Table of Contents

Binary Axicon EDoF properties

```
F = @(x) fftshift(fft2(ifftshift(x)));
iF = @(x) fftshift(ifft2(ifftshift(x)));
prop = @(f,h) iF(F(f).*h);

%Physical parameters of miniscope (unchagable) in real space
NA0 = 0.45; %NA of the system -> range of ATF (2*NA is range of OTF)
f = 3.3; %mm
dimag = 10; %mm
lambda = 0.00054; %Wavelength in mm
Nyquest = lambda/(4*NA0); %Nyquest requirement for incoherent system
 for comparison
pixel = 0.003; %ideal pixel size in mm
mag = 10; %magnification of system
dx = pixel/mag; %Real space 'pixel' size in mm
N = 400; %desired # of pixels
fov = N*dx; %N = 1000 in this case
[xx,yy] = meshgrid([-N/2:N/2-1]*dx); %Spatial grid for use generating
 phase mask
DoF = (lambda/(2*NA0^2)); %Native depth of field in this system
```

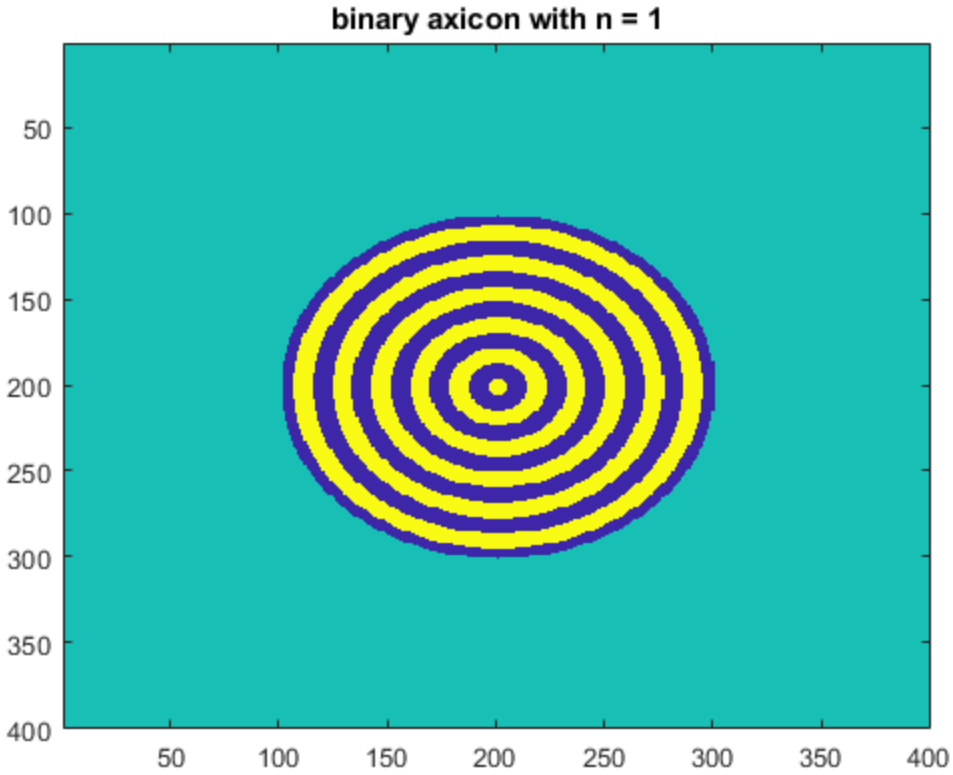
Fourier space

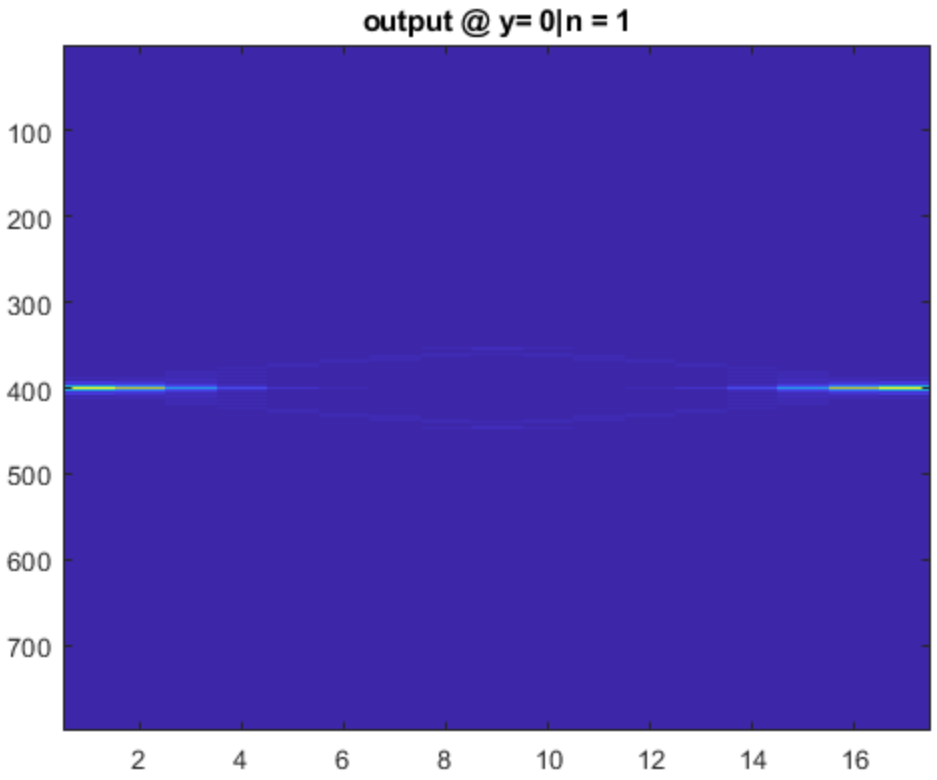
```
du = 1/fov;
[uu,vv] = meshgrid([-N/2:N/2-1]*du); %In Fourier plane, du is defined
 as 1/FOV
NAx = uu*lambda; %converting to NA space (alottable angles) ->
 unitless! easy of scaling and design
NAy = vv*lambda;
NA = sqrt(NAx.^2+NAy.^2); %NA @ any given point
```

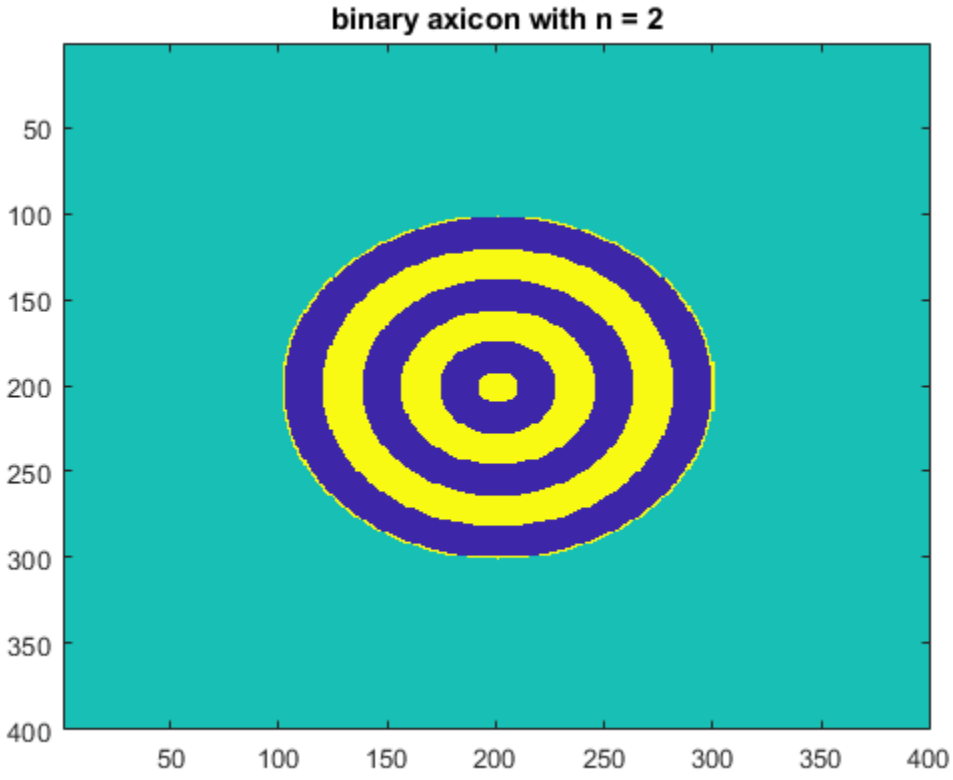
Loop to compare range of defocus of Axicon for different parameters

