>

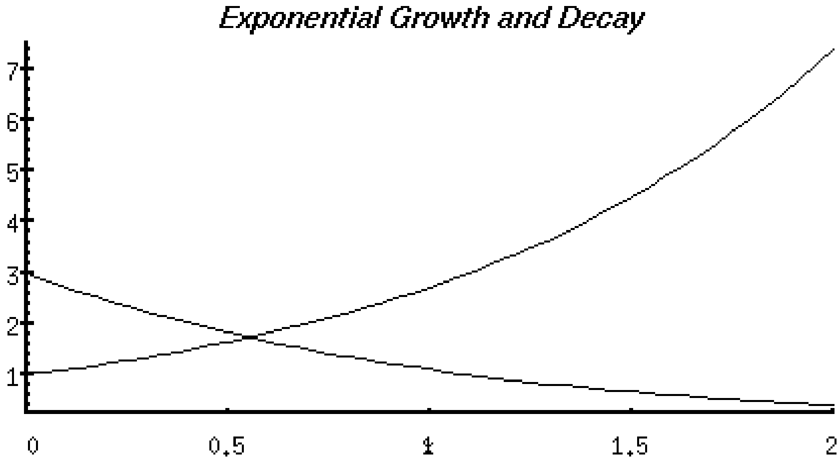
Maple Worksheet to Accompany the Lecture Presented by Prof. Jim Herod

Prepared by Doug Meade (from a text file by Jim Herod)
6 April 1995

File: herod.ms

Figure 1 -- Exponential Growth and Decay

```
> restart;
> plot({exp(x), 3*exp(-x)}, x=0..2, title=`Exponential Growth and Decay`);
```



>

Figure 2 -- Logistic Growth

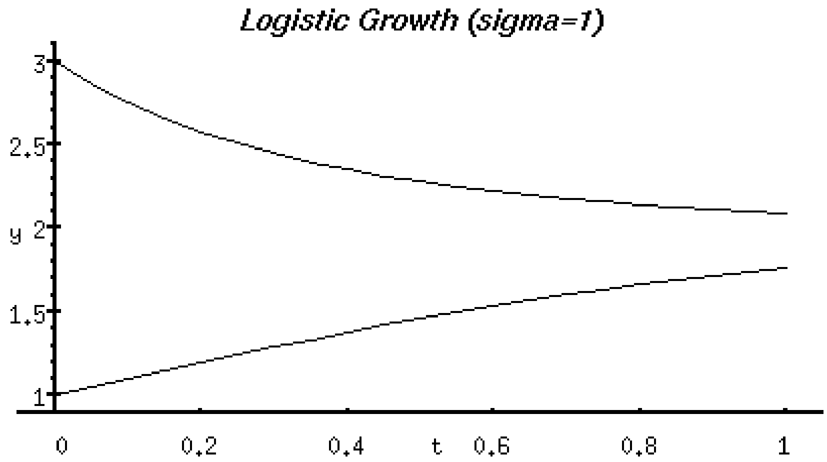
In each plot, what is the behavior of the solutions as t increases to infinity?

```
> restart;
  with( DEtools ):
> logistic := diff( y(t), t ) = y(t)*(2-y(t)^sigma);
```

$$logistic := \frac{\partial}{\partial t}\,y(t) = y(t)\left(2 - y(t)^{\sigma}\right)$$

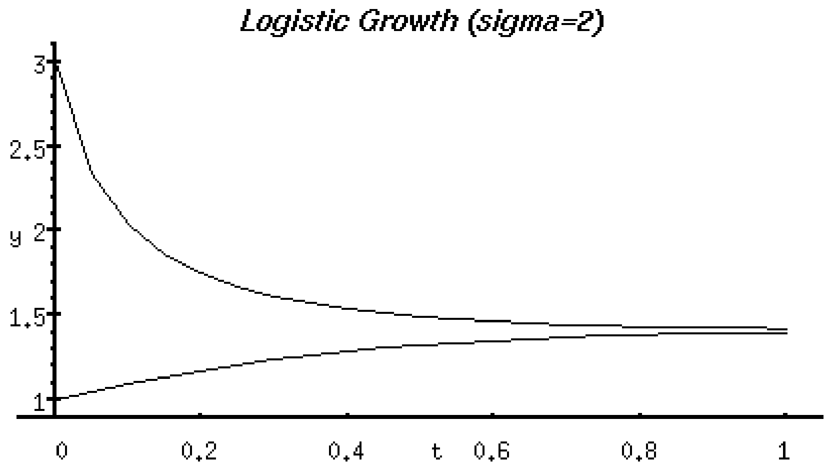
```
> sigma := 1;
  DEplot( logistic, [t,y], t=0..1, { [0,1], [0,3] },
          title=`Logistic Growth (sigma=1)` );
```

