Parameter Identification and Spread of Information

A marketing firm wants to know how fast a new product will be learned about. Information can be obtained either via mass media or via personal communication. The mass communication refers to newspapers, journals, radio and television. The personal communication refers to discussions with other people. The personal communications can have large changes in the number of informed people once a significant proportion of the population knows about the new product. In effect, every informed person can become a "broadcaster" of the information.

The marketing firm would like to know how to best inform a certain segment of the population about a given piece of information. On any day what number of people will be informed about the new product?

We may model the two modes of information dissemination. Moreover, we may consider a discrete model for small populations, or a continuum model for much larger populations. The four models for the spread of information are

spread of information via mass communication and discrete,

spread of information via mass communication and continuous,

spread of information via personal communication and discrete and

spread of information via personal communication and continuous.

Spread of Information via Mass Communication and Discrete.

This model is very similar to the cooling model. Let u(i) be the total number who are informed by day i. Let dt = 1 day. And suppose the total population is 20000, and so, the uninformed group is equal to 20000 - u(i). If either dt or 20000 - u(i) are large, then we expect the change in the number who are informed to be large. One way to model this is to claim the change in the number who are informed is directly proportional to the product of the change in time, dt, and the number who do not know, 20000 - u(i),

$$u(i+1) - u(i) = dt*k*(20000 - u(i))$$

where k reflects the effectiveness of the mass media. This is the Euler model of the following continuous model.

Spread of Information via Mass Communication and Continuous.

The continuous model can be used for very large populations where u(t) is viewed as a continuous function of time, t, and u(t) is the percentage (or number) of people who know at any time t. In the previous model we considered the discrete model. Here we have a similar formula for the change in the number who know

$$u(t + dt) - u(t) \approx dt*k*(20000 - u(t)).$$

Now the number who know, u(t), will change a little as t moves from t to t+dt, and this is the reason we do not have an equality sign for the above change in the number who know. If we let dt go to zero, we just get zero on both sides! However, if we divide by dt and let dt go to zero, we get the differential equation

$$u' = k\,(20000 - u).$$

This differential equation with known k and given initial condition $u(0) = u_0$ is the continuous model.

Spread of Information via Personal Communication and Discrete.

If information is being learned from those who already know, then the constant, k, in the above two models must be changed to reflect the number of the informed people. One reasonable choice might be k = cu where c is another constant. Then the change in the number of people who know would be directly proportional to the product of the change in time, dt, the number who know, y, and the number who do not know

$$u(i+1) - u(i) = dt*c*u(i)*(20000 - u(i)).$$

This is the Euler model of the following continuous model.

Spread of Information via Personal Communication and Continuous.

A similar modification for the continuous model gives

$$u(t + dt) - u(t) \approx dt*c*u(t)*(20000 - u(t)).$$

Now divide by dt and let it go to zero to get the differential equation

$$u' = cu\,(20000 - u).$$

The differential equation with c and an initial condition $u(0) = u_0$ make up the continuous model. A differential equation of the form

$$u' = cu(M - u)$$

is called a **logistic differential equation,** and it and its variations are an important class of models.

In this lesson we will use the Matlab command ode45 to solve our differential equation. This command is a robust implementation of a variable step size method and the fourth and fifth order Runge-Kutta method. Matlab has a number of solvers for initial value problems, and the reader can learn about these by either executing the Matlab command odedemo or using helpdesk and looking under differential equations.