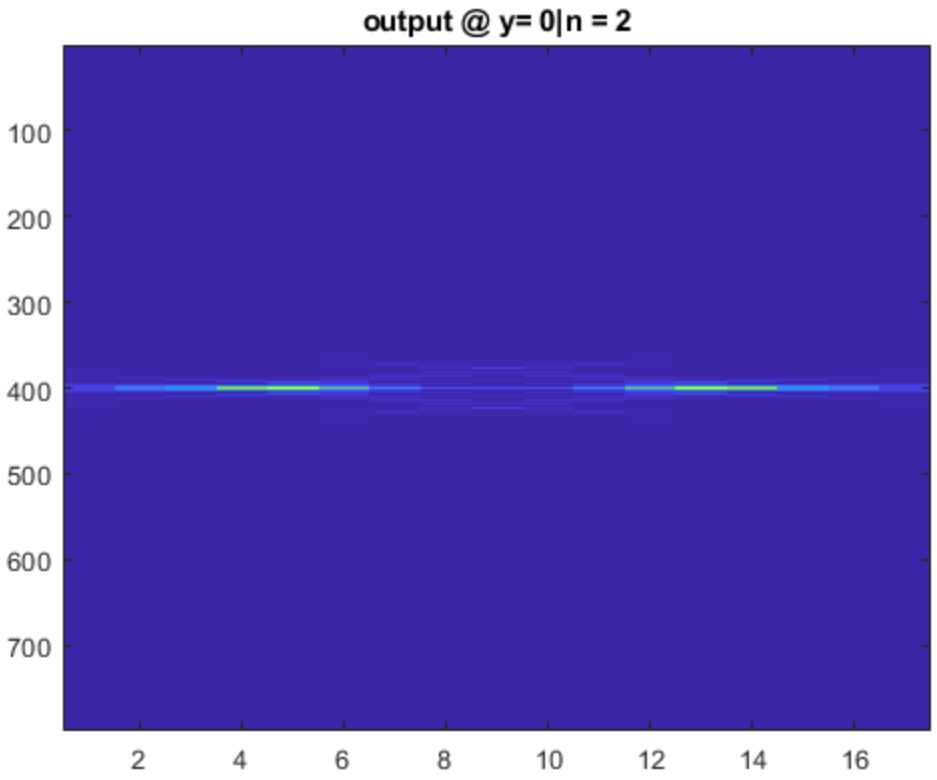
```
dz = [-0.02:0.0025:0.02];
ip_axi = zeros(1,length(dz));
%3D Matrix that will hold 2D slices representing PSF at certain output
EDoF = zeros(2*N-1,2*N-1,length(dz));

for n = 1:5
    %Generate mask and OTF/MTF
```