$$\sigma := 1$$




```
> sigma := 2;
  DEplot( logistic, [t,y], t=0..1, { [0,1], [0,3] },
          title=`Logistic Growth (sigma=2)` );
```

$$\sigma := 2$$




```
>
```

DELAY DIFFERENTIAL EQUATIONS

Consider now the differential equation

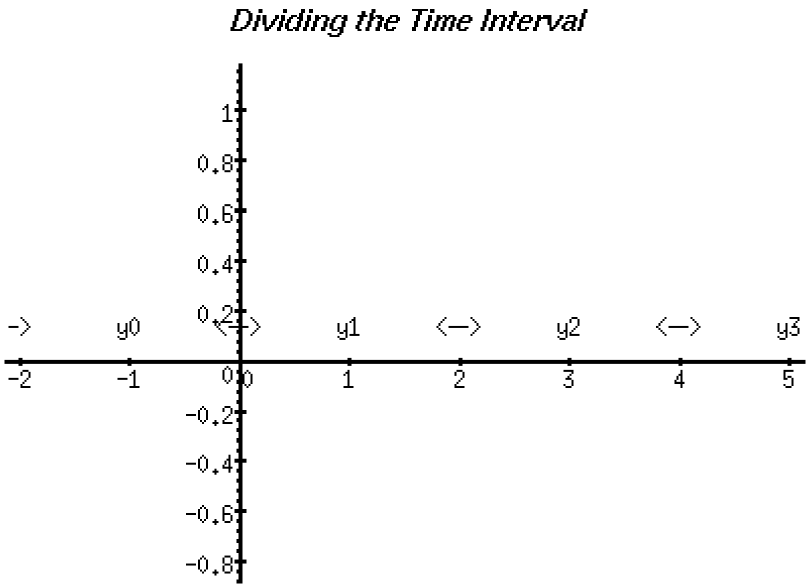
$$y'(t) + y(t) = y(t-2) * (2 - y(t-2))$$

Note that this is a DELAY differential equation because the terms on the right hand side are evaluated in the past (the rate of change depends on the size of the population 2 time units in the past). If we proceed carefully, and in 2 time unit steps, then at each stage will KNOW the right-hand side. For example, if y is known for -2 <= t <= 0 and can solve the first-order linear ODE on the interval 0 <= t <= 2. Then, knowing the solution on [0, 2], we now have the data to compute the solution on another time interval of length 2. See Figure 3 for a graphical view of this step-by-step process.

Note that at each stage the ODE is first-order and linear. The integrating factor is :
mu(t) = exp(t)

Figure 3 -- Schematic Figure for Dividing the Time Interval

```
> restart;
  with( plots ):
> textplot({[-1,0.15,y0],[1,0.15,y1],[3,0.15,y2],
           [5,0.15,y3],[-2,0.15,`->`],[-0.1,0.15,`<-`,align=LEFT],
           [0.1,0.15,`->`,align=RIGHT],[1.9,0.15,`<-`,align=LEFT],
           [2.1,0.15,`->`,align=RIGHT],[3.9,0.15,`<-`,align=LEFT],
           [4.1,0.15,`->`,align=RIGHT]},
           title=`Dividing the Time Interval`);
```




```
>
```

