EE562 PROJECT #3

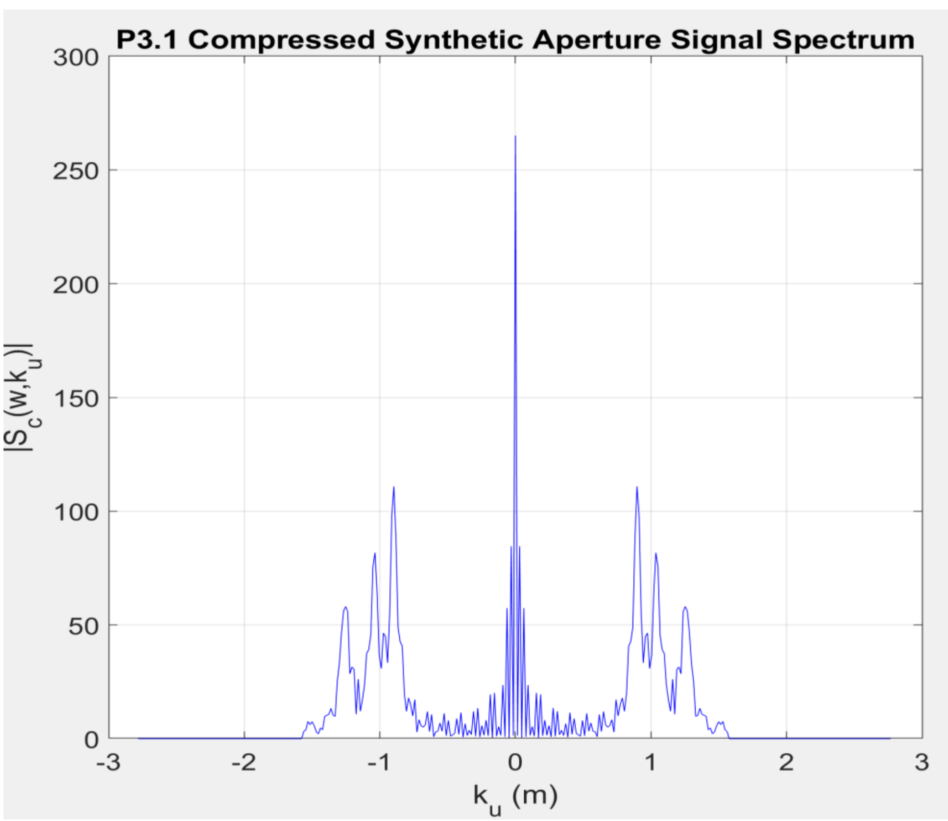
Based on the given problem, Cross range imaging target area is twice the cross range size of the target area ($2Y_0$). Initially the targets are assigned in the given target area. According to the problem, there are 4 targets and the radar is in the location (0,u). As the imaging follows squint mode, the squint angle, squint range and the Doppler frequency shift due to squint is calculated. Then slow time compression is performed to save the PRF (this is done to reduce the hardware limitations of the radar system). Then the synthetic aperture signal s(w,u) is mixed with the exp(-cj*kus*uc) for baseband conversion. The signal defined is transformed into the frequency domain by taking fast Fourier transform of it.

After that the reference signal is defined, where it is given as s0=exp(cj*2*k*sqrt(Xc^2+(Yc-uc).^2)). Then the synthetic aperture signal is mixed with the reference signal for compression. This signal is called as compressed synthetic signal aperture. The important fact is that the synthetic aperture signal s(w,u) is passband and not baseband converted. The compressed signal is converted into its frequency domain by taking Fourier transform.

