Prepared by Dr. Ashok Kumar Pandey, IIT Hyderabad on 8 April 2011 for Nonlinear Oscillation course

Primary Resonance & Weak excitation

Equation: $\ddot{u} + \omega_0^2 u = \varepsilon\left(q\cos\Omega t - 2\beta\omega_0\dot{u} - \gamma u^3\right)$ — (1)

Assumed sol: (Multiple Scale) $u(t,\varepsilon) = u_0(T_0,T_1) + \varepsilon u_1(T_0,T_1) + O(\varepsilon^2)$

ε^0: $u_0 = A(T_1)e^{i\omega_0 T_0} + \bar{A}e^{-i\omega_0 T_0}$

$u_1 = \frac{\gamma}{8\omega_0^2}A^3 e^{i3\omega_0 T_0} + cc$

Finding Unknown $A(T_1) = \frac{1}{2}a e^{i\phi}$; $a(T_1)$ & $\phi(T_1) \in \mathbb{R}$

$\Omega = \omega_0 + \varepsilon\sigma$

Modulation equation: (Autonomous form)

(2) —
$$a' = -\beta\omega_0 a + \frac{q}{2\omega_0}\sin\psi$$
$$\psi' = \sigma - \frac{3\gamma}{8\omega_0}a^2 + \frac{q}{2\omega_0 a}\cos\psi$$

Stationary solution: $a' = 0$; $\psi' = 0$

$\Rightarrow \left[(2\beta\omega_0^2)^2 + \left(2\sigma\omega_0 - \frac{3}{4}\gamma\tilde{a}^2\right)^2\right]\tilde{a}^2 = q^2$, $\tan\psi = \dfrac{-\beta\omega_0}{\sigma - \frac{3\gamma}{8\omega_0}a^2}$ we get after solving

Rewriting it in terms of Ω, we get

(3) — $$\frac{\Omega}{\omega_0} = 1 + \frac{3(\varepsilon\gamma)}{8\omega_0^2}\tilde{a}^2 \pm \sqrt{\frac{(\varepsilon q)^2}{(2\omega_0^2\tilde{a})^2} - (\varepsilon\beta)^2}$$

∴ you can assume $\varepsilon = 1$ here.

Eigen Values at Stationary soln.: $\lambda = -\beta\omega_0 \pm \sqrt{-\left(\sigma - \frac{3\gamma}{8\omega_0}\tilde{a}^2\right)\left(\sigma - \frac{9\gamma}{8\omega_0}\tilde{a}^2\right)}$

Instability exist if

$$\left(\sigma - \frac{3\gamma}{8\omega_0}\tilde{a}^2\right)\left(\sigma - \frac{9\gamma\tilde{a}^2}{8\omega_0}\right) + (\beta\omega_0)^2 < 0$$

(4) — or $$\left(\Omega - \omega_0 - \frac{3(\varepsilon\gamma)}{8\omega_0}\tilde{a}^2\right)\left(\Omega - \omega_0 - \frac{9(\varepsilon\gamma)}{8\omega_0}\tilde{a}^2\right) + (\varepsilon\beta\omega_0)^2 < 0$$

Critical values: ~~Solving~~ Substituting the stationary solution in the instability condn, we get critical values of a & Ω.

From above equation, we get

(5) — $$\frac{\Omega}{\omega_0} = \left(1 + \frac{3\gamma\tilde{a}^2}{16\omega_0^2} + \frac{9\gamma\tilde{a}^2}{16\omega_0^2}\right) \pm \sqrt{\left(1 + \frac{3\gamma a^2}{16\omega_0^2} + \frac{9\gamma a^2}{16\omega_0^2}\right)^2 - \left[\left(1 + \frac{3\gamma a^2}{8\omega_0^2}\right)\left(1 + \frac{9\gamma a^2}{8\omega_0^2}\right) + \beta^2\right]}$$

Equating with stationary solution will result critical values.

