Change alpha, find out steady states:

Functions are used in the following order :

1. function GLF (x, alpha, beta nu)
2. function GLFss (xss, alpha, beta, nu)
3. WCss (alpha, beta, nu) >>>> here alpha is used to be a vector

function [yss\_hi, yss\_mid, yss\_lo] = WCss (alpha, beta, nu)

% beta,nu -- constants; alpha is a vector

yss\_hi=nan(size(alpha));

yss\_mid=nan(size(alpha));

yss\_lo=nan(size(alpha));

for i=1:numel(alpha)

if nu>=1

xss=fsolve(@(x) GLFss(x,alpha(i),beta,nu), [0.01 0.5 0.99], optimset('Display', 'off'));

yss=GLF(xss,alpha(i),beta,nu);

else

xss=fsolve(@(x) GLFss(x,alpha(i),beta,nu), [0.99 0.5 0.01], optimset('Display', 'off'));

yss=GLF(xss,alpha(i),beta,nu);

end

%label ss points

if abs(yss(1)-xss(1)) < 0.01

yss\_lo(i) = yss(1);

end

if abs(yss(2)-xss(2)) < 0.01

yss\_mid(i) = yss(2);

end

if abs(yss(3)-xss(3)) < 0.01

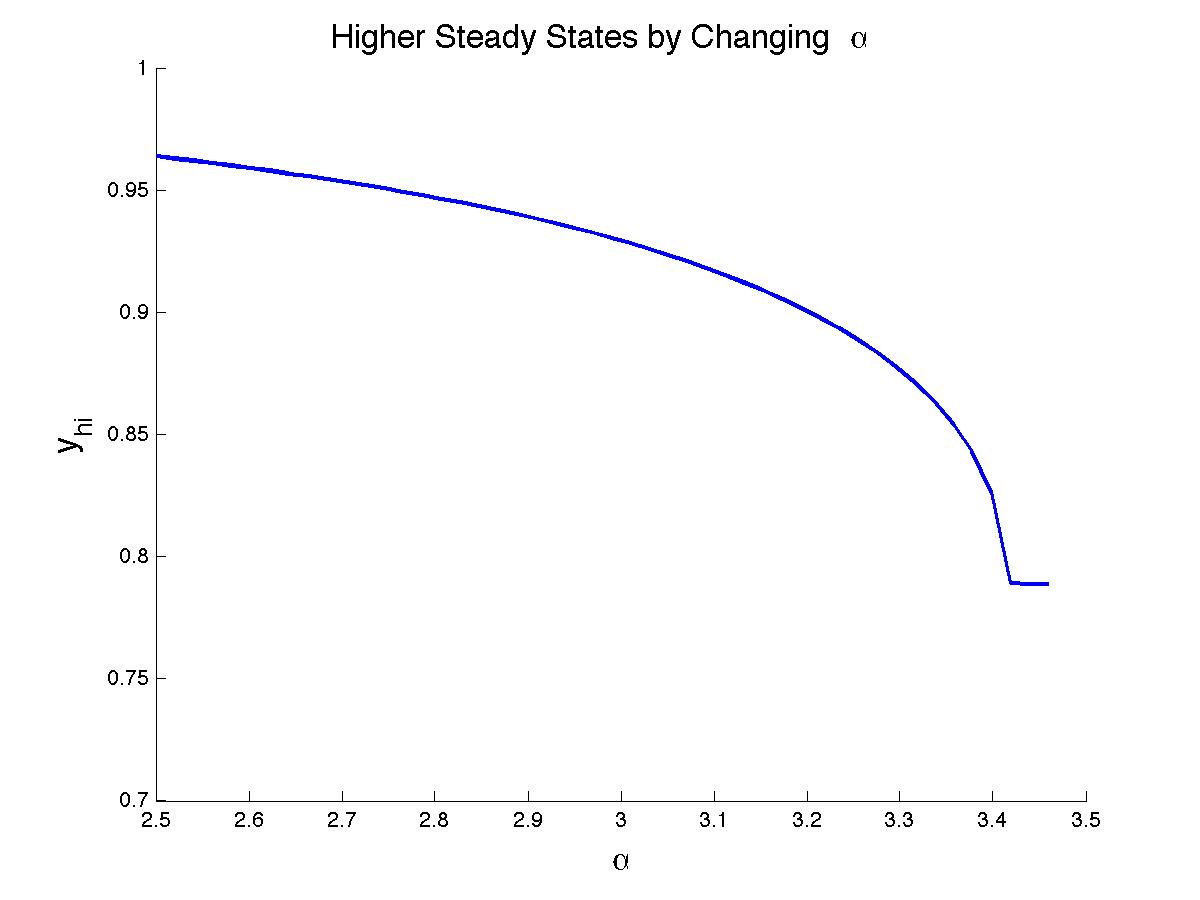
yss\_hi(i) = yss(3);