Matlab program was generated and the results are as follows:

P3.1

Result

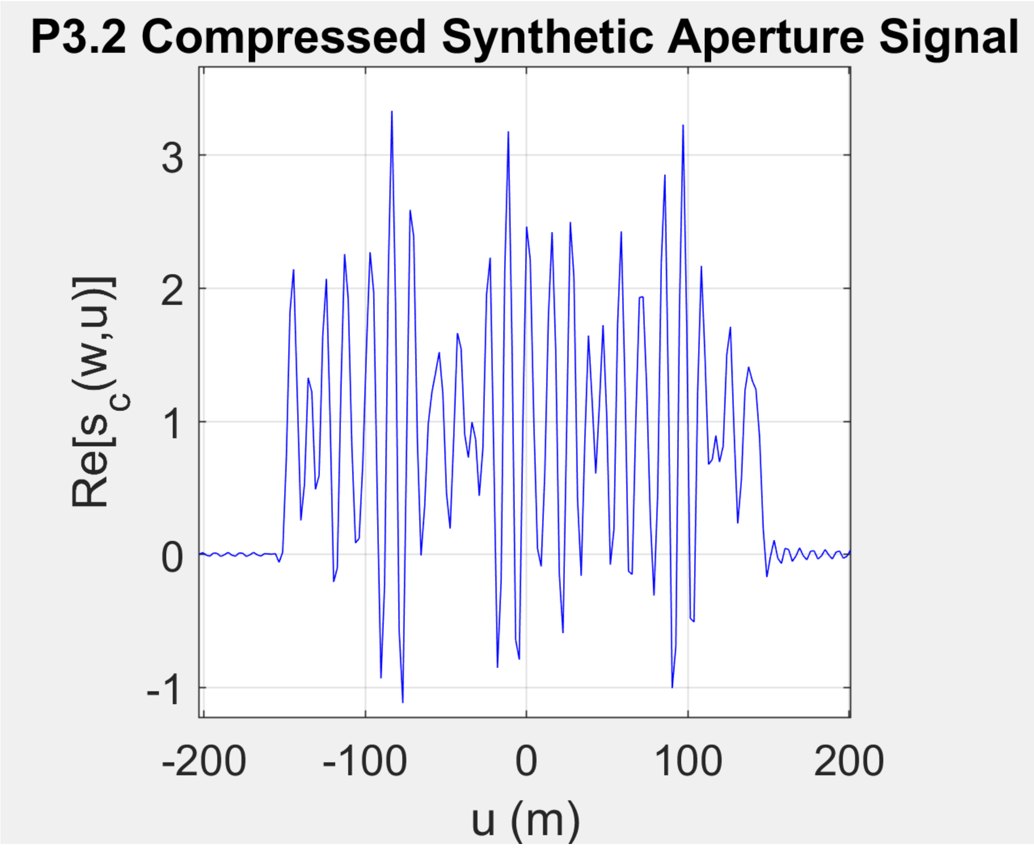


Analysis

In this problem, the compressed signal has to be up-sampled. The compressed signal is converted in frequency domain by using fast fourier transform. For up-sampling, this signal has to be zero padded. For zero padding, the zeros are added in the front and back of the signal. The number of zeros to be added is obtained from the compressed number of samples and actual number of samples on the aperture. Then the signal is plotted with respect to k_u in k_u domain.