I) Numerical solution of modulation equations given by eqn. (2) using ODE45 (RK4 method):

a) Write function file and save it as rigid.m

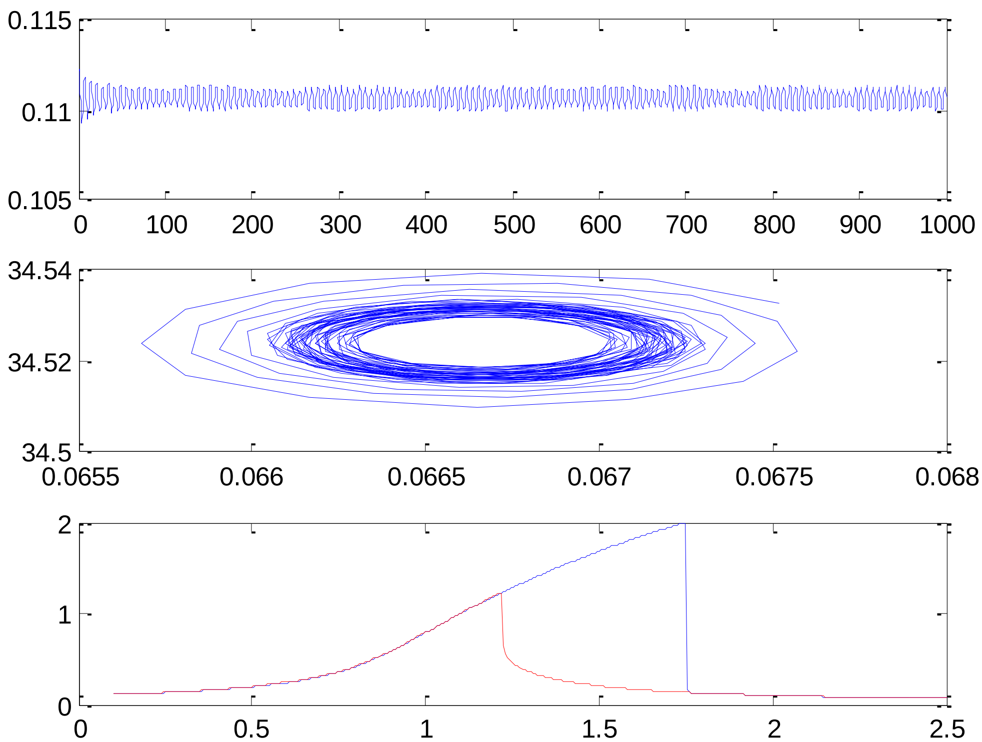
```
function dy=rigid(t,y,w)

global w w0 gamma1 beta1 q
dy=zeros(2,1);


dy(1)=-beta1*w0*y(1)+q/(2*w0)*sin(y(2));
dy(2)=(w-w0)-3*gamma1/(8*w0)*y(1)^2+q/(2*w0*y(1))*cos(y(2));
end
```

b) Write script file and save it as Duffing1.m

```
clc
%close all
clear all
global w w0 gamma1 beta1 q
gamma1=0.5;
beta1=0.05; q=0.2;
w0=1;
np=400;
ww=linspace(.1,2.5,np);
ww1=linspace(2.5,.1,np);
yy=[];
yy1=[];
Y0=[0.1 0.1];
% options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);
% Rp=size()
 for i=1:1:length(ww)
     w=ww(i)
 [T,Y] = ode45(@rigid,[0 200],Y0);

 nn=length(Y(:,1));
 ymax=max(Y(nn-round(nn/4):nn,1));
 ymax1=max(Y(nn-round(nn/4):nn,2));
 Y0=[ymax ymax1];

 yy=[yy;ymax];

 subplot(3,1,1)
 plot(T,Y(:,1))
 subplot(3,1,2)
 plot(Y(:,1),Y(:,2))


 end

subplot(3,1,3)
plot(ww,yy,'r')
 hold on
```

```
%Backsweeping

ymax1=max(Y(nn-round(nn/4):nn,1));
ymax2=max(Y(nn-round(nn/4):nn,2));
for j=1:1:length(ww1)
w=ww1(j)
[T,Y1] = ode45(@rigid,[0 1000],[ymax1 ymax2]);
nn1=length(Y1(:,1));
ymax1=max(Y1(nn1-round(nn1/4):nn1,1));
ymax2=max(Y1(nn1-round(nn1/4):nn1,2));
yy1=[yy1;ymax1];
 subplot(3,1,1)
 plot(T,Y1(:,1))
 subplot(3,1,2)
 plot(Y(:,1),Y(:,2))
end

subplot(3,1,3)
plot(ww1,yy1, 'b')
```

(C) Running Duffing1.m with gamma1=0.5; beta1=0.05; q=0.2; w0=1; will give the following graph (Transient solution, Phase plot, Frequency response)



