy}{dt}$ so that the original Malthusian ODE $\frac{dN}{dt} = rN$ becomes

$$N_0\frac{dy}{dt} = rN \quad \Rightarrow \quad \frac{dy}{dt} = r\frac{N}{N_0} \quad OR \quad \frac{dy}{dt} = ry.$$

(b)

(c)

(d) $\tau = tr$ is dimensionless. By the chain rule,

$$\frac{dy}{dt} = \left(\frac{dt}{d\tau}\right)\left(\frac{d\tau}{dt}\right) = \frac{dy}{d\tau}r$$

So the differential equation becomes $r\frac{dy}{d\tau} = ry$ or

$$\frac{dy}{d\tau} = y.$$

The initial condition $N(0) = N_0$ **becomes** $y(0) = \dfrac{N(0)}{N_0} = \dfrac{N_0}{N_0} = 1$ **so that**

$$y(0) = 1.$$

---

8. In addition to the approximate root finding methods and approximate IVP solvers mentioned in **Lecture 2** and **Tutorial 2**, Matlab has some corresponding in-built functions which some of you might find easier to use:

   (a) `fzero` - used for approximating roots of functions of one variable. For example, on page 25 of the **Lecture 2** notes, in the equation involving r one could bring everything over to one side of the equation then define a function M file or function handle *(carefully!)* for that expression and then use `fzero` to find the solution. Type

   `help fzero`

   in the Matlab command window and see

   http://uk.mathworks.com/help/matlab/ref/fzero.html

   for more.

   (b) `roots` - used for approximating roots of **polynomials**. Type

   `help fzero`

   in the Matlab command window and see

   http://uk.mathworks.com/help/matlab/ref/roots.html

   for more.

   For example, to find the roots of

   $x^2 - 10x + 25$, type `roots([1,-10,25])`. The answer is 5 (repeated);

   $x^2 - 1$, type `roots([1,0,-1])`. The answer is $-1, 1$;

   $x^3 - 8x^2 + 37x - 50$, type `roots([1,-8,37,-50])`. The answer is $3 + 4i$, $3 - 4i$, 2.

   (c) `fsolve` - used for approximating roots of functions of more than 1 variable, *i.e.,* solving $\vec{F}(\vec{x}) = \vec{0}$. For example, to solve the system given in *Appendix A* of **Lecture 3**, you could first define the function handle for the vector field $\vec{F}$ as follows:

   `F = @(x)` $[3 * x(1) - cos(x(2) * x(3)) - 1/2; x(1)^2 - 81 * (x(2) + 0.1)^2 + sin(x(3)) + 1.06; exp(-x(1) * x(2)) + 20 * x(3) + (10 * pi - 3)/3]$;

   Then type

   `fsolve(F, [0.1; 0.1; -0.1])`

   to get a vector of approximate solutions. Type

   `help fsolve`

   in the Matlab command window and see

   http://uk.mathworks.com/help/optim/ug/fsolve.html

   for more.

   (d) `ode45` and several other related ODE solvers - used to solve initial value problems. For example, to do EXAMPLE 10 from **Lecture 2** using $h = 0.2$ with an in-built fourth order Runge-Kutta solver, we could first define the function handle for the right hand side of the ODE:

   `f = @(t,y) -y/(2*t)`

then type

```
[T,Y] = ode45(f, 1:0.2:2, 12)
```

For more on this you can type `help ode45` and also visit the page

http://uk.mathworks.com/help/matlab/ref/ode45.html

Importantly, we will be able to use these ODE solver commands with *systems of ODEs* also - see **Lecture 4** and later.