Figure 4 -- Logistic ODE with Delay

Consider now the differential equation
*y'(t) + y(t) = y(t-2) * (2 - y(t-2))*
Note that this is a DELAY differential equation because the terms on the right hand side are evaluated in the past (the rate of change depends on the size of the population 2 time units in the past). If we proceed carefully, and in 2 time unit steps, then at each stage will KNOW the right-hand side and can solve the first-order linear ODE using the integrating factor:
mu(t) = exp(t)

```
> restart;
  with(plots):
> sigma := 1;
```

$$\sigma := 1$$

```
> f.0:=t->0.8;
```

$$f0 := .8$$

```
> n:=0:
  exp(-t)*f.n(0)+int(exp(-t+s)*f.n(s-2)*(2-f.n(s-2)^sigma),s=0..t):
  f.(n+1):=unapply(",t);
```

$$f1 := t \to .8\, e^{(-t)} + .9600000000 - .9600000000\, e^{(-1.\, t)}$$

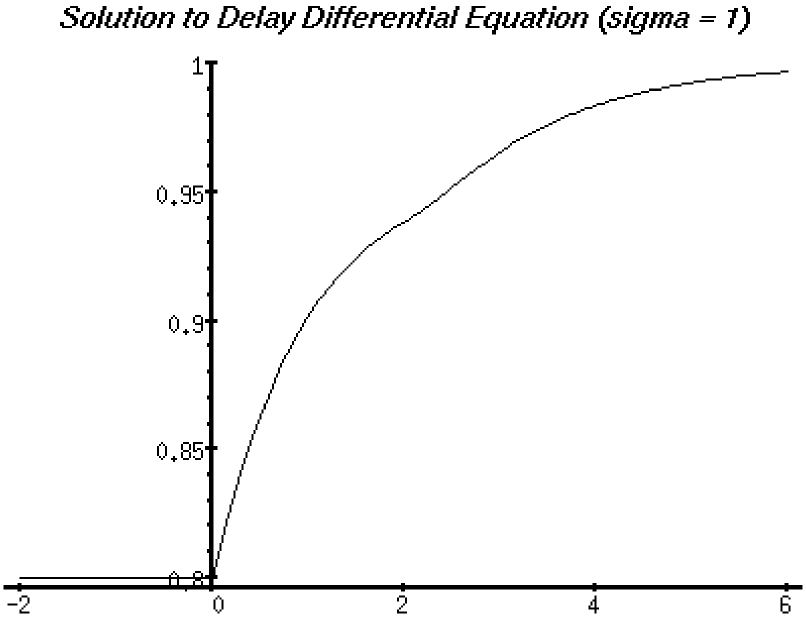
```
> n:=1:
  exp(-t+2)*f.n(2)+int(exp(-t+s)*f.n(s-2)*(2-f.n(s-2)^sigma),s=2..t):
  f.(n+1):=unapply(",t);
```

$$f2 := t \to e^{(-t+2)}\,(.8\, e^{(-2)} + .8300781281) + .01280000000\, \frac{-7.389056099\, e^t\, t + 109.1963001 + 78.\,(e^t)^2}{(e^t)^2} - 7.377233609\, \frac{1}{e^t}$$

```
> n:=2:
  exp(-t+4)*f.n(4)+int(exp(-t+s)*f.n(s-2)*(2-f.n(s-2)^sigma),s=4..t):
  f.(n+1):=unapply(",t);
```

$$f3 := t \to e^{(-t+4)} \left(e^{(-2)}\,(.8\, e^{(-2)} + .8300781281) + .01280000000\, \frac{-29.55622440\, e^4 + 109.1963001 + 78.\,(e^4)^2}{(e^4)^2} - 7.377233609\, \frac{1}{e^4} \right) + .1907813114\, 10^{-20} \Big(-.315520264\, 10^{19}\,(e^t)^3\, t + .36169183\, 10^{22}\,(e^t)^2 + .5241590136\, 10^{21}\,(e^t)^4 - .892213389\, 10^{23}\, e^t + .189014583\, 10^{22}\,(e^t)^2\, t + .1017500327\, 10^{25} + .2559999999\, 10^{21}\,(e^t)^2\, t^2 - .5861004441\, 10^{18}\,(e^t)^3\, t^2 - .2795425281\, 10^{23}\, e^t\, t \Big) \Big/ (e^t)^4 - 54.97300317\, \frac{1}{e^t}$$

```
> plot({[t,f.0(t),t=-2..0],
        [t,f.1(t),t=0..2],
        [t,f.2(t),t=2..4],
        [t,f.3(t),t=4..6]},
        title=‘Solution to Delay Differential Equation (sigma = ‘.sigma.‘)‘);
```