II) Analytical solution of Duffing equation

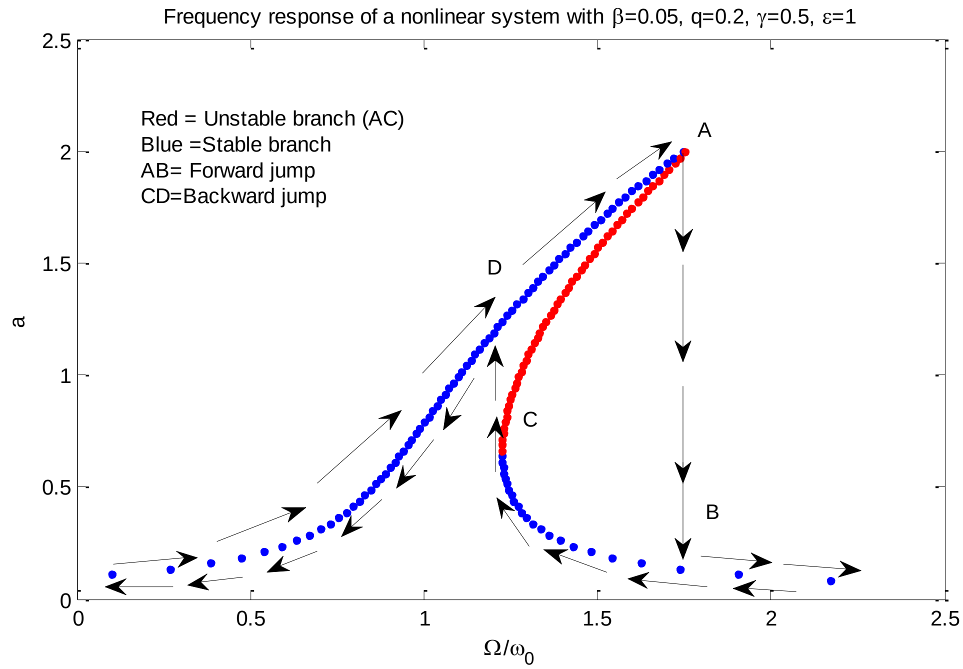
```
%% Matlab code to find the frequency response of Single Duffing oscillator

clear all;
clc;

% Given values

gamma = 1;    % Nonlinear Parameter
q = 0.2;      % Forcing amplitude
beta = 0.05; % linear damping
w0=1;         % Primary resonance

a=linspace(0.01,2.5,100);    % Range of a

%Finding the values of excitation frequency Omega/w0= F and G.
% And corresponding eigen values
%

for ii=1:1:length(a)

    F(ii)=(1+3*gamma/(8*w0^2)*a(ii)^2+sqrt((q/(2*w0^2*a(ii)))^2-beta^2));

    lamF1=sqrt(-(F(ii)-w0-3*gamma/(8*w0)*a(ii)^2)*(F(ii)-w0-
                              9*gamma/(8*w0)*a(ii)^2))-(beta*w0)
    lamF2=-sqrt(-(F(ii)-w0-3*gamma/(8*w0)*a(ii)^2)*(F(ii)-w0-
                              9*gamma/(8*w0)*a(ii)^2))-(beta*w0)

    G(ii)=(1+3*gamma/(8*w0^2)*a(ii)^2-sqrt((q/(2*w0^2*a(ii)))^2-beta^2));

    lamG1=sqrt(-(G(ii)-w0-3*gamma/(8*w0)*a(ii)^2)*(G(ii)-w0-
                              9*gamma/(8*w0)*a(ii)^2))-(beta*w0)
    lamG2=-sqrt(-(G(ii)-w0-3*gamma/(8*w0)*a(ii)^2)*(G(ii)-w0-
                              9*gamma/(8*w0)*a(ii)^2))-(beta*w0)


   if lamF1==conj(lamG1)%    % To terminate the loop

      break

   else

        if gamma>0   % For spring hardening effect

           plot(G(ii),a(ii),'b.');
           hold on

          FG=(F(ii)-w0-3*gamma/(8*w0)*a(ii)^2)*(F(ii)-w0-
                        9*gamma/(8*w0)*a(ii)^2)+(beta*w0)^2;

           if FG<0
              plot(F(ii),a(ii),'r.')
```

```
        else
           plot(F(ii),a(ii),'b.')

        end

   else

         plot(F(ii),a(ii),'b.');
         hold on

        GF=(G(ii)-w0-3*gamma/(8*w0)*a(ii)^2)*(G(ii)-w0-
                     9*gamma/(8*w0)*a(ii)^2)+(beta*w0)^2;

     if GF<0

         plot(G(ii),a(ii),'r.')

    else

        plot(G(ii),a(ii),'b.')

    end

   end

  end
end

xlim([0,2.5]);  % Range of x-axis
ylim([0,2.5]);  %Range of y-axis

xlabel('\Omega/\omega_{0}')    % Label of x-axis
ylabel('a')                    % Label of y-axis

%title('Frequency response of a nonlinear system with \beta=0.05, q=0.2,
\gamma=0.5, \epsilon=1')

% title('Effect of forcing paramter q for the system with \beta=0.05,
\gamma=0.5, \epsilon=1')

% title('Effect of damping paramter beta for the system with q=0.2,
\gamma=0.5, \epsilon=1')


%title('Effect of nonlinear paramter \gamma for the system with \beta=0.05,
q=0.2, \epsilon=1')
```