P3.2

Result

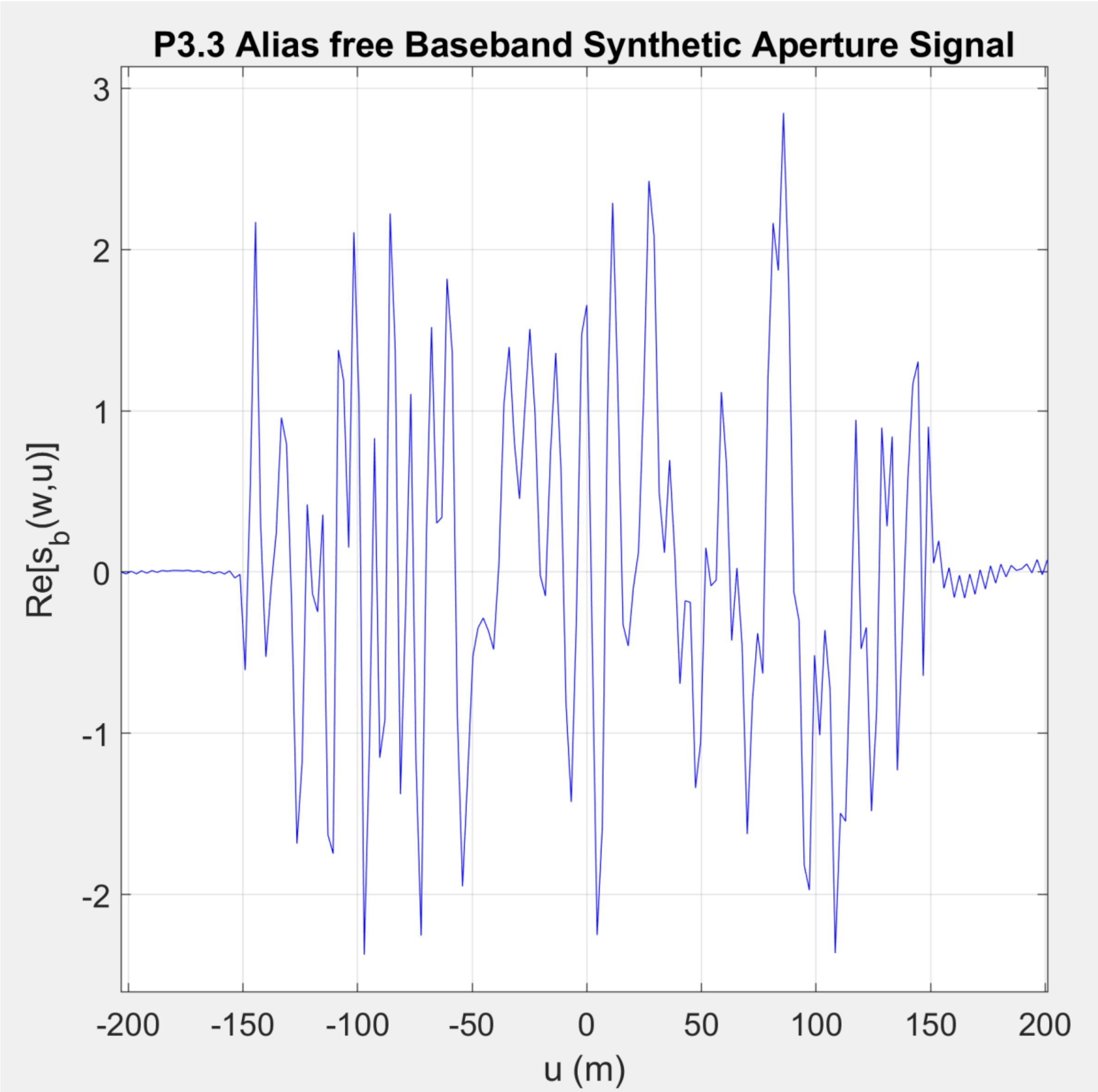


Analysis

The compressed upsampled signal is has to be converted in the u domain in order to plot it with respect to u. So the signal is converted using inverse fast fourier transform. The previous plot of compressed signal before up-sampling was only from [-L,L] i.e. [-150,-150], which is not the same in this case. The signal is present even beyond the limit [-L,L].