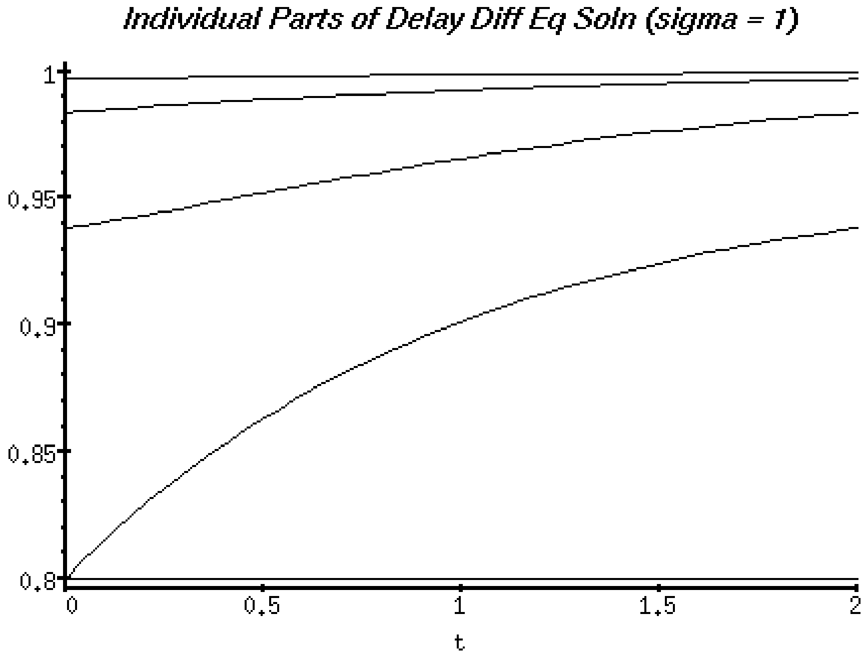



```
> display( [ seq( plot( { seq( [t,f.j(t),t=2*(j-1)..2*j], j=0..i ) } ), i=0..3 ) ],
          insequence=true, title=‘Animated Solution to Delay DE‘ );
>
```

Figure 5 -- Logistic ODE with Delay (using a loop)

Notice the pattern in each step of the above solution process. This repeated set of steps can be put inside a single for loop.

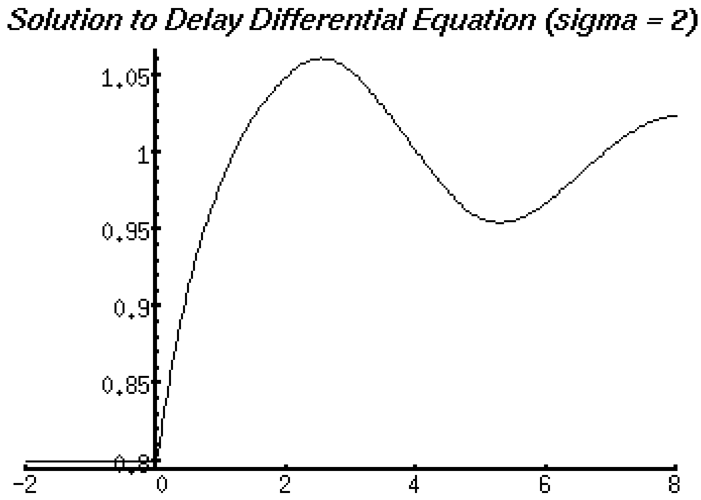
```
> f.0:=t->0.8:
> for n from 0 to 3 do
     exp(-t+2*n)*f.n(2*n)+int(exp(-t+s)*f.n(s-2)*(2-f.n(s-2)^sigma),s=2*n..t);
     f.(n+1):=unapply(",t);
  od:
> plot({f.0(t-2),f.1(t),f.2(t+2),f.3(t+4),f.4(t+6)},t=0..2,
        title=‘Individual Parts of Delay Diff Eq Soln (sigma = ‘.sigma.‘)‘);
```



