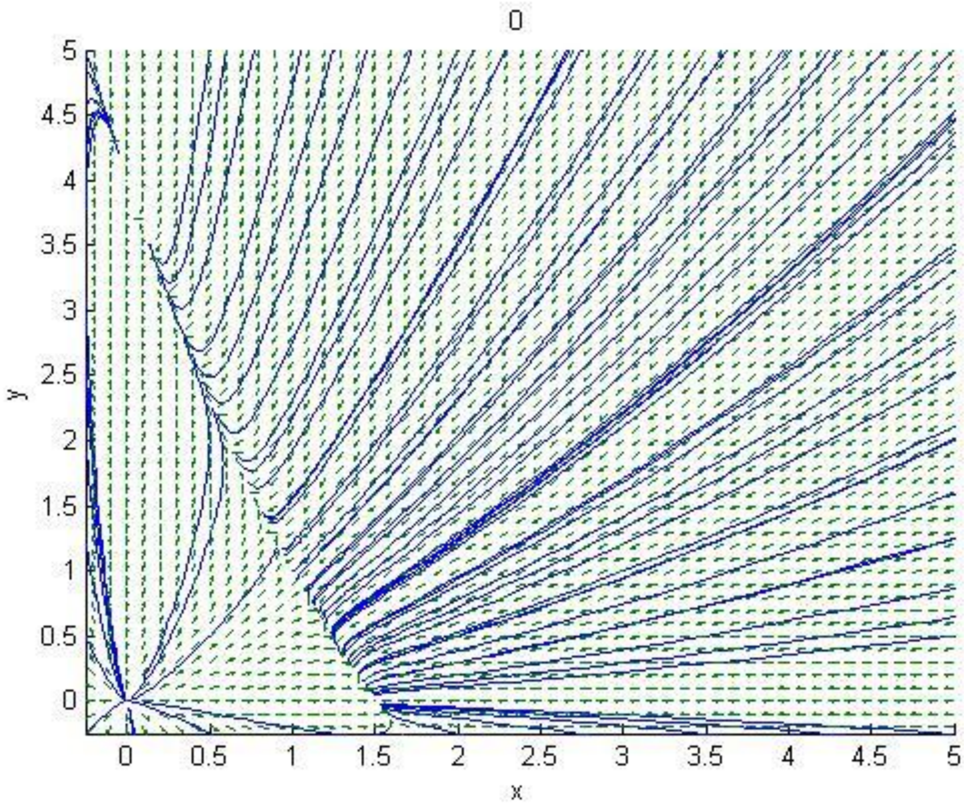
Lion vs. Crocodile

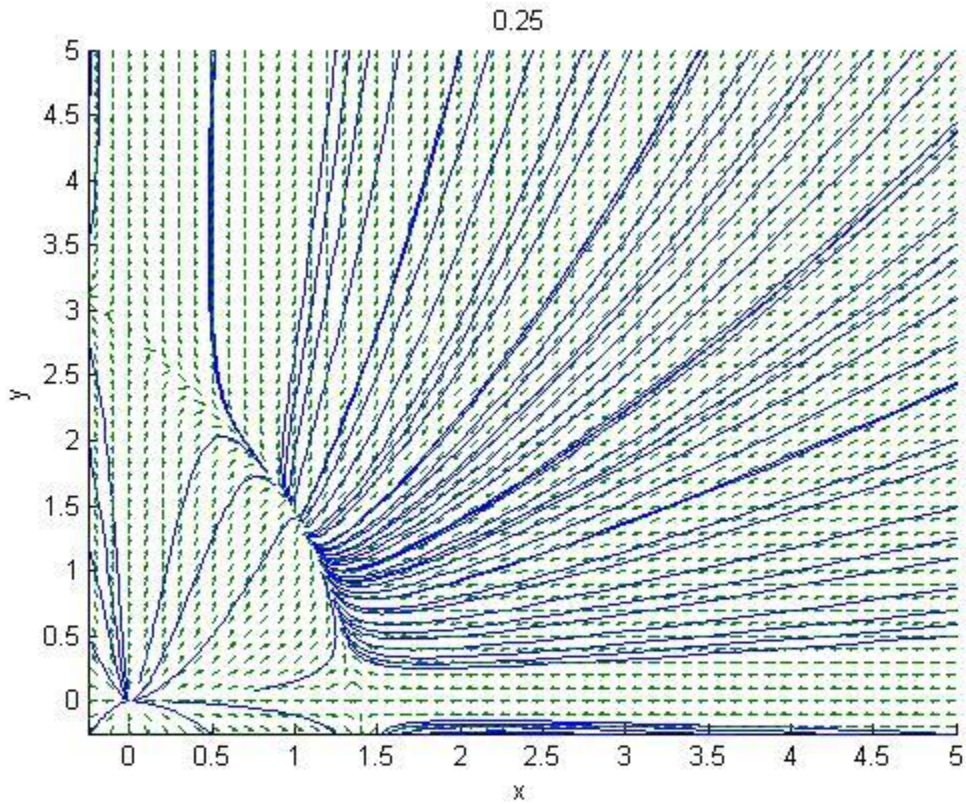
Korey Hopkins

To view this transformation visit youtube:
http://www.youtube.com/watch?v=ehZGCk79erQ

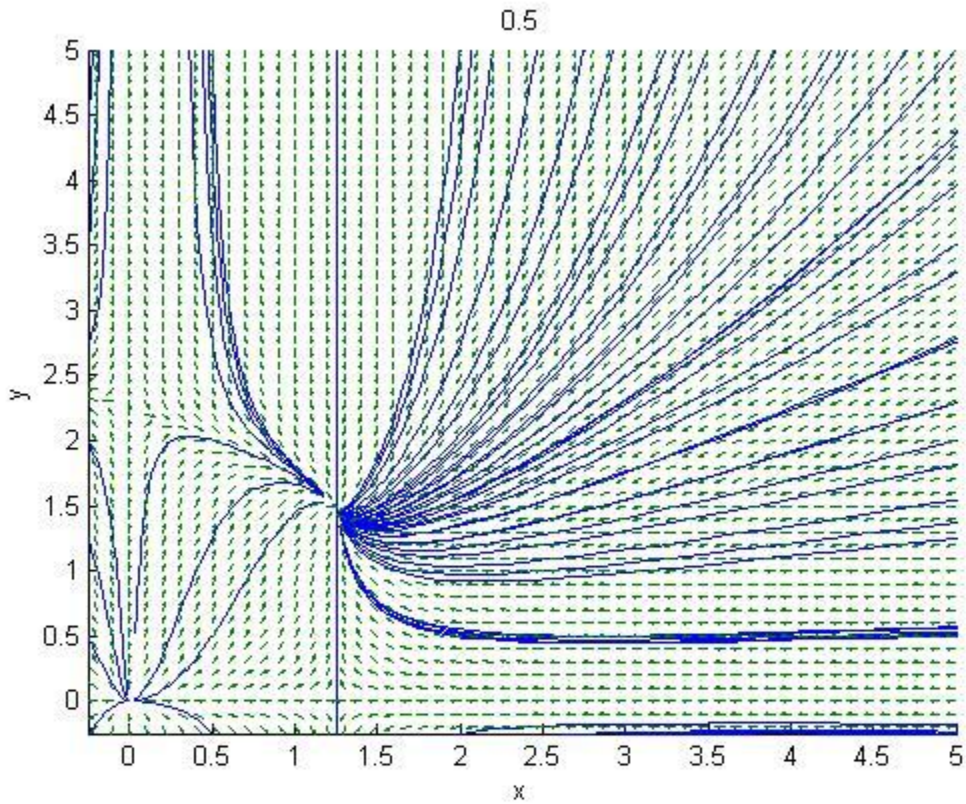

This is a model of a predator-prey relationship. To keep things interesting, we will assume a simple model containing Lions represented by **y** and Crocodiles represented by **x**. Both animals eat Gazelle, and for this simple model we will assume that Gazelle are the main food supply in this model.



