Tom Green Reg number:1202807

Lab 5

1) **Starting with the sample programs test23.txt and testfun.txt solve the IVP**

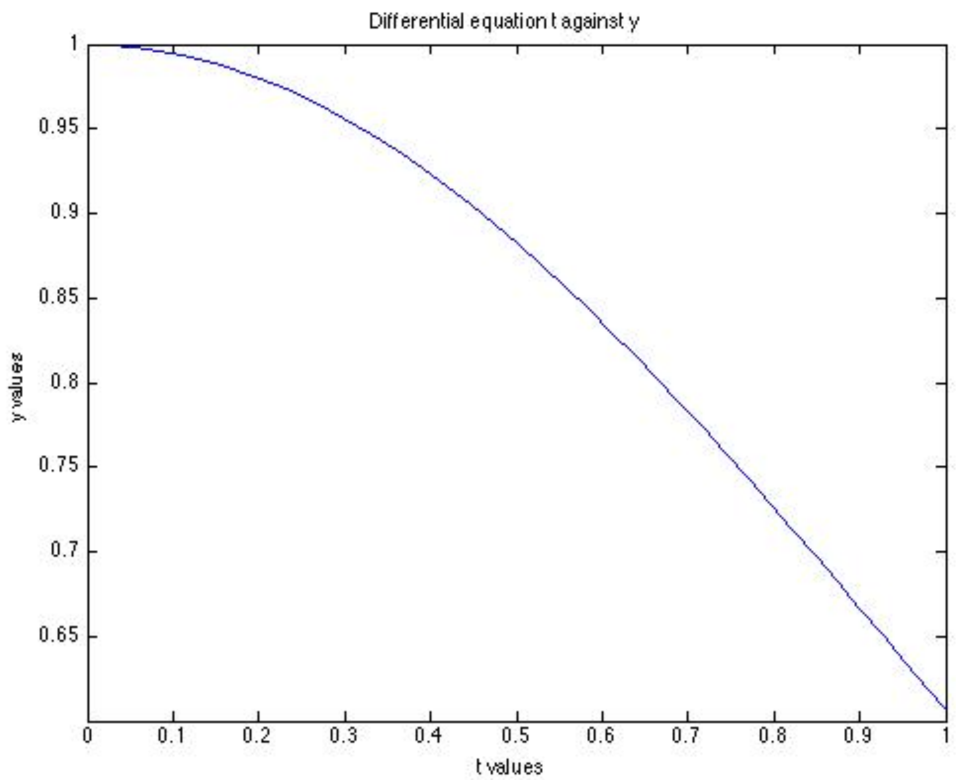
$\frac{dy}{dt} = -ty$ with y(0)=1

On the interval [0,1]

Running the programs I get after 33 steps:

```
t-values:
         0
    0.0548
    0.1096
    0.1561
    0.1958
    0.2321
    0.2663
    0.2989
    0.3303
    0.3608
    0.3905
    0.4196
    0.4482
    0.4763
    0.5040
    0.5313
    0.5584
    0.5853
    0.6120
    0.6385
    0.6649
    0.6912
    0.7174
    0.7435
    0.7696
    0.7958
    0.8219
    0.8481
    0.8744
    0.9008
    0.9273
    0.9539
    0.9807
    1.0000


y-values:
    1.0000
    0.9985
    0.9940
    0.9879
    0.9810
    0.9734
    0.9652
    0.9563
    0.9469
    0.9370
```

```
0.9266
0.9157
0.9045
0.8928
0.8807
0.8683
0.8556
0.8426
0.8292
0.8156
0.8017
0.7875
0.7731
0.7585
0.7437
0.7286
0.7133
0.6979
0.6823
0.6665
0.6506
0.6345
0.6182
0.6065
```



2) **The Prey Predator model can be made more realistic by arguing that the ecological system can only sustain up to a maximum population of the prey, that predators will only be seeking a kill at certain times**

(e.g. when they are awake and hungry) etc. This leads to a revised system of equations which focuses on population per unit area and which can be written in the form:

$$\frac{dx}{dt} = x\left(1 - \frac{x}{k}\right) - \frac{mxy}{1+x}$$

$$\frac{dy}{dt} = -cy + \frac{mxy}{1+x},$$

Use ode23 to solve this problem over the time range [0,200] in the case when the initial values are given to be x(0)=y(0)=0.5.