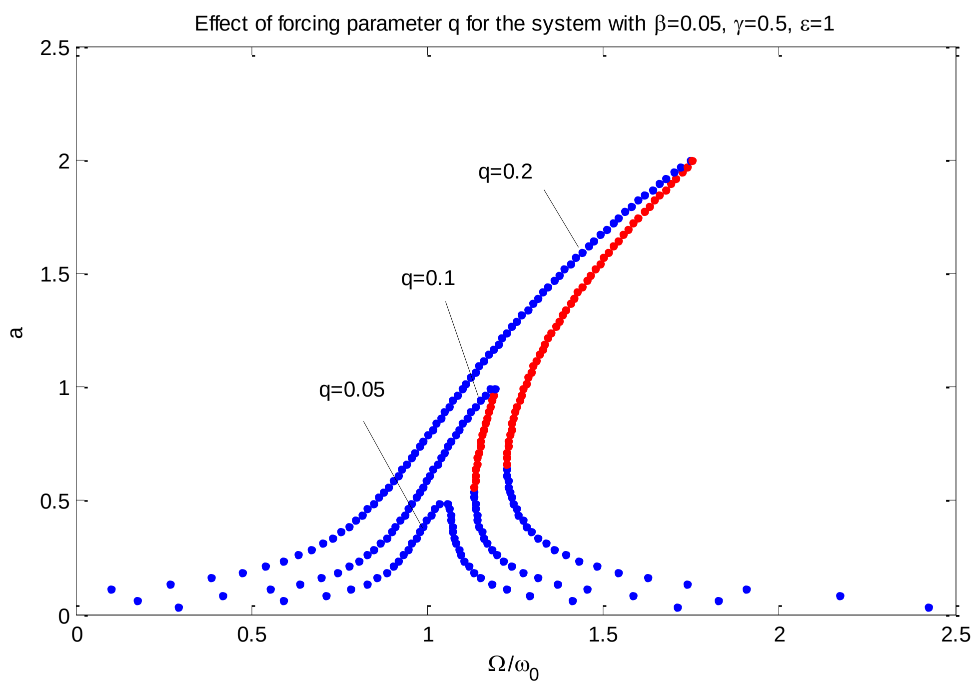
(a) A typical nonlinear response of Duffing equation for a given values of beta, q and gamma.

For this case use the following command in Matlab to write title:

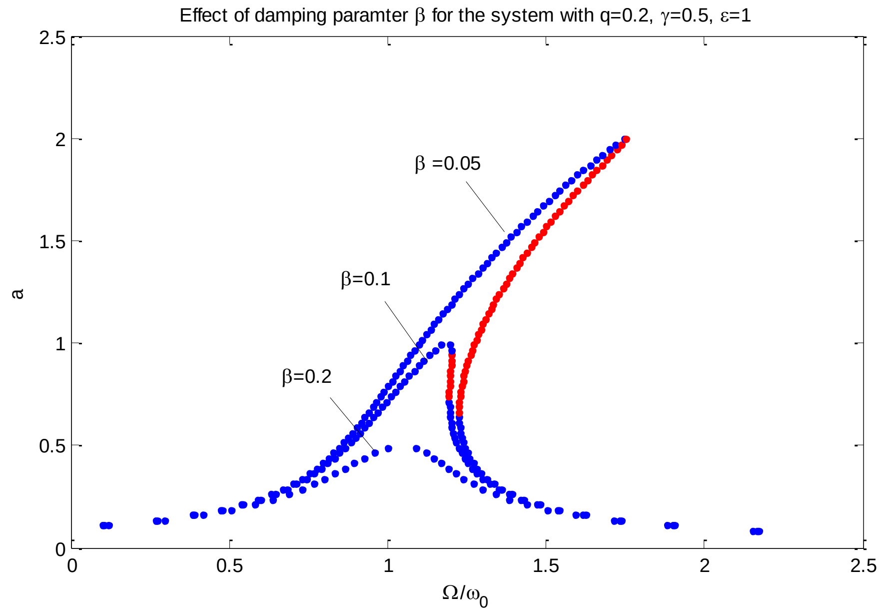
```
title('Frequency response of a nonlinear system with \beta=0.05, q=0.2,
                                        \gamma=0.5, \epsilon=1')
```



