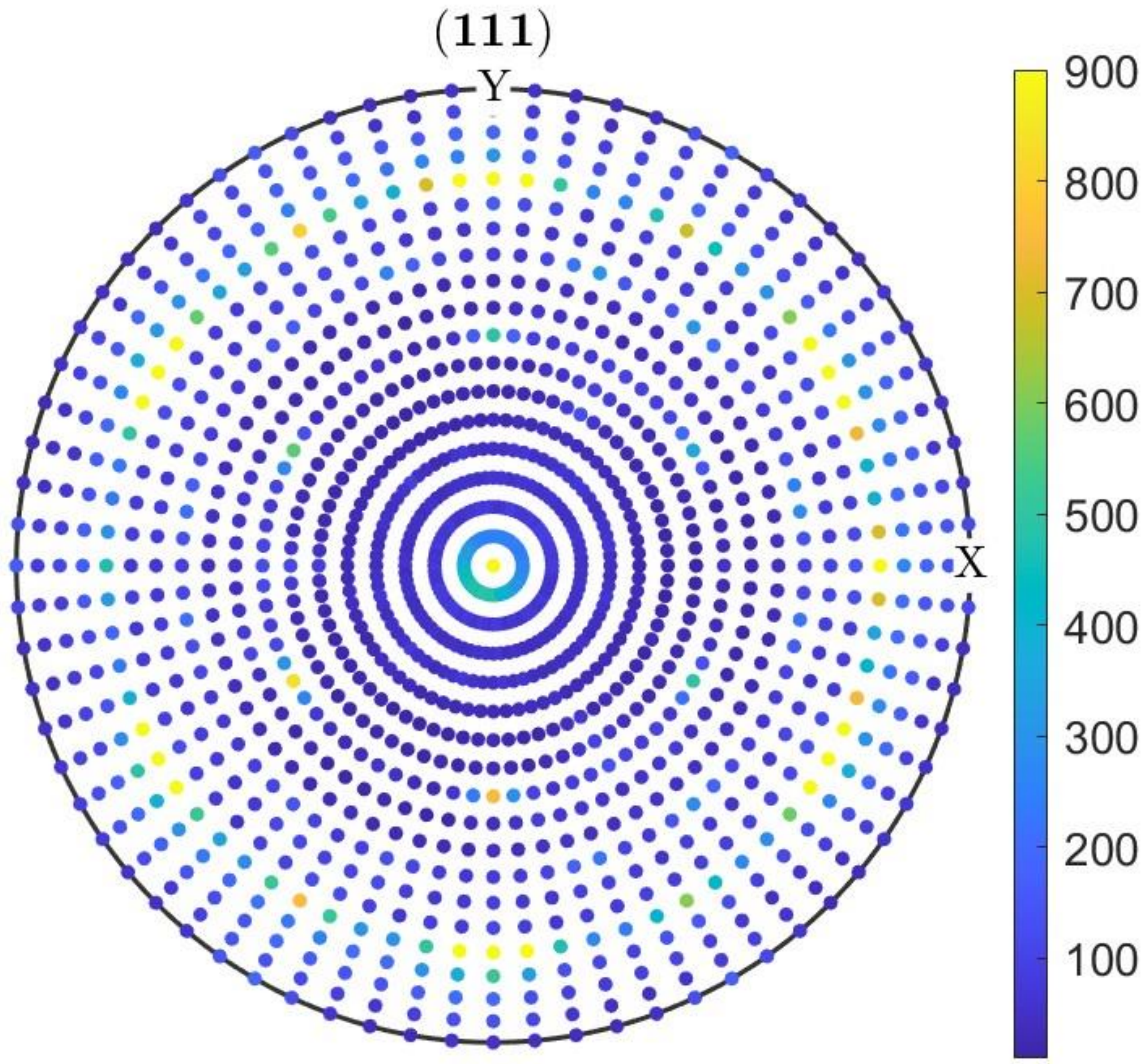