P3.3

Result

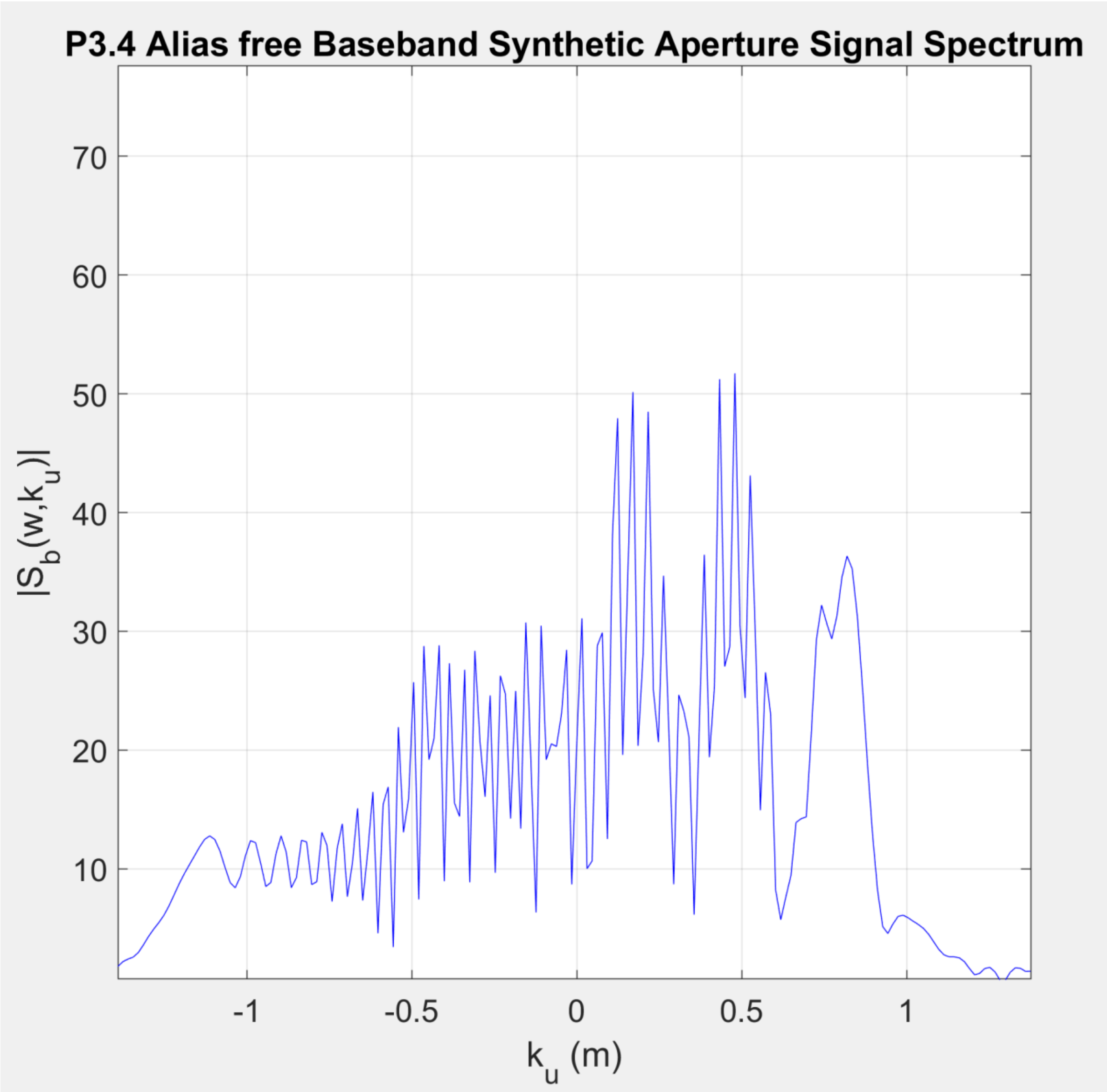


Analysis

After up-sampling of the compressed signal, the signal has to be sent for decompression. In decompression the compressed signal is mixed with the reference signal, this signal is called as the decompressed signal. Then is signal has to be baseband converted which is done by mixing the obtained signal with exp(-j*kus*u). This is plotted with respect to u in the u domain.