end

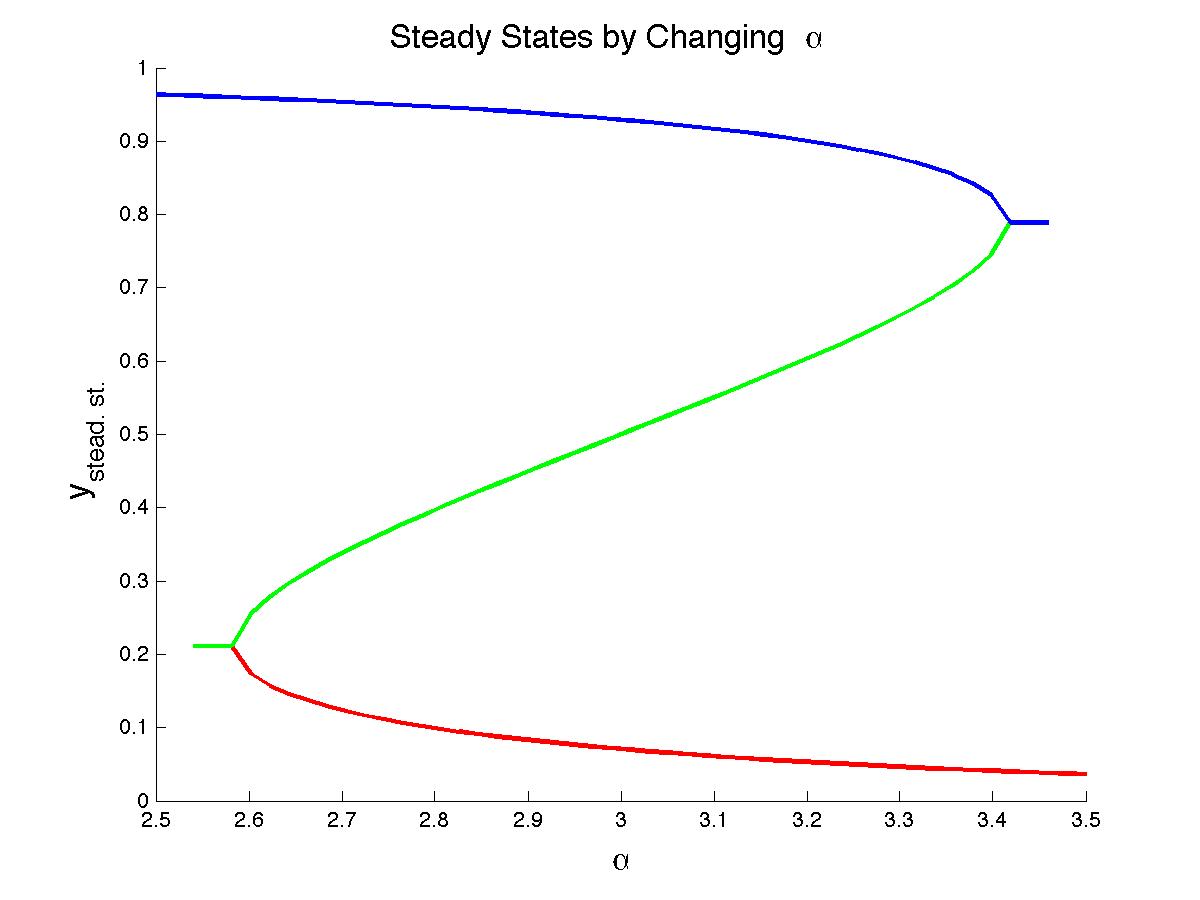
end

return;