My value for b is 807 that makes my values for m,c and k to be:

$$m = 9.5 + 0.001 * 807 = 10.307$$
$$c = 5.5 - 0.001 * 807 = 4.693$$
$$\bar{k} = \frac{10.307 + 4.693}{10.307 - 4.693} \approx 2.67189$$
$$k = 1.3 * 2.67189 \approx 3.473459$$

I started by creating a file that essentially has the same purpose as testfun.m in q1, here is the code:

```
function dy = q2functions(t,y);

b=807.;
m=9.5+0.001*b;
c=5.5-0.001*b;
kbar=(m+c)/(m-c);
k=1.3*kbar;

% y and dy are vector columns equivalent to f(t,y) of the system of
equations
% y(1,1) = x in the original question
% y(2,1) = y in the original question

dy = zeros(2,1);
dy(1,1) = y(1,1)*(1.-y(1,1)/k)-(m*y(1,1)*y(2,1))/(1.+y(1,1));
dy(2,1) = -c*y(2,1)+(m*y(1,1)*y(2,1))/(1.+y(1,1));
```

So as you can see this file creates the functions given in the question.

I then created a second function that would be able to plot the iterations for these ODEs, for x, y and t.

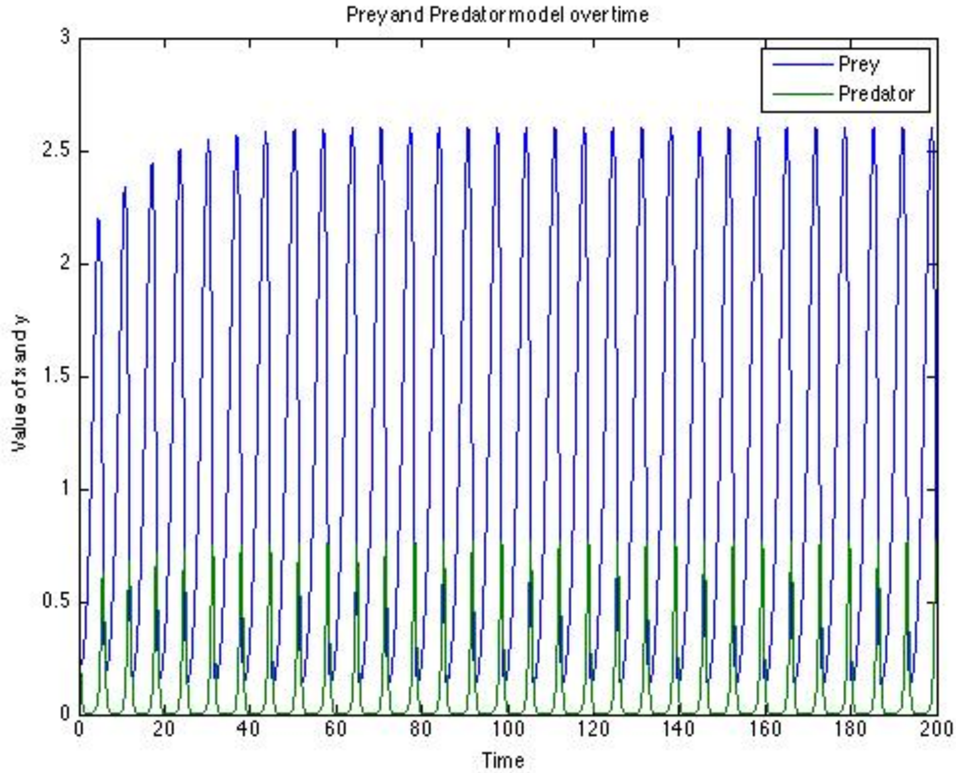
```
t_begin = 0.0;
t_end = 200.0;
tspan = [t_begin t_end];

y0 = [0.5; 0.5];
Rtol = 1.0e-6;

options = odeset('RelTol',Rtol);
```

