

Math 2280-1

Applying the logistic model to U.S. populations
January 20, 2006

The logistic equation for population change which we have just been studying is:

$$\frac{dP}{dt} = kP(M - P).$$

Following the text and its equation numbering at page 88, we can write the same differential equation as

$$\frac{dP}{dt} = aP + bP^2 \quad (1)$$

with $a = kM$ and $b = -k$. In our model the parameters k, M, a, b are related to assumptions about birth and death rates. Suppose you have a real population and want to pick parameters a and b to make a good model. One way would be to try to estimate fertility and morbidity rates based on birth and death data, but that could get quite complicated. For example, if you want to develop an accurate model of world population growth based on this sort of analysis you would probably need to collect data from different regions of our planet and develop different parameters for different societies, solve the problem in each part of the globe and then add your results together. A more simple-minded approach is to see if existing population data is consistent with a logistic model, for appropriate choices of a and b . The book explains a good way to do this on pages 88-90, in the context of modeling U.S. populations over the past two centuries.

If you divide the logistic DE, equation (1) above, by P , you get

$$\left[\frac{1}{P} \right] \left[\frac{dP}{dt} \right] = a + bP \quad (2)$$

If you have multi-year population data you can get good estimates for the left side of (2) by using

difference quotients to estimate $\frac{dP}{dt}$. Dividing by P gives an estimate for $\left[\frac{1}{P} \right] \left[\frac{dP}{dt} \right]$. Carrying this

computation out for several different times yields a collection of points approximating $\left[P, \left[\frac{1}{P} \right] \left[\frac{dP}{dt} \right] \right]$.

If these points seem to lie approximately along a line, then (2) will be a good model for the population problem, and you can estimate the parameters " a " and " b " by getting the vertical axis-intercept and slope of the line which best fits the point data, respectively.

For example, looking at the USA data in Figure 2.1.8 on page 90, we can estimate $\frac{dP}{dt}$ in 1800 in a "centered" way by taking the difference $(P(1810) - P(1790))/20$ (in units of people/year). Centered differences as in (3) page 86, generally give more accurate estimates for the derivative than the one-sided differences used in the limit definition. This is shown geometrically in Figure 2.1.7, page 89. Finally we would divide our centered estimate for dP/dt in 1800 by $P(1800)$ to get an estimate for

$\left[\frac{1}{P} \right] \left[\frac{dP}{dt} \right]$ in 1800, for the USA population:

[

```

[ > restart:
[ > P1:=5.308;
    #population in 1800, in millions.
    #Remember you can do multiline commands
    #by holding down the shift key when you hit return
    #or enter.
P1prime:=(7.240-3.929)/20;
    #estimate for dP/dt in 1800, in millions of
    #people per year, see top right entry in
    #Figure 2.1.8 page 90

                P1 := 5.308

                P1prime := 0.1655500000
[ > P1primeoverP1:=P1prime/P1;
    #estimate for (1/P)*(dP/dt) in 1800;

                P1primeoverP1 := 0.03118877167
[ > point1:=[P1,P1primeoverP1];
    #the left-most point on the graph of Figure 2.1.9

                point1 := [5.308, 0.03118877167]
[ >

```

We can automate this process. We are still using the table 2.1.8 on page 90. You should verify how these numbers were extracted from the tables.

```

[ > with(linalg):
Warning, the protected names norm and trace have been redefined and unprotected
[ > pops:=matrix(21,2,[[1790,3.9],[1800,5.3],[1810,7.2],
    [1820,9.6],[1830,12.9],[1840,17.1],
    [1850,23.2],[1860,31.4],[1870,38.6],
    [1880,50.2],[1890,63.0],[1900,76.2],
    [1910,92.2],[1920,106.0],[1930,123.2],
    [1940,132.2],[1950,151.3],[1960,179.3],
    [1970,203.3],[1980,225.6],[1990,248.7]]);
    #matrix of populations
[ >