>

Figure 6 -- Logistic ODE with Delay, now letting Maple do the solution of the ODE

```
> restart;
  with( plots ):
  readlib(spline):
> sigma:=2:
  N:=4:
> x0seq:=[seq(j/5,j=0..10)]:
  f.0:=t->0.8:
  J.0 := plot( f.0,-2..0 ):
> for n from 1 to N do
    ODE := diff(y(t),t)+y(t)=''f.(n-1)(t-2)''*(2-''f.(n-1)(t-2)''^sigma);
    IC := y(2*(n-1))=f.(n-1)(2*(n-1));
    sol:=dsolve({ ODE, IC }, y(t), numeric, output=listprocedure);
    g:=rhs(sol[2]);
    xseq:=map(t->t+2*(n-1),x0seq);
    gxseq:=map(g,xseq);
    F:=spline(xseq,gxseq,t,cubic);
    f.n:=`spline/makeproc`(F,t);
    J.n:=plot(f.n,2*(n-1)..2*n);
  od:
> display({J.0,J.1,J.2,J.3,J.4},
        title=`Solution to Delay Differential Equation (sigma = `.sigma.`)`);
```



>

And, for the finale, let's look at a well-known model for the spread of an infectious disease.

In this model it is assumed that a population with a fixed size is divided into three subgroups:
SU - those SUsceptible to catch the disease
IN - those currently infected with the disease
RE - those who have recovered from the disease (and now have immunity)

Those in the susceptible class become infected at a rate proportional to the product of SU and IN. An infected individual recovers (or is removed) from the influence of the disease at a rate proportional to the number of individuals currently infected with the disease.

The differential equation for each subgroup is in the same form as the container problems we studied earlier this semester:

rate of change = rate in - rate out

```
> restart;
  with(share): readshare(ODE,plots):
  with(plots):
See ?share and ?share,contents for information about the share library
```

```
> r:=1:
  a:=1:
> rsS:=(t,SU,IN,R) -> -r*SU*IN;
  rsI :=(t,SU,IN,R) ->   r*SU*IN - a*IN;
  rsR:=(t,SU,IN,R) ->              a*IN;
```

$$rsS := (t, SU, IN, R) \to -r\,SU\,IN$$

$$rsI := (t, SU, IN, R) \to r\,SU\,IN - a\,IN$$

$$rsR := (t, SU, IN, R) \to a\,IN$$

```
> output1:=rungekuttahf( [ rsS, rsI, rsR ], [ 0, 2.8, 0.2, 0 ], 0.1, 200):
>
```

Just to illustrate how the output from the above command can be used, let's look at the solutions at the first and last time for which the solution was computed. Thinking of time as DAYS, this shows the population on the first day (t=0) and almost 7 months later (t=200).

```
> output1[0];
```

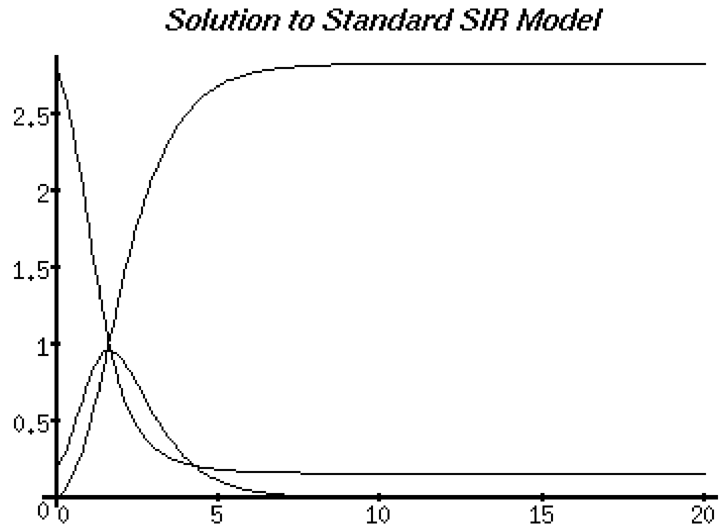
$$[0, 2.800000000000000, .2000000000000000, 0]$$

```
> output1[200];
```

$$[20.00000000000001, .1642964321387427, .4396652173760199\ 10^{-6}, 2.835703128196039]$$