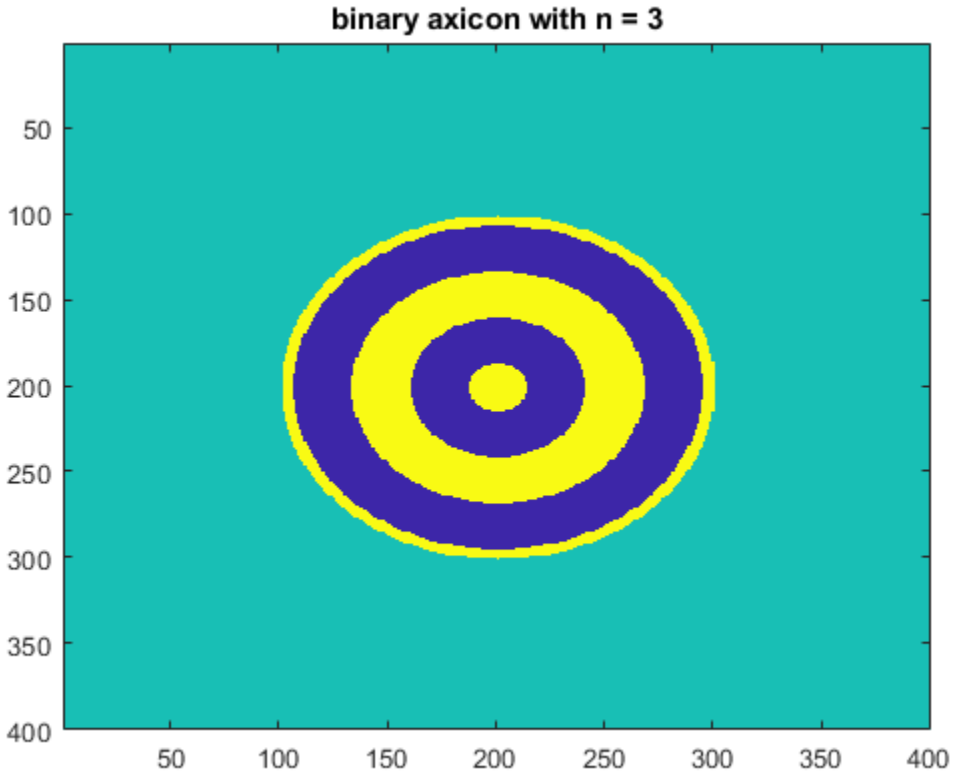
```
    alpha =  NA0^2*f/(n*lambda)*10^-2;
    binaxi = Generate_Binary_Axicon(alpha,NAx,NAy,0);
    %Generate on-axis MTF
    OTF_axi = xcorr2(binaxi,binaxi);
    MTF_axi = abs(OTF_axi);
    figure
    imagesc(binaxi)
    title(['binary axicon with n = ' num2str(n)])
    %Generate defocus
    for k = 1:length(dz)
        defocus = dz(k);%Defocus distance
        defocus_prop =
 exp(1i*pi*lambda*defocus.*(uu.^2+vv.^2)); %Fresnel Kernel
        dOTF_axi = xcorr2(binaxi.*defocus_prop,binaxi.*defocus_prop);
        dMTF_axi = abs(dOTF_axi);
        EDoF(:,:,k) = iF(dOTF_axi);%iPSF = iF{OTF}
    end
    figure
    %Plot PSF for all dz @ y = 0 to determine depth of field created
 by this axicon
    %Center of Output is at N due to autocorrelation length being 2N-1
    imagesc(squeeze(real(EDoF(:,N,:))))%squeeze compresses slice of 3d
 into 2d
    title(['output @ y= 0|n = ' num2str(n)])
    %plot(ip_axi./max(ip_axi))
end
```