Tom Green Reg number:1202807

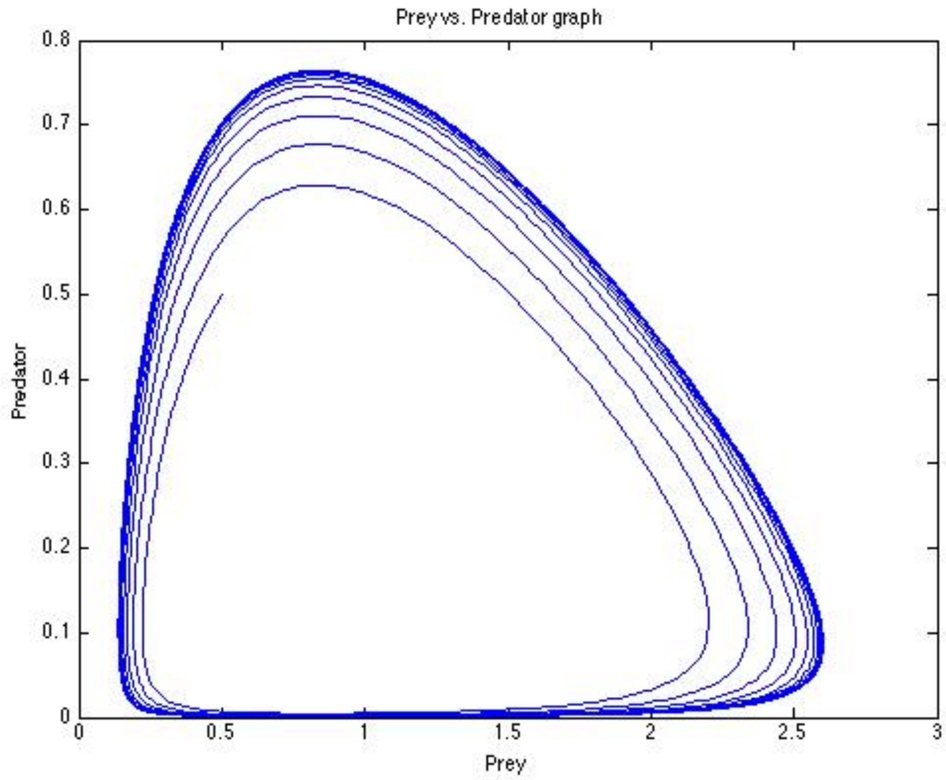
```
[T Y] = ode23(@q2functions,tspan,y0,options);

[N,M] = size(Y);

N1 = N - 1;

disp(' ')
disp('RelTol:')
disp(Rtol)
disp(' ')

disp('Number of steps:')
disp(N1)
disp(' ')

disp('t-values:')
disp(T)
disp(' ')

disp('y-values:')
disp(Y)
disp(' ')

figure(1)
plot(T,Y)

figure(2)
plot(Y(:,1),Y(:,2))
```

This gets a graph of

Tom Green Reg number:1202807



This shows the population of prey and predator over a 200 year period, as you would expect the population of the prey has larger fluxuations than that of the predator.