```

1790	3.9
1800	5.3
1810	7.2
1820	9.6
1830	12.9
1840	17.1
1850	23.2
1860	31.4
1870	38.6
1880	50.2
1890	63.0
1900	76.2
1910	92.2
1920	106.0
1930	123.2
1940	132.2
1950	151.3
1960	179.3
1970	203.3
1980	225.6
1990	248.7

pops :=

```
> for i from 1 to 11 do
  lspoints[i]:=[pops[i+1,2],
    (pops[i+2,2]-pops[i,2])/(20*pops[i+1,2])];
od;
>
```

$lspoints_1 := [5.3, 0.03113207547]$

$lspoints_2 := [7.2, 0.029861111111]$

$lspoints_3 := [9.6, 0.02968750000]$

```

lspoints4 := [12.9, 0.02906976744]
lspoints5 := [17.1, 0.03011695906]
lspoints6 := [23.2, 0.03081896552]
lspoints7 := [31.4, 0.02452229300]
lspoints8 := [38.6, 0.02435233160]
lspoints9 := [50.2, 0.02430278884]
lspoints10 := [63.0, 0.02063492064]
lspoints11 := [76.2, 0.01916010498]

```

We will follow investigation A and find the least squares line fit to this data, in order to extract values for "a" and "b" in equation (2). We remember how to do this from the Math 2270 chapter on orthogonality, which included the method of least squares as a special subtopic.... You could look at class notes from last semester,

<http://www.math.utah.edu/~korevaar/2270fall05/oct21.pdf>

```

> A:=matrix(11,2);
  B:=vector(11);

                                A := array(1 .. 11, 1 .. 2, [ ])
                                B := array(1 .. 11, [ ])

> for i from 1 to 11 do
  A[i,1]:=lspoints[i][1]:
  A[i,2]:=1:
  B[i]:=lspoints[i][2]:
od:

> evalm(A);evalm(B);
  #check work

```

5.3	1
7.2	1
9.6	1
12.9	1
17.1	1
23.2	1
31.4	1
38.6	1
50.2	1
63.0	1
76.2	1

```
[0.03113207547, 0.02986111111, 0.02968750000, 0.02906976744, 0.03011695906,
  0.03081896552, 0.02452229300, 0.02435233160, 0.02430278884, 0.02063492064,
  0.01916010498]
```

```
> linsolve(transpose(A)* (A) , transpose(A)*B);
#least squares solution
```

```
[-0.0001694136800, 0.03185105239]
```

```
> a:=.3185105239e-1;
#intercept
b:=-.1694136800e-3;
#slope
```

```
a := 0.03185105239
```

```
b := -0.0001694136800
```

Now we can create figure 2.1.9:

```
[ > with(plots): #load the plotting package
```

```
> pict1:=pointplot({seq(
  lspoints[i],i=1..11)}):
#a plot of the points,
#with output suppressed. Make sure
#to end this command with a colon! If you use a
#semicolon you get a huge mess when you have
#a lot of points
```

```
> line:=plot(a+b*P,P=0..100, color=black): #same warning here
#about using a colon vs semicolon
```

```
> display({pict1,line}, title="Figure 2.1.9");
```

```
#now use a semicolon, and
#get a picture containing the line and the
#three points we computed.
```