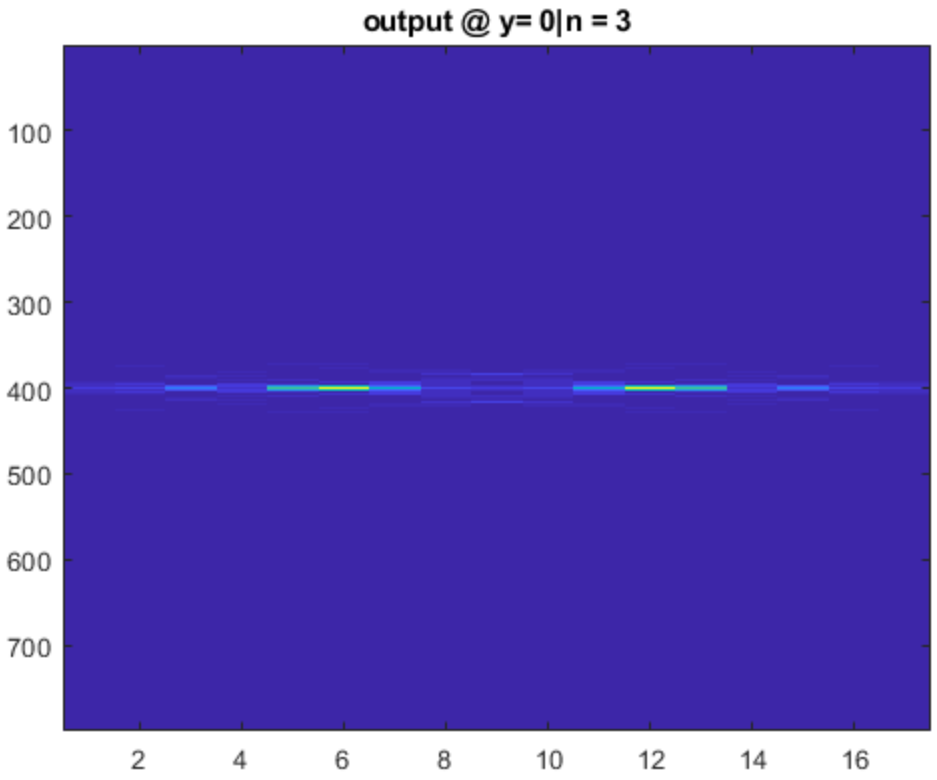


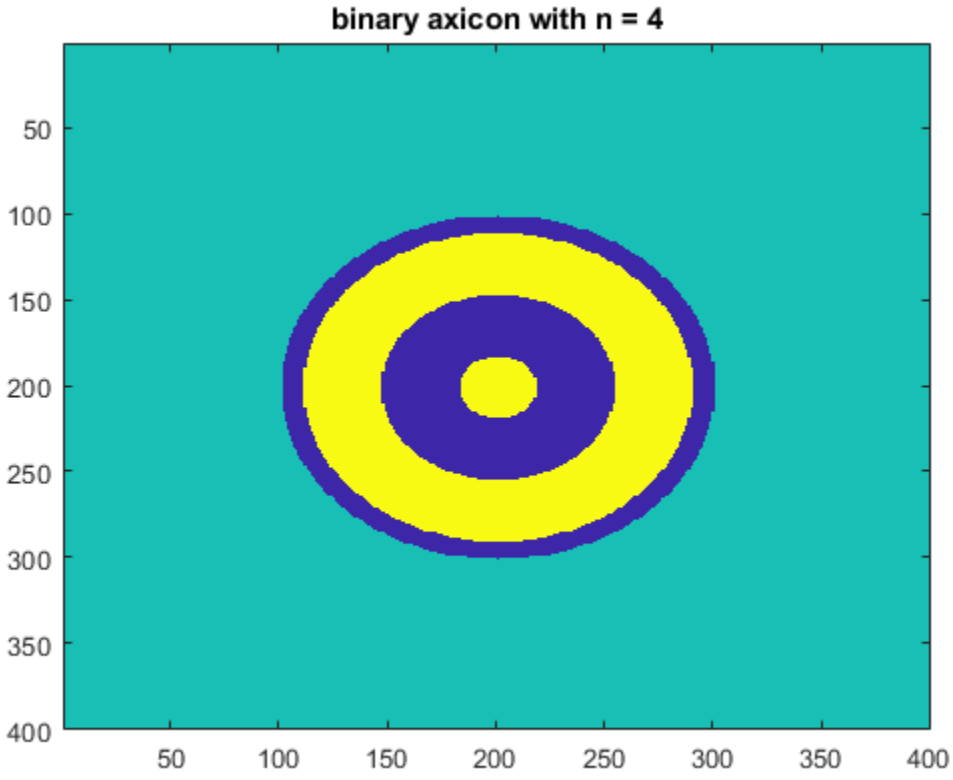

output @ y= 0|n = 1
100
200
300
400
500
600
700
2
4
6
8