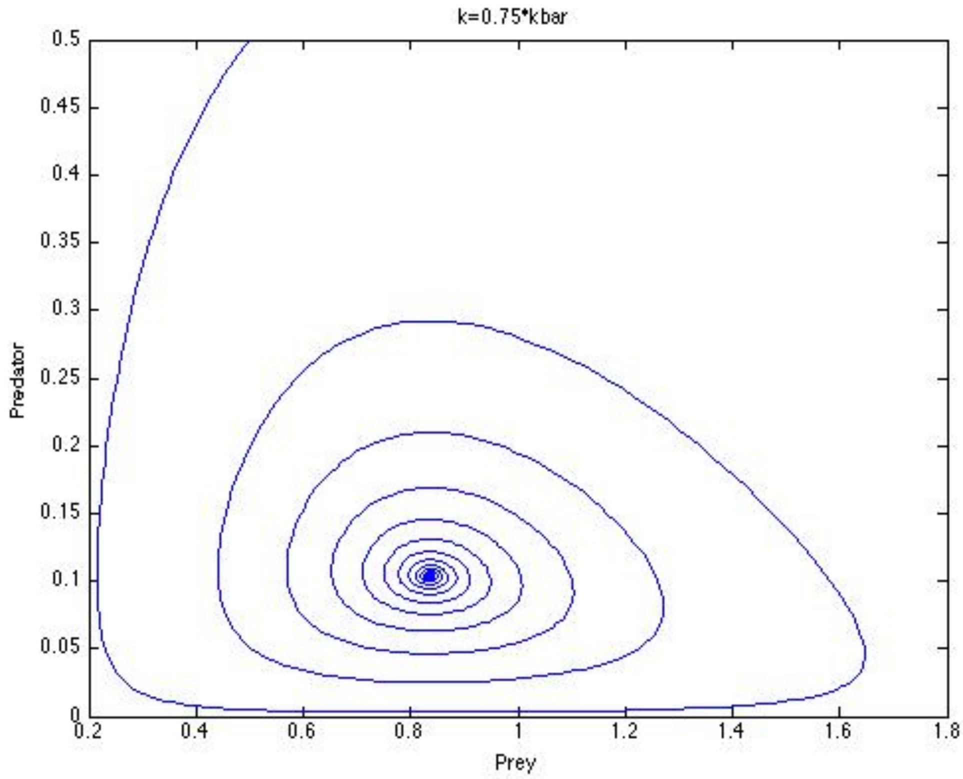
Tom Green Reg number:1202807

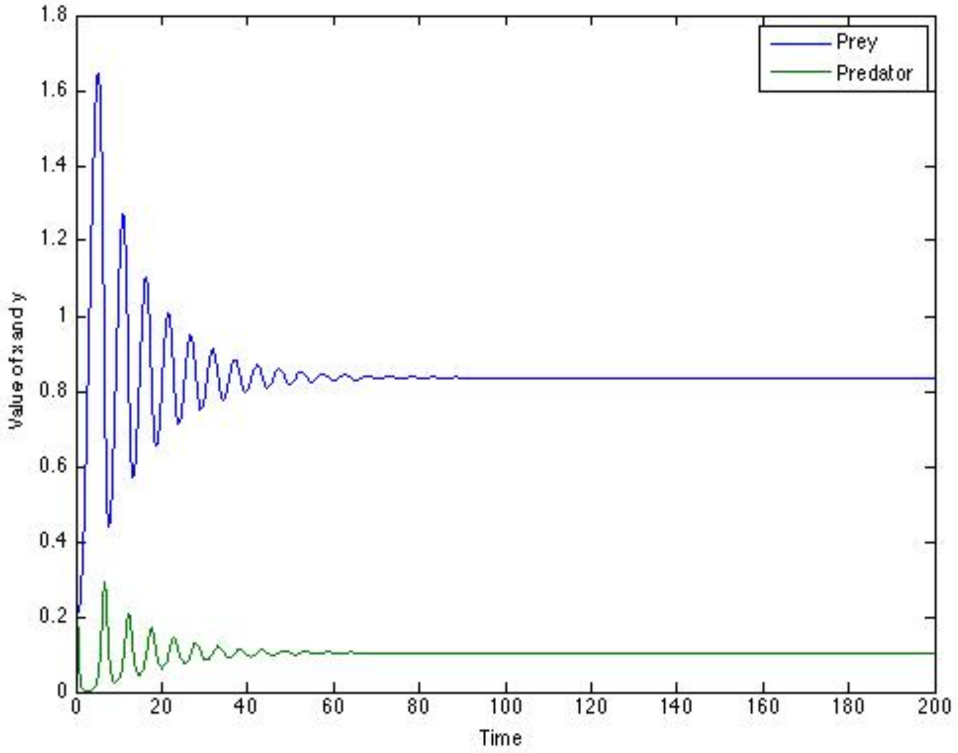
These graphs are what we would expect for a simple ecosystem that involves both a predator and a prey, initially the prey would have larger population than the predator thus allowing the predator to easily hunt for food so they will reproduce more. All of this hunting will lower the population of the prey thus making it harder for the predator (with a now much larger population) to hunt for food, so the predators population will decrease. With less predators the population of the prey will once again increase and the circle of the ecosystem will repeat itself.