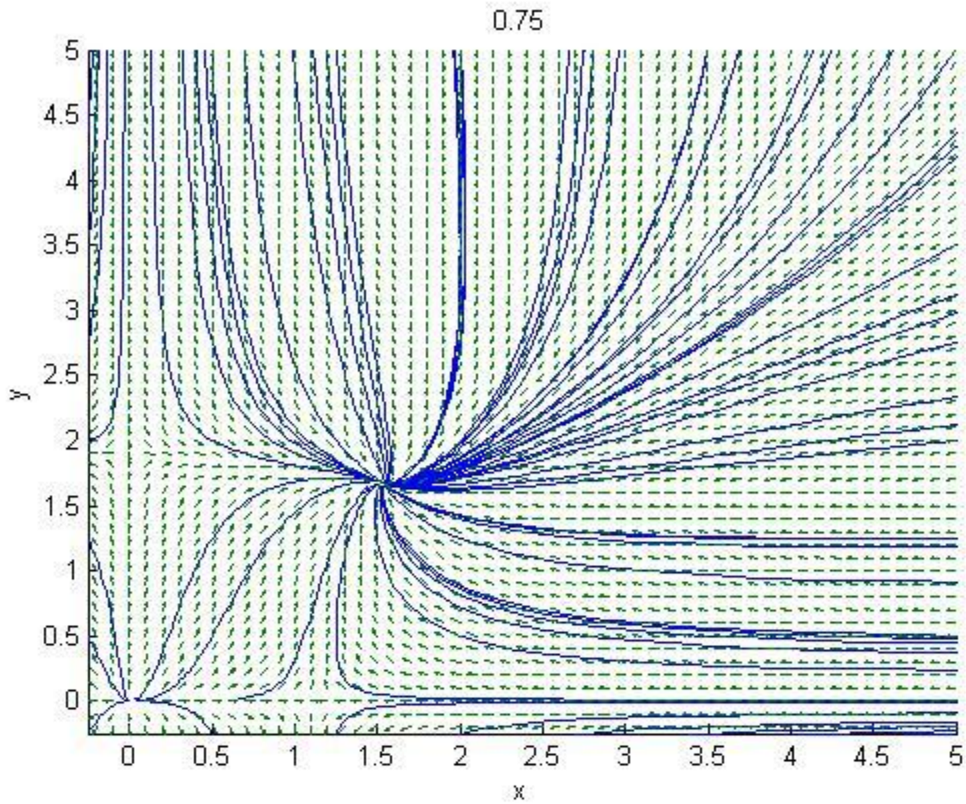
This portrait represents the initial system. Initially, there are stable points at (0,0), (0,4), (1,1), and (1.5,0). Points of 0 represent a species being extinct. Therefore, we see that we start out with an equal number of Lions and Crocodiles. All is well.



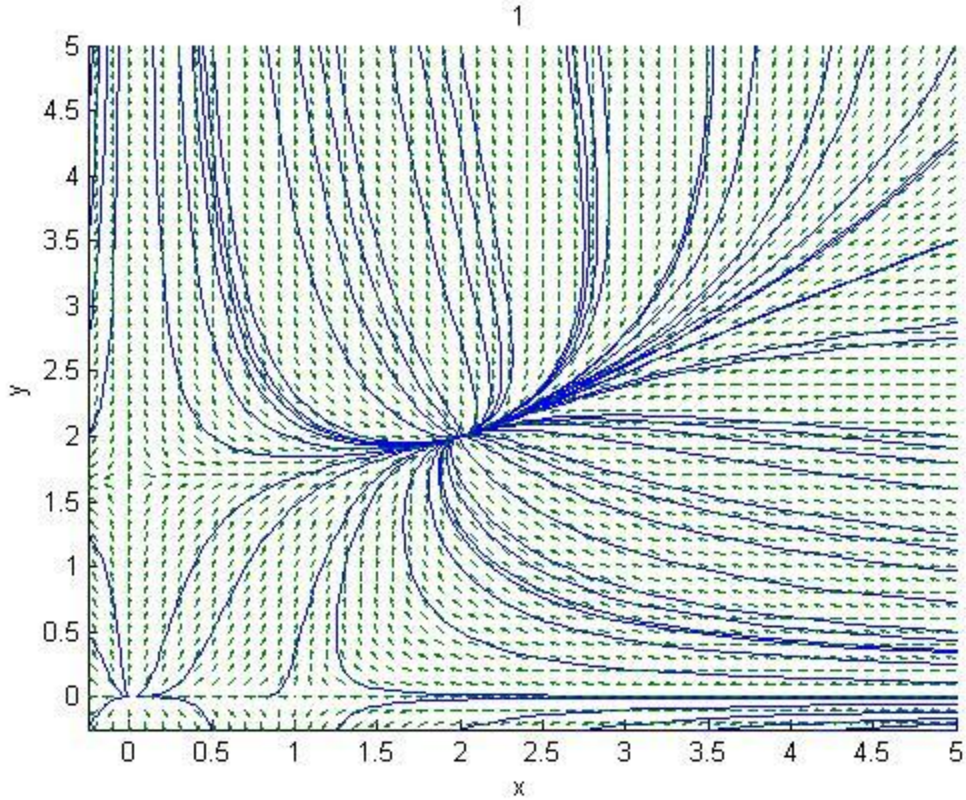
As time passes and Gazelles start frolicking via land and sea, however we see that Lions are getting most of the Gazelle by looking at the Lion's population. The Gazelle don't seem to like water as much as land.