10
12
14
16

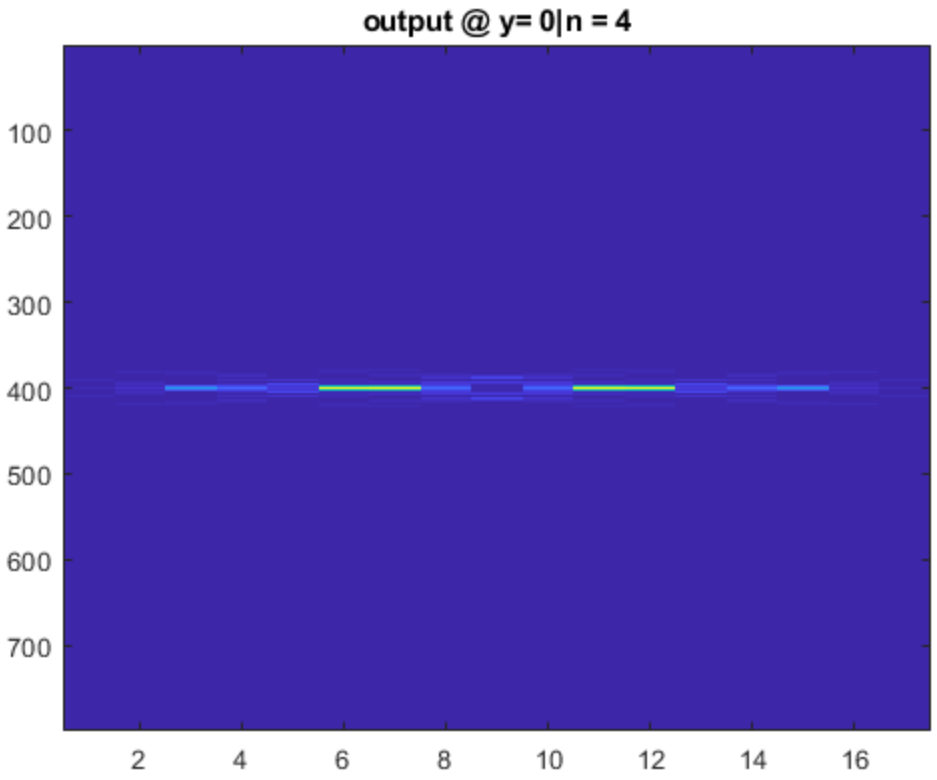

binary axicon with n = 2
50
100
150
200
250
300
350
400
50
100
150
200
250
300
350
400