Then repeating this for

$$k = 0.75 * 2.67189 \approx 2.0039$$

You get these graphs





What we can assume from these graphs is that k is the rate of reproduction, now that it has been taken to below 1 that means that they are reproducing at a much slower rate and thus the circle of the ecosystem is damaged. For the first 40 years there is fluctuations for both the predator as well as the prey but then they both just level out and have constant levels.

3) **Let b be the three digit integer from your registration card in q2. Using the program Lagrange.txt calculate the coefficients of the quadratic polynomial which interpolates the function:**

$$f(x) \equiv (2 + 0.001 * b)\exp\left(\frac{x}{6}\right) - (0.5 + 0.001 * b)\sin\left(\frac{\pi x}{3}\right)$$

at the points x=-2, x=-1 and x=1. Repeat the excersice for a cubic polynomial at the same points plus x=2.

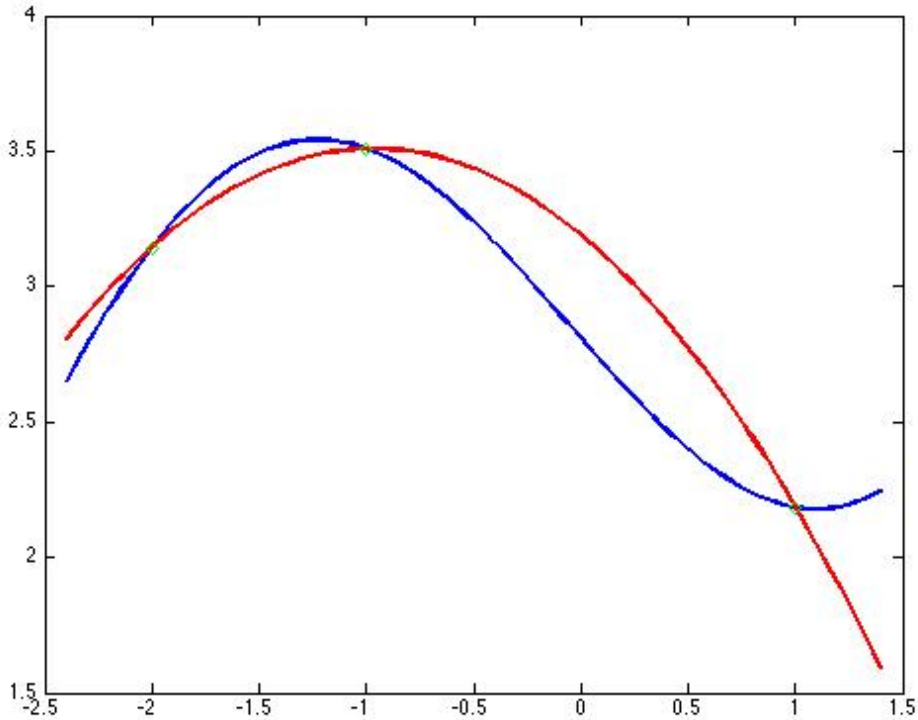
My b=807 and my code for this program was:

```
clear all

b=807;

f = inline('(2.+0.001*807)*exp(x/6.)-
(0.5+0.001*807)*sin(pi*x/3.)','x');   %
```

```
x = [-2 -1 1];
n = length(x)-1;

fval = zeros(1,n+1);

% Now get the function-values for the given data-points (or they
could be read in, instead)

for i = 1:(n+1)
  fval(i) = f(x(i));
end


% Calculate the coefficients of the Lagrange polynomials based on the
points in the vector x

L = zeros(n+1,n+1);

for k = 1:(n+1)
  V=1;
  for j=1:(n+1)
    if (k ~=j)
      V = conv(V,poly(x(j)))/(x(k)-x(j));
    end
  end
  L(k,:) = V;
end


% Now we can calculate the coefficients of the interpolating
polynomial

C = fval*L;



% Calculate the data for the two plots

m = 100;
lower = min(x) - 0.4;
upper = max(x) + 0.4;
diff = upper - lower;

h = diff / m;

xx = zeros(m+1,1);
true_f = xx;
f_interp = xx;

for i = 1:(m+1)
  xx(i) = lower + (i-1)*h;
  true_f(i) = f(xx(i));
  f_interp(i) = polyval(C,xx(i));
end

plot(xx,true_f,'LineWidth',2);

hold on;
```