P3.4

Result

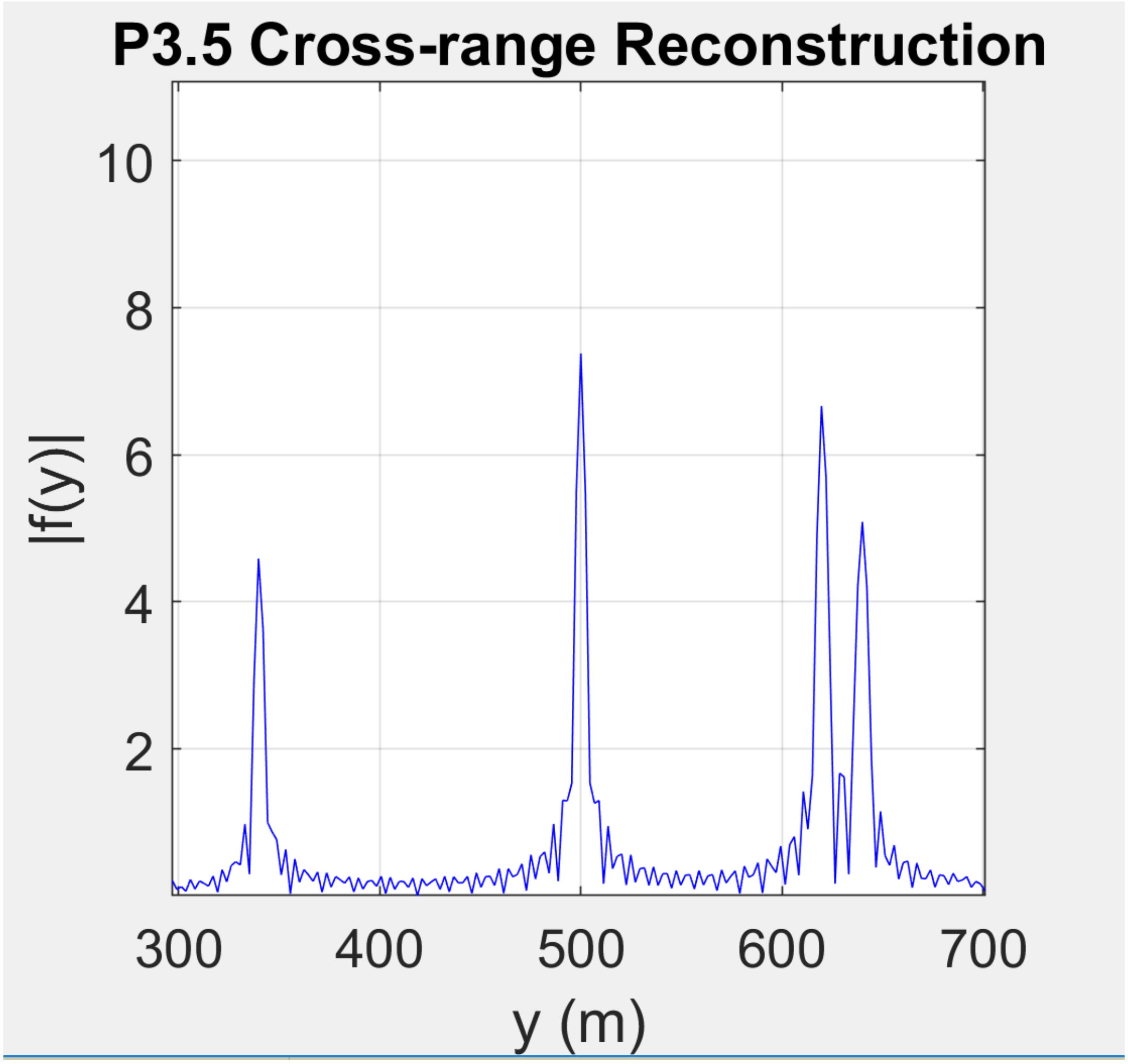


Analysis

Now the signal from the previous problem is converted in k_u domain using the fourier transform. This gives the baseband synthetic aperture signal. The obtained signal is plotted with respect to k_u in the k_u domain.

P3.5

Result



Analysis

Finally matched filtering has to be performed. For that, the reference signal is defined. The reference signal is then baseband converted. The baseband converted reference signal is mixed with the baseband converted decompressed signal, which gives the matched filtered signal. This is plotted with respect to y.