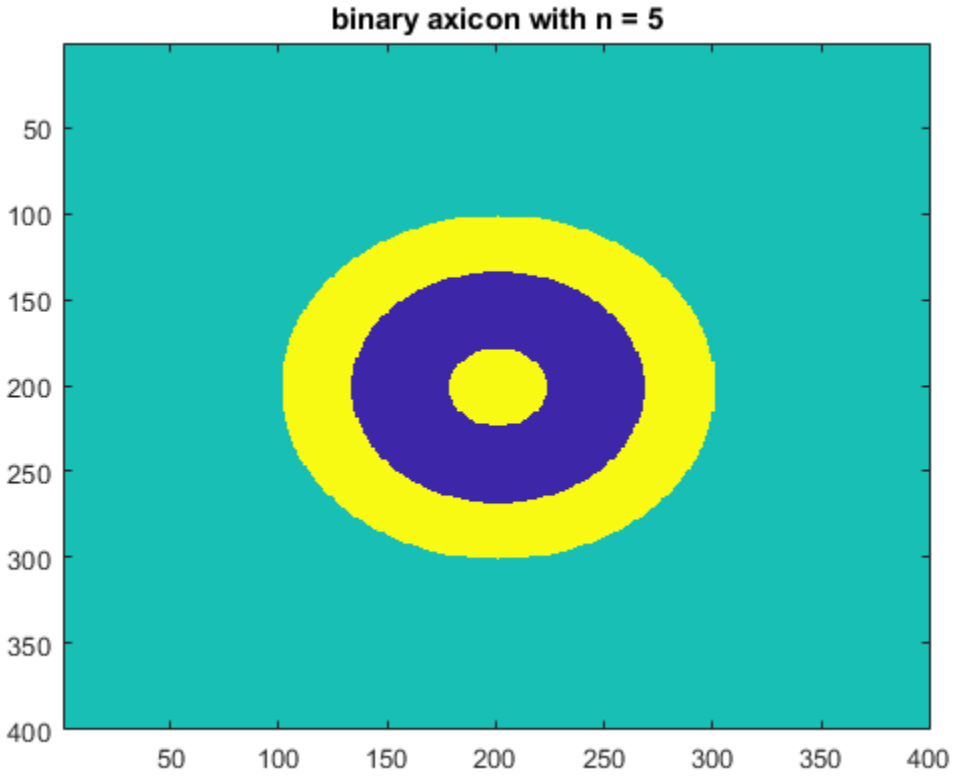

output @ y= 0|n = 2
100
200
300
400
500
600
700
2
4
6
8
10
12
14
16

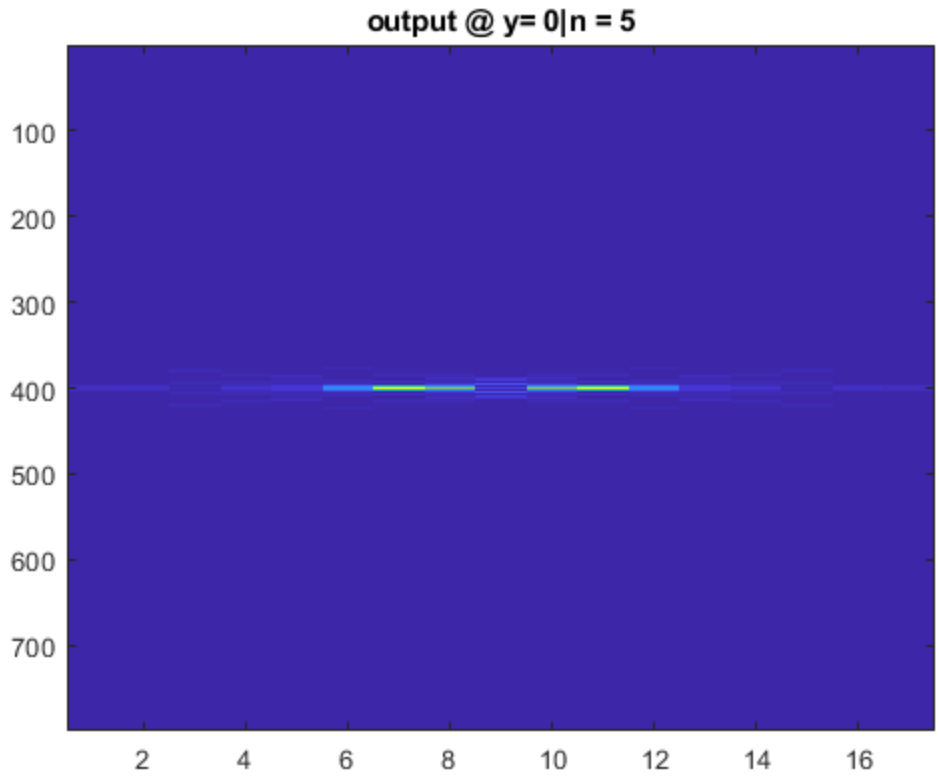

binary axicon with n = 3
50
100
150
200
250
300
350
400
50
100
150
200
250
300
350
400


output @ y= 0|n = 3
100
200
300
400
500
600
700
2
4
6
8
10
12
14
16


binary axicon with n = 4
50
100
150
200
250
300
350
400
50
100
150
200
250
300
350
400


output @ y= 0|n = 4
100
200
300
400
500
600
700
2
4
6
8
10
12
14
16


binary axicon with n = 5
50
100
150
200
250
300
350
400
50
100
150
200
250
300
350
400



Published with MATLAB® R2017b