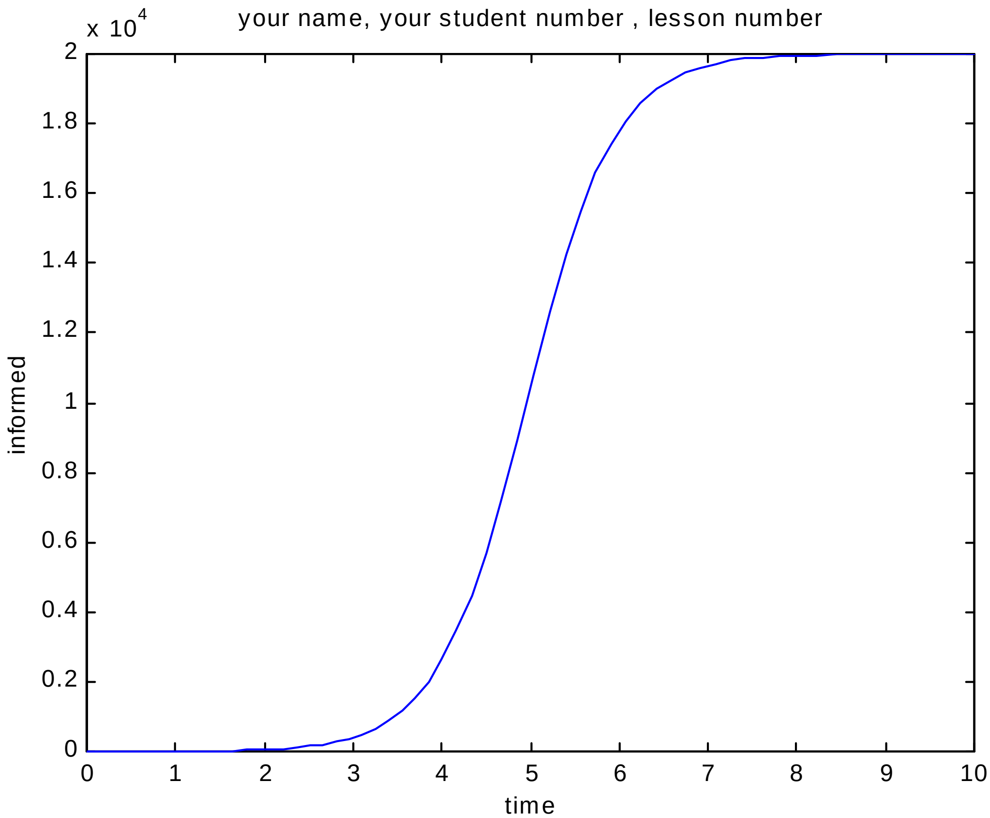
An easy way to execute the ode45 command is to create two m-files: **ypinfo.m** and **info.m**. The ypinfo.m file contains the right side of the differential equation for the personal communication model. In the second line of the ypinfo.m file we have the logistic model with harvesting where c = .0001 and M = 20000; here the symbol y(1) represents the number who are informed out of population equal to 20000. The c was computed as given in the next section from two observations: initially one person knew and after one day 200 people knew. The output is given in the graph below.

ypinfo.m

```
function ypinfo = ypinfo(t,y)
  ypinfo(1) = (.0001)*y(1)*(20000 -y(1));
```

info.m

```
%your name, your student number, lesson number
clear;
yo = [1];
to = 0;
tf = 10;
[t y] = ode45('ypinfo',[to tf],yo);
plot(t,y)
title('your name, your student number, lesson number')
xlabel( 'time')
ylabel( 'informed')
```