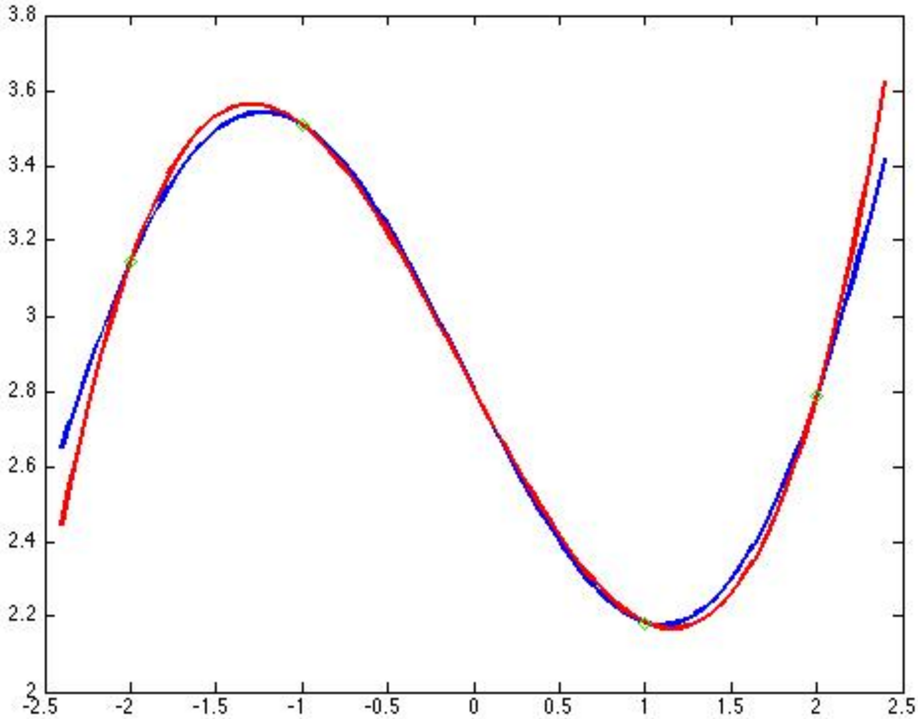
```
plot(xx,f_interp,'-r','LineWidth',2);

hold on;

plot(x,fval,'gd');
```

When I run this for the quadratic polynomial I get this plot



Then when I change it to the cubic polynomial I get

Tom Green Reg number:1202807



The function poly, is where you give the roots and it gives the coefficients of the polynomials.

The function conv multiplies two polynomials (convilusion)

Polywhile you supply the coefficients of the polynomial and it gives you p(x)