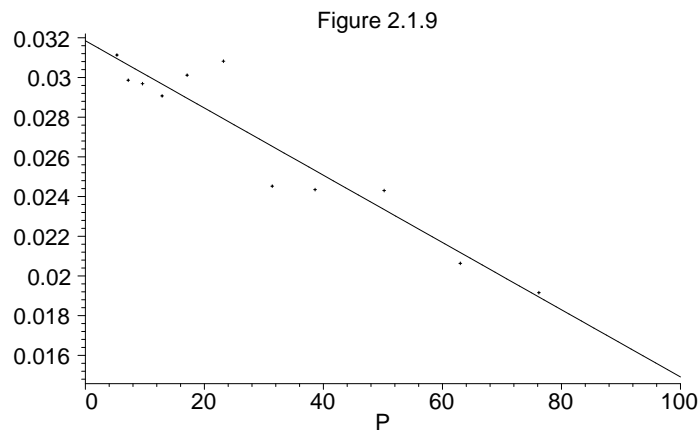


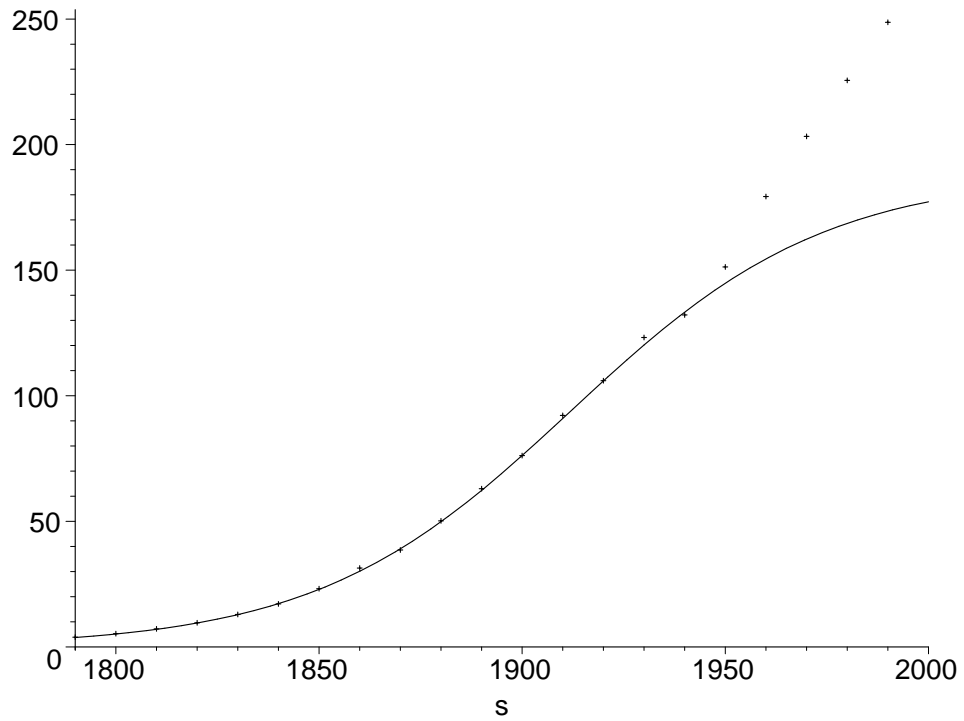
Figure 2.1.9 shows that the logistic model is a pretty good one for the U.S. population in the 1800's. Let's use the "a" and "b" which we found with the least squares fit, use the population in 1900 for our initial condition, and see how the logistic model works when we try to extend it to the 1900's:

```
> with(DEtools): #load the DE package
Warning, the name adjoint has been redefined
> deqtn1:=diff(x(t),t)=a*x(t) + b*x(t)^2;
#logistic eqtn with our parameters
deqtn1 :=  $\frac{d}{dt} x(t) = 0.03185105239 x(t) - 0.0001694136800 x(t)^2$ 
> P:=dsolve({deqtn1,x(0)=76.21},x(t));
#take 1900 as t=0 and solve the initial value
#problem
P := x(t) =  $\frac{3185105239}{16941368 + \frac{189400358372}{7621} e^{-\frac{3185105239 t}{1000000000000}}}$ 
> f:=s->evalf(subs(t=s,rhs(P))); #extract the right-hand side
#from the above expression to make your solution function
f:=s->evalf(subs(t=s,rhs(P)))
> f(s); #check that those weird subs and rhs commands really work
 $\frac{0.3185105239 10^{10}}{0.16941368 10^8 + 0.2485242860 10^8 e^{(-0.03185105239 s)}}$ 
```

We can see how the model works by plotting actual populations against predicted ones, for the 1900's.

```
> actual:=pointplot({seq([pops[i,1],pops[i,2]],i=1..21)}):
```

```
model:=plot(f(s-1900),s=1790..2000,color=black):  
display({actual,model});
```



Can you think of possible reasons why the model began to fail so badly around 1950? There are several.