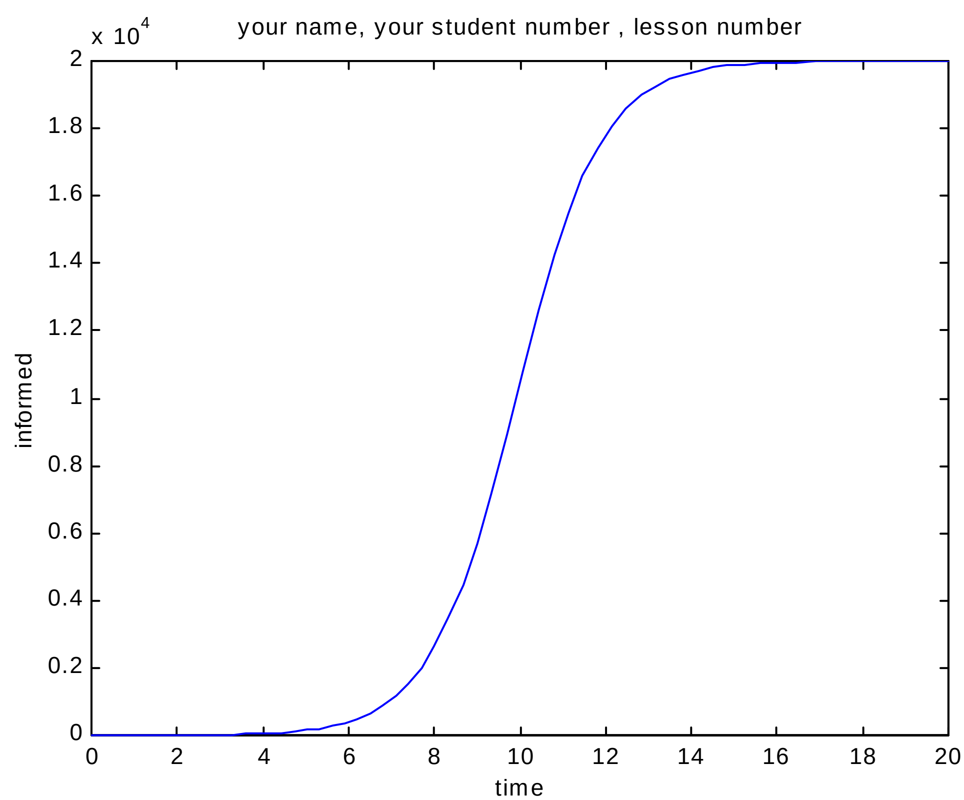
Suppose u(0) = 1 and u(1) = 200 are the number who know initially and after one day. Then for the mass communication model we can find k by using u' = k(20000 - u) and the improved Euler method

200 - 1 = (1/2)* [k* (20000 - 1) + k* (20000 - 200))] and then k = .01.

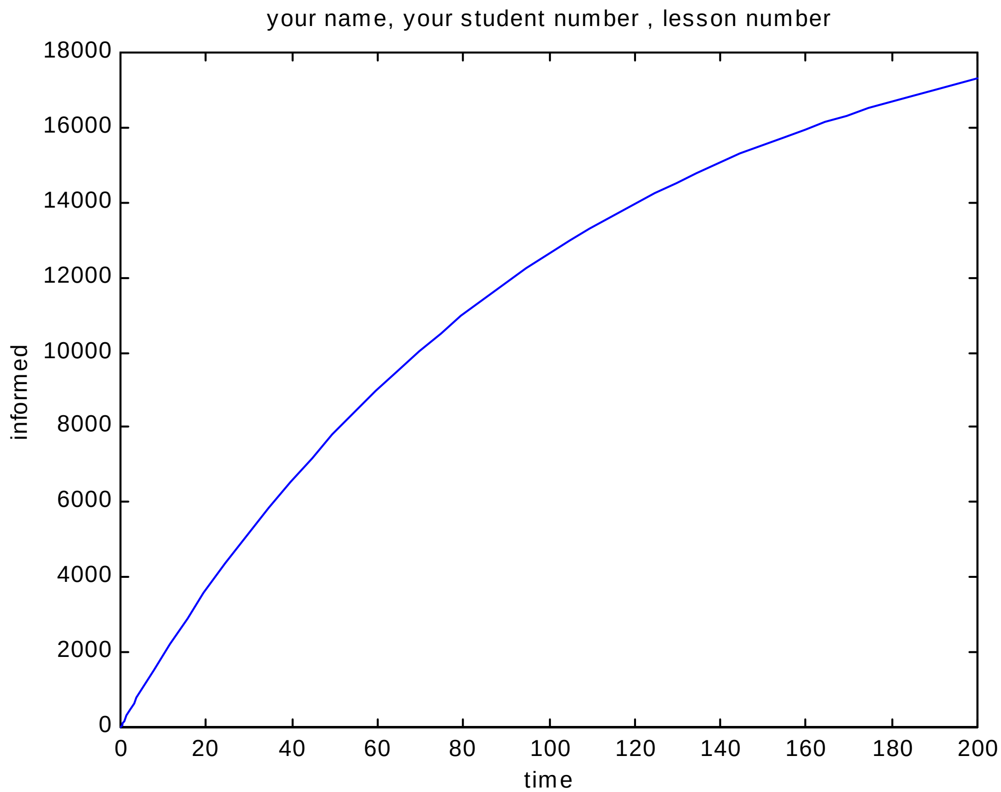
For the personal communication model we find the c by using the logistic differential equation u' = cu(20000 - u) and the improved Euler method

200 - 1 = (1/2)*[c*1*(20000 - 1) + c*200*(20000 - 200)] and then c = .0001.