MATLAB CODE:

```
clear all
close all

cj=sqrt(-1);
pi2=2*pi;

c=3e8;
fc=200e6;
lambda=c/fc;
k=pi2/lambda;
Xc=2e3;

L=150;
Y0=200;
Yc=500;

theta_c=atan(Yc/Xc);
Rc=sqrt(Xc^2+Yc^2);
kus=2*k*sin(theta_c);

Xcc=Xc/(cos(theta_c)^2);

du=(Xcc*lambda)/(4*(Y0+L));
duc=(Xcc*lambda)/(4*Y0);

L_min=max(Y0,L);


mc=2*ceil(L_min/duc);
uc=duc*(-mc/2:mc/2-1);
dkuc=pi2/(mc*duc);
kuc=dkuc*(-mc/2:mc/2-1);

dku=dkuc;

m=2*ceil(pi/(du*dku));
du=pi2/(m*dku);
u=du*(-m/2:m/2-1);
ku=dku*(-m/2:m/2-1);

ntarget=4;

yn(1)=0;          fn(1)=1;
yn(2)=.7*Y0;      fn(2)=0.8;
yn(3)=.6*Y0;      fn(3)=1;
```

```
yn(4)=-0.8*Y0;      fn(4)=0.6;

s=zeros(1,mc);
for i=1:ntarget;
    dis=sqrt(Xc^2+(Yc+yn(i)-uc).^2);
    s=s+fn(i)*exp(-cj*2*k*dis).*(abs(uc) <= L);
    s1=s;
end;

s=s.*exp(-cj*kus*uc);

fs=fty(s);

sc=s1.*exp(cj*2*k*sqrt(Xc^2+(Yc-uc).^2));

y=(kuc)*Rc/(2*k*cos(theta_c))+Yc;


sc = [sc,sc(mc:-1:1)];
fsc = fty(sc);

mz= m-mc;
fsc = m/mc*[zeros(1,mz),fsc,zeros(1,mz)];
dku=dku*(-m:m-1);

sc=ifty(fsc);
sc=sc(:,1:m);

figure(1)
plot(dku,abs(fsc),'b-'); grid on;
xlabel('k_u (m)')
ylabel('|S_c(w,k_u)|');
title('P3.1 Compressed Synthetic Aperture Signal Spectrum');
axis('square');
set(gca,'fontsize',16);


figure(2)
plot(u,real(sc),'b-'); grid on;
xlabel('u (m)')
ylabel('Re[s_c(w,u)]');
title('P3.2 Compressed Synthetic Aperture Signal');
axis([u(1) u(m) 1.1*min(real(sc)) 1.1*max(real(sc))]);
axis('square');
set(gca,'fontsize',16);
```

```
s=sc.*exp(-cj*2*k*sqrt(Xc^2+(Yc-u).^2));

sb=s.*exp(-cj*kus*u);

fsb=fty(sb);

figure(3)
plot(u,real(sb),'b-'); grid on;
xlabel('u (m)')
ylabel('Re[s_b(w,u)]');
title('P3.3 Alias free Baseband Synthetic Aperture Signal');
axis([u(1) u(m) 1.1*min(real(sb)) 1.1*max(real(sb))]);
axis('square');
set(gca,'fontsize',16);

figure(4)
plot(ku,abs(fsb),'b-'); grid on;
xlabel('k_u (m)')
ylabel('|S_b(w,k_u)|');
title('P3.4 Alias free Baseband Synthetic Aperture Signal Spectrum');
axis([ku(1) ku(m) 1.1*min(abs(fsb)) 1.5*max(abs(fsb))]);
axis('square');
set(gca,'fontsize',16);

kx=4*k^2-(ku+kus).^2;
kx=sqrt(kx);

fs0b=exp((cj*kx*Xc)+cj*(ku+kus)*Yc);

fsm=fsb.*fs0b;

sm=ifty(fsm);

figure(5)
plot(u+Yc,abs(sm),'b-'); grid on;
xlabel('y (m)')
ylabel('|f(y)|');
title('P3.5 Cross-range Reconstruction');
axis([u(1)+Yc u(m)+Yc 1.1*min(abs(sm)) 1.5*max(abs(sm))]);
axis('square');
set(gca,'fontsize',16);
```