(b) Frequency response for different levels of system excitations (q=0.05, 0.10, 0.2) with gamma=0.5 and beta =0.05. (Select the appropriate command for title)



(c) Frequency responses for different levels of viscous damping (beta) with gamma=0.5, q=0.2.



(d) Frequency response for varying levels of nonlinearity (gamma) with q=0.5, beta=0.05.

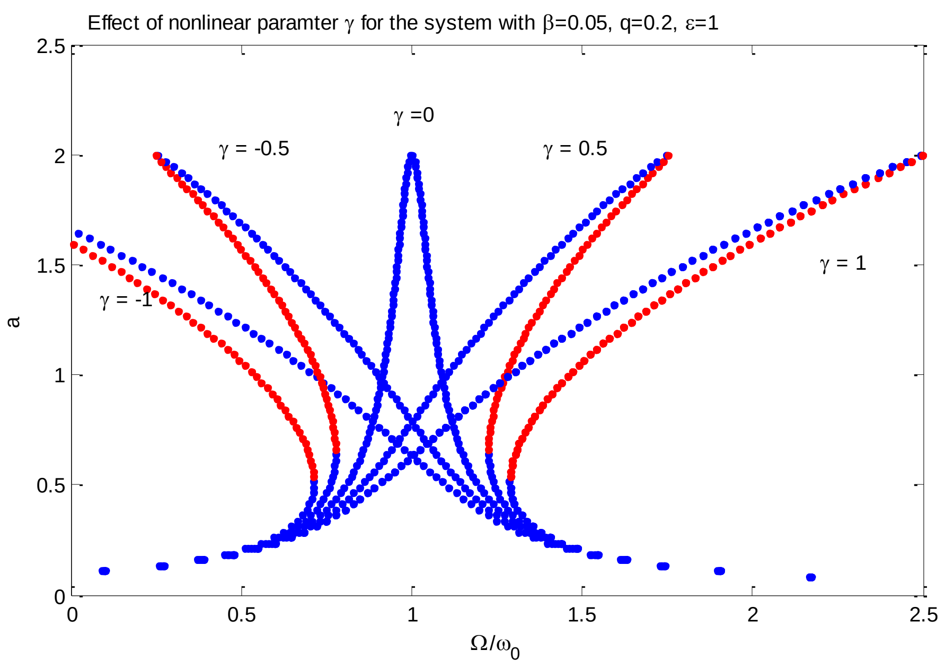


Figure. Frequency response for varying levels of nonlinearity gamma (red …. =Unstable, Blue …. = Stable)

III) The procedure to draw backbone curve (critical values of response function) using the following matlab program:

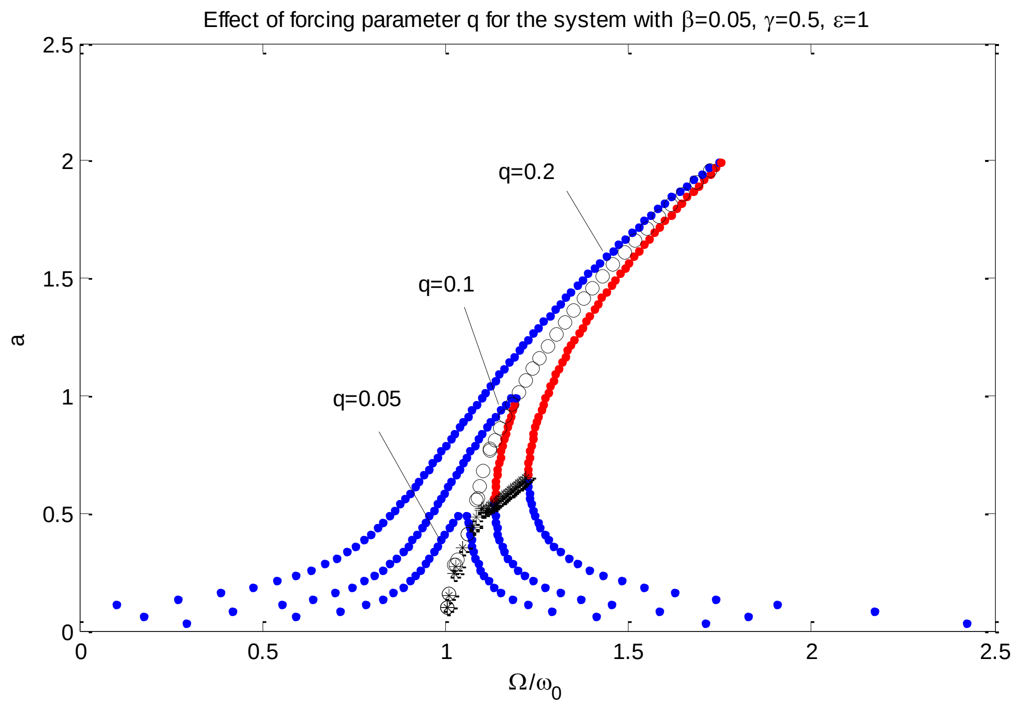
(a) Save this function file as funcAC.m

```
function F = funcAC(a)

global gamma q beta w0;

F(1)=((1+3*gamma/(8*w0^2)*a(1)^2+sqrt((q/(2*w0^2*a(1)))^2-beta^2)))-
((1+3*gamma/(16*w0^2)*a(1)^2+9*gamma/(16*w0^2)*a(1)^2)+sqrt(((1+3*gamma/(16*w
0^2)*a(1)^2+9*gamma/(16*w0^2)*a(1)^2))^2-
((1+3*gamma/(8*w0^2)*a(1)^2)*(1+9*gamma/(8*w0^2)*a(1)^2)+beta^2)));

F(2)=((1+3*gamma/(8*w0^2)*a(2)^2-sqrt((q/(2*w0^2*a(2)))^2-beta^2)))-
((1+3*gamma/(16*w0^2)*a(2)^2+9*gamma/(16*w0^2)*a(2)^2)-
sqrt(((1+3*gamma/(16*w0^2)*a(2)^2+9*gamma/(16*w0^2)*a(2)^2))^2-
((1+3*gamma/(8*w0^2)*a(2)^2)*(1+9*gamma/(8*w0^2)*a(2)^2)+beta^2)));
```

(b) Write script file to draw backbone curve

```
clear all; clc;
format short;
global gamma q beta w0;
gamma = 0.5;

qq=[0.001:0.005:0.2]; %Forcing function

beta = 0.05;
w0=1;

%%%%%%%% initial conditions %%%%%%%%
a0=[0.1,0.1];

for iq=1:1:length(qq)
    q=qq(iq);
options = optimset('TolFun',1e-15);
ac =(fsolve(@(ac) funcAC(ac), a0, options))
a0=[real(ac(1)),real(ac(2))];

OM(iq,1)=((1+3*gamma/(16*w0^2)*ac(1)^2+9*gamma/(16*w0^2)*ac(1)^2)+sqrt(((1+3*
gamma/(16*w0^2)*ac(1)^2+9*gamma/(16*w0^2)*ac(1)^2))^2-
((1+3*gamma/(8*w0^2)*ac(1)^2)*(1+9*gamma/(8*w0^2)*ac(1)^2)+beta^2)));
OM(iq,2)=((1+3*gamma/(16*w0^2)*ac(2)^2+9*gamma/(16*w0^2)*ac(2)^2)-
sqrt(((1+3*gamma/(16*w0^2)*ac(2)^2+9*gamma/(16*w0^2)*ac(2)^2))^2-
((1+3*gamma/(8*w0^2)*ac(2)^2)*(1+9*gamma/(8*w0^2)*ac(2)^2)+beta^2)));
plot(real(OM(iq,2)), real(ac(2)),'ko')
hold on
plot(real(OM(iq,1)), real(ac(1)),'k*')

end
```

c) First draw II(b) and hold on and run III (b) using III(a) as function file, we get backbone curves (critical values of 'a')



IV) Comparison between numerical and analytical solution