In the above graph, for the personal communication model, most people were informed after seven days. If the constant c , the "gossip" constant, is smaller, then it would take longer for the entire populations to learn the information. The second graph is for c = 0.00005 and not 0.0001., and the final time is changed from 10 to 20. Here it takes about 14 days for most of the population to become informed.



The next calculation is for the mass media communication model where $u' = .01(20000 - u)$. Note it takes much longer for most of the population to learn the information. In the graph below not even 90% were informed after 200 days. The $k = .01$ is the "advertising" constant, and if it is increased, then most of the population will learn the information sooner.



Many advertisements are very clever or cute. This is to encourage people to "talk" about the advertisement so that **both** the mass media and personal communication are used to spread information. In this case the combined continuous model is

$u' = cu(M - u) + k(M - u)$ where the constants c, M, k and $u(0) = u_0$ are to be found from additional observations at given times. The following Matlab codes, **info_parid.m** and **info_fmin.m**, approximate the first three parameters by using least squares for quadratic polynomials and fminsearch(), respectively. In both cases we use some of the data collected at 17 times steps in the following table.

Times	**Informed People**
1	01.0
2	02.9
3	06.2
4	10.5
5	17.3
6	26.6
7	38.1
8	50.9
9	63.5
10	74.6
11	82.9
12	89.0
13	93.1
14	95.7
15	97.4
16	98.4
17	99.6

info_parid.m

```
% This code uses least squares to identify the three parameters in the
% spread of information model:
%   u_t = c(M - u)u + k(M - u) where
%               u = u(t) in the number (or percent) who know a fact,
%               M = size of the population (or 100 percent),
%               c = "gossip" coefficient and
%               k = "advertising" coefficient.
% The data is given in the vector ud and is adjusted by a random
variable.
% The data is used in the finite difference approximation of the above:
%   (u_i+1 - u_i-1)/(2 dt) = c(M - u_i)u_i + k(M - u_i)
%                          = (-c)(u_i)^2 + (cM - k)(u_i)^1 + kM.
% Least squares is used to compute the quadratic polynomial
coefficients.
% The first 8 data points are used.
%
%function [t y] = infoid
%    global oldc oldM oldk
%    yo = [1];
%    to = 0;
%    tf = 20;
%    [t y] = ode45('ypinfoid',[to tf],yo);
%
%function ypinfoid = ypinfoid(t,y)
%  global oldc oldM oldk
%  ypinfoid(1) = oldc*y(1)*(oldM-y(1)) + oldk*(oldM-y(1));
%
clear; clf(figure(1))
global oldc oldM oldk
oldc = 0.005; oldM = 100; oldk = 0.01;
ud = [ 1 2.9 6.2 10.5 17.3 26.6 38.1 50.9 ...
       63.5 74.6 82.9 89.0 93.1 95.7 97.4 98.4 99.6 ];
ud = ud + .1*rand(1,17) - .05;
td = 0:1:16;
for i = 2:1:15
    d(i) = (ud(i+1) - ud(i-1))/(td(i+1) - td(i-1));
    A(i,1) = ud(i)^2; A(i,2) = ud(i); A(i,3) = 1.0;
end
%
x = A(2:8,:)\d(2:8)';
c = -x(1);
M = (-x(2) - (x(2)^2 - 4*x(1)*x(3))^0.5)/(2*x(1));
k = x(3)/M;
%
[oldc oldM oldk]
[c M k]
plot(td(1:1:8),ud(1:1:8),'*',td,ud,'x')
oldc = c; oldM = M; oldk = k;
[t y] = infoid;
hold on
plot(t, y, 'r')
```