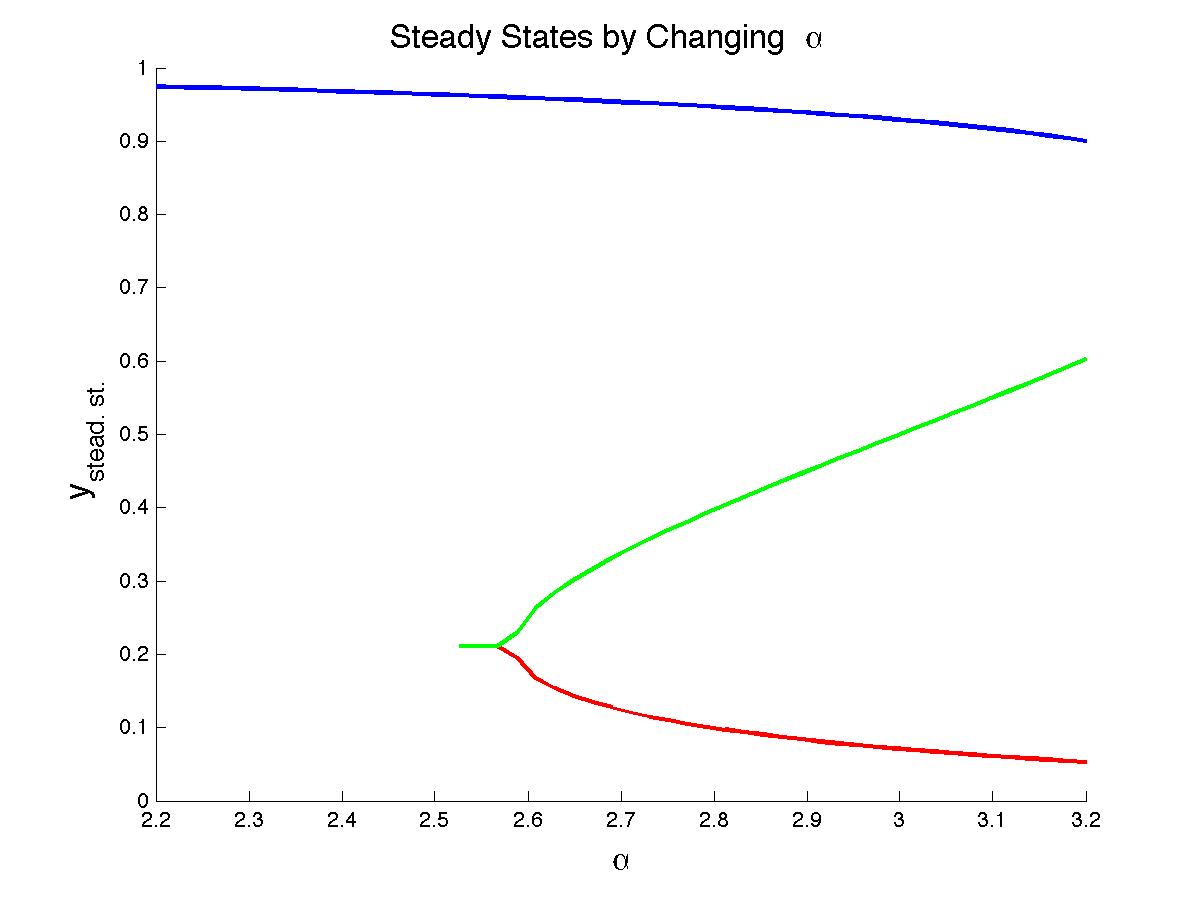
How does steady states between apprx. 0.7 and 1 look like:



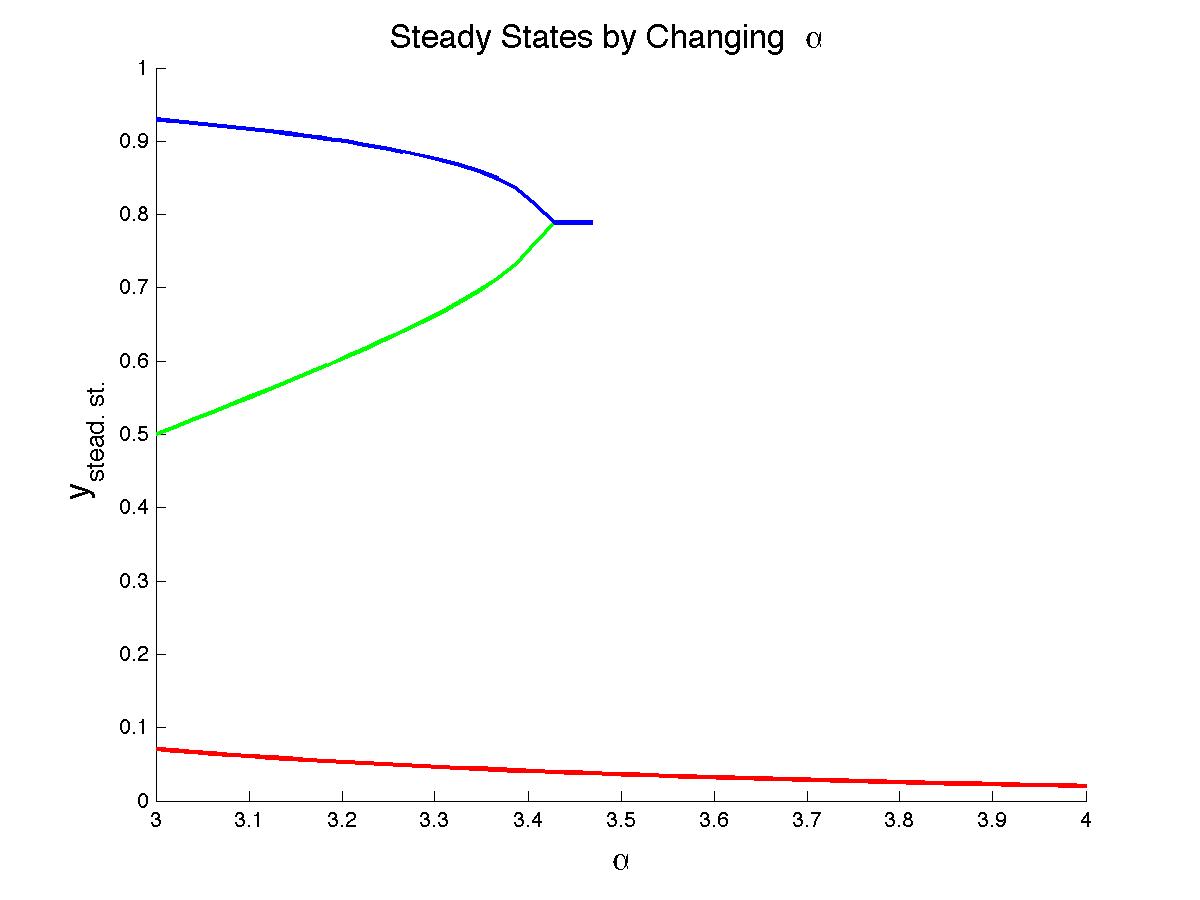
alpha\_try=linspace(2.5,3.5,50);



Change alpha : alpha\_try=linspace(2.2,3.2,50);



Change alpha: alpha\_try=linspace(3,4,50);



Note:

help isnan

ISNAN True for Not-a-Number.

ISNAN(X) returns an array that contains 1's where

the elements of X are NaN's and 0's where they are not.

For example, ISNAN([pi NaN Inf -Inf]) is [0 1 0 0].

Lets define another function WCss”, which takes input “s” and finds out steady states as shown below.

s\_level = linspace(0.1,0.9, n\_lev);