At this point, The Lions have the upper hand on the Crocodiles. The original stable point of (1,1) has shifted to about (1,1.5). The Crocodiles cannot seem to compete with the Lions ability to catch the Gazelle on land.



The Crocodiles are starting to get smart. The stable point is starting to shift to the right faster than up. The Crocodiles want the Gazelle just as bad as the Lions so they are working on their hunting skills.



At the end of the day, the crocodiles have accomplished what they needed to do in order to catch the Gazelles. They honed their skills and had luck on their side. At this point the stable point of interest has shifted to (2,2) indicating the population of both groups doubled. The Lions and Crocodiles have managed to prosper and hunt Gazelle in harmony.

```
%Korey Hopkins
%MATH246 predator-prey model
%systems 2 and 6 from section 9.5
%problem 2
close all
clear all
syms x y
system1 = x*(1.5 - x -0.5*y);
system2 = y*(2 - 0.5*y -1.5*x);
%problem 6
system3 = x*(1-x+0.5*y);
system4 = y*(2.5 - 1.5*y + 0.25*x);

%Find and analyze critical points of inital and final systems
[xcrit,ycrit] = solve(system1, system2, x, y);
[xcri, ycri] = solve(system3, system4, x,y);

A = jacobian([system1 system2], [x y]);
B = jacobian([system3 system4], [x y]);
[VA, DA] = eig(A);
[VB, DB] = eig(B);

      disp ('Critical Points: ');
      disp ([xcrit ycrit]);
      disp ([xcri ycri]);



% graphs
%Input alpha values...
v = input('Input values between 0 and 1 (0.01) in vector form. v = ');
alpha = sort(v);
[row, col] = size(alpha);

%Graphs
if alpha(1) > 0
       ini = 1;
else
       ini = 2;
end

if alpha(col) == 1
        fin = col-1;
else
        fin = col;
end

    for n = 1:col
    A = 1 - alpha(n);
    B = alpha(n);
    [X, Y] = meshgrid ( -1:.1:5, -1:.1:5 );
    U = A*(X .* (1.5 - X - 0.5*Y))+ B*(X .* (1-X+0.5*Y));
    V = A*(Y .* (2 - 0.5*Y - 1.5*X))+ B*(Y .* (2.5 - 1.5*Y + 0.25*X));
    L = sqrt (U.^2 + V.^2);

    warning off all
```

```
f = @(t, x) [(A*(x(1)*(1.5 - x(1) -0.5*x(2))))+...
    (B*x(1)*(1-x(1)+0.5*x(2))) ;...
    (A*x(2)*(2 - 0.5*x(2) -1.5*x(1)))+...
    (B*x(2)*(2.5 - 1.5*x(2) + 0.25*x(1)))];

figure;
hold on
for a = -.25:.75:5
     for b = -.25:.75:5
      [t, xa] = ode45 (f, [0 2.5], [a b]);
     plot (xa (:,1), xa (:,2))
     [t, xa] = ode45 (f, [0 -2.5], [a b]);
     plot (xa (:,1), xa (:,2))
    end
end
axis ([-0.25 5 -0.25 5])

xlabel 'x', ylabel 'y', title (alpha(n))
quiver (X, Y, U./L, V./L, .4)
end
```