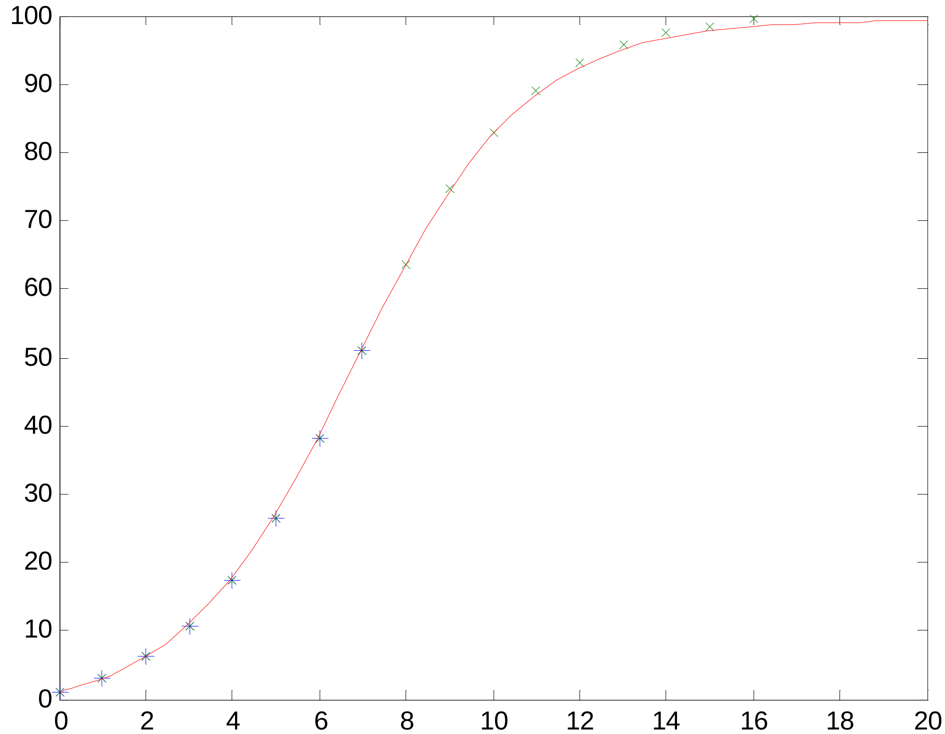
```
>> info_parid

ans =

    0.0050  100.0000    0.0100

ans =

    0.0049   99.4029    0.0113
```



info_fmin.m

```
% This code uses fminsearch() to identify the three parameters in the
% spread of information model:
%   u_t = c(M - u)u + k(M - u) where
%               u = u(t) in the number (or percent) who know a fact,
%               M = size of the population (or 100 percent),
%               c = "gossip" coefficient and
%               k = "advertising" coefficient.
% The data is given in the vector ud and is adjusted by a random
variable.
% In the least squares function only the first 8 data points are used.
% The general form of the solution to the above DE can be found
% by either separation of variables or dsolve().
%
% function r =info_sol(x1,x2,x3,t)
%    u0 = 1.;
%    x4 = (1/(x1*x2+x3))*log((u0*x1+x3)/(x2-u0));
%    E = exp((t+x4)*(x1*x2+x3));
%    r = (x2*E-x3)/(x1+E);
%
%function ls = ls_info(xx)
%    global p
%    t = 0:1:7;
%    ls = 0;
%    for i = 2:8
%        ls = ls + (p(i)-info_sol(xx(1),xx(2),xx(3),t(i)))^2;
%    end
%
clear
format long
global p
ud = [ 1 2.9 6.2 10.5 17.3 26.6 38.1 50.9 ...
       63.5 74.6 82.9 89.0 93.1 95.7 97.4 98.4 99.6 ];
p = ud + .2*rand(1,17) - .1;
td = 0:1:16;
%
options = optimset('MaxIter', 2000, 'MaxFunEvals', 5000);
[xx lsval]=fminsearch('ls_info',[.001 99 .001 ],options)
%
t = 0:1:20;
for i = 1:21
    informed(i)= info_sol(xx(1),xx(2),xx(3),t(i));
end
plot(td(1:8),p(1:8),'*',td,p,'x',t, informed)
```

```
>> info_fmin

xx =

  1.0e+002 *

   0.000049721304113   1.001516345974975   0.000101542136732

lsval =

   0.073574425606748
```