Now, let's plot all three components on the same set of axes.

```
> plot( { makelist(output1,1,2), makelist(output1,1,3), makelist(output1,1,4) },
        xtickmarks=5, ytickmarks=5, title='Solution to Standard SIR Model' );
```




```
>
```

Here is the same model, but with the added feature that the immunity is not permanent. That is, members of the recovered class become susceptible at a rate proportional to the size of the recovered class on the previous day..

```
> restart;
  readlib(spline):
  with( plots ):
> r:=1;
  a:=1;
```

$$r := 1$$

$$a := 1$$

```
> f.0:=t->2.8;
  g.0:=t->0.2;
  h.0:=t->0;
  J.0 := plot( {f.0,g.0,h.0}, -1..0 ):
```

$$f0 := 2.8$$

$$g0 := .2$$

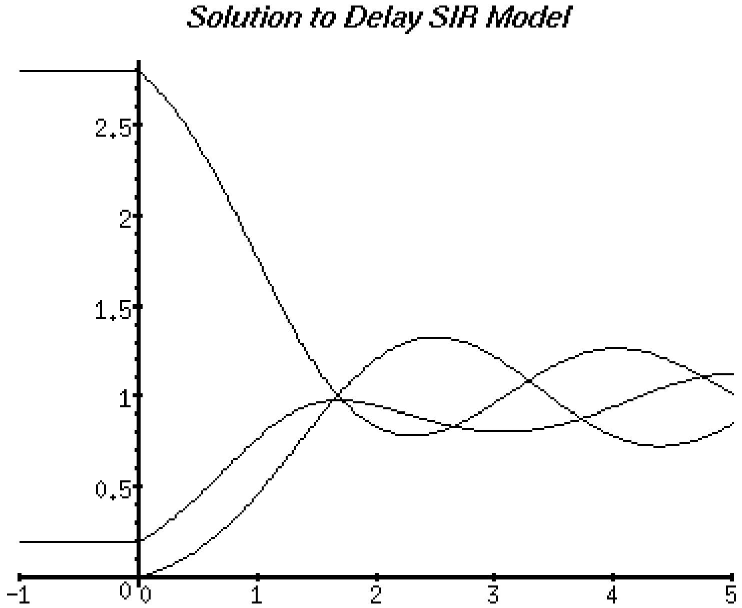
$$h0 := 0$$

```
> x0seq:=[seq(j/5,j=0..5)]:
  N:=5;
```

$$N := 5$$

```
> for n from 1 to N do
    ODE := diff(SU(t),t)=-r*SU(t)*IN(t)+''h.(n-1)(t-1)'',
            diff(IN(t),t)=r*SU(t)*IN(t)-a*IN(t),
            diff(R(t),t)=a*IN(t)-''h.(n-1)(t-1)'';
    IC    := SU(n-1)='f.(n-1)(n-1)',
            IN(n-1)='g.(n-1)(n-1)',
            R(n-1)='h.(n-1)(n-1)';
    sol:=dsolve({ ODE, IC }, { SU(t), IN(t), R(t) }, numeric, output=listprocedure):
    f:=subs( sol, SU(t) ):
    g:=subs( sol, IN(t) ):
    h:=subs( sol, R(t) ):
    xseq:=map(t->t+n-1,x0seq):
    fxseq:=map(f,xseq);
    gxseq:=map(g,xseq);
    hxseq:=map(h,xseq);
    F:=spline(xseq,fxseq,t,cubic):
    G:=spline(xseq,gxseq,t,cubic):
    H:=spline(xseq,hxseq,t,cubic):
    f.n:=`spline/makeproc`(F,t):
    g.n:=`spline/makeproc`(G,t):
    h.n:=`spline/makeproc`(H,t):
    J.n:= plot({f.n,g.n,h.n}, (n-1)..n):
  od:
```

```
> display( [ J.0, J.1, J.2, J.3, J.4, J.5], insequence=true,
          title=‘Animated Solution to Delay SIR‘);
> display( [ seq( display( { seq( J.j, j=0..i ) } ), i=0..N ) ], insequence=true,
          title=‘Animated Solution to Delay SIR‘ );
> display( { J.0, J.1, J.2, J.3, J.4, J.5},
          title=‘Solution to Delay SIR Model‘);
```




```
>
```

END OF WORKSHEET

```
>
```