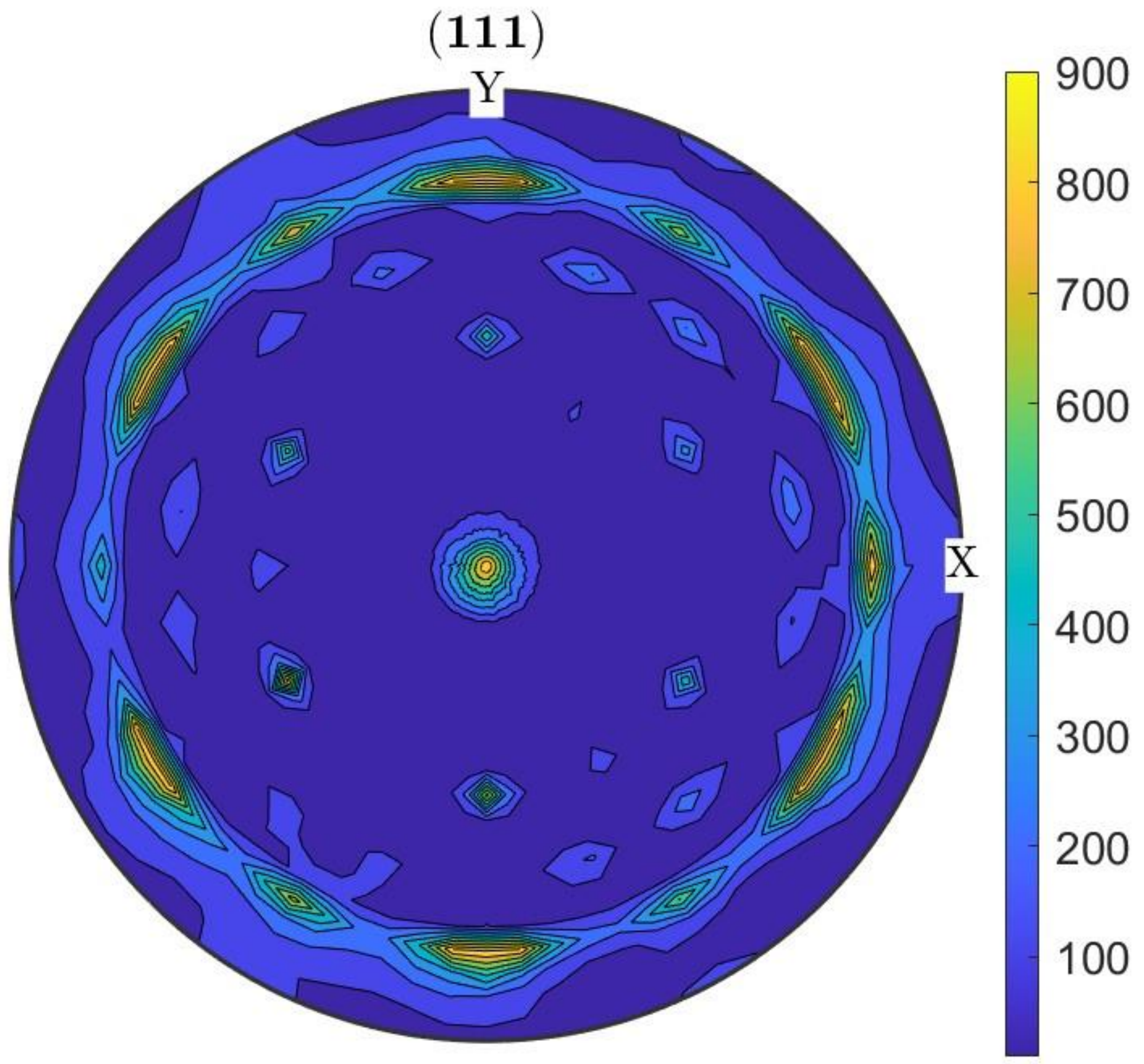
My code + data (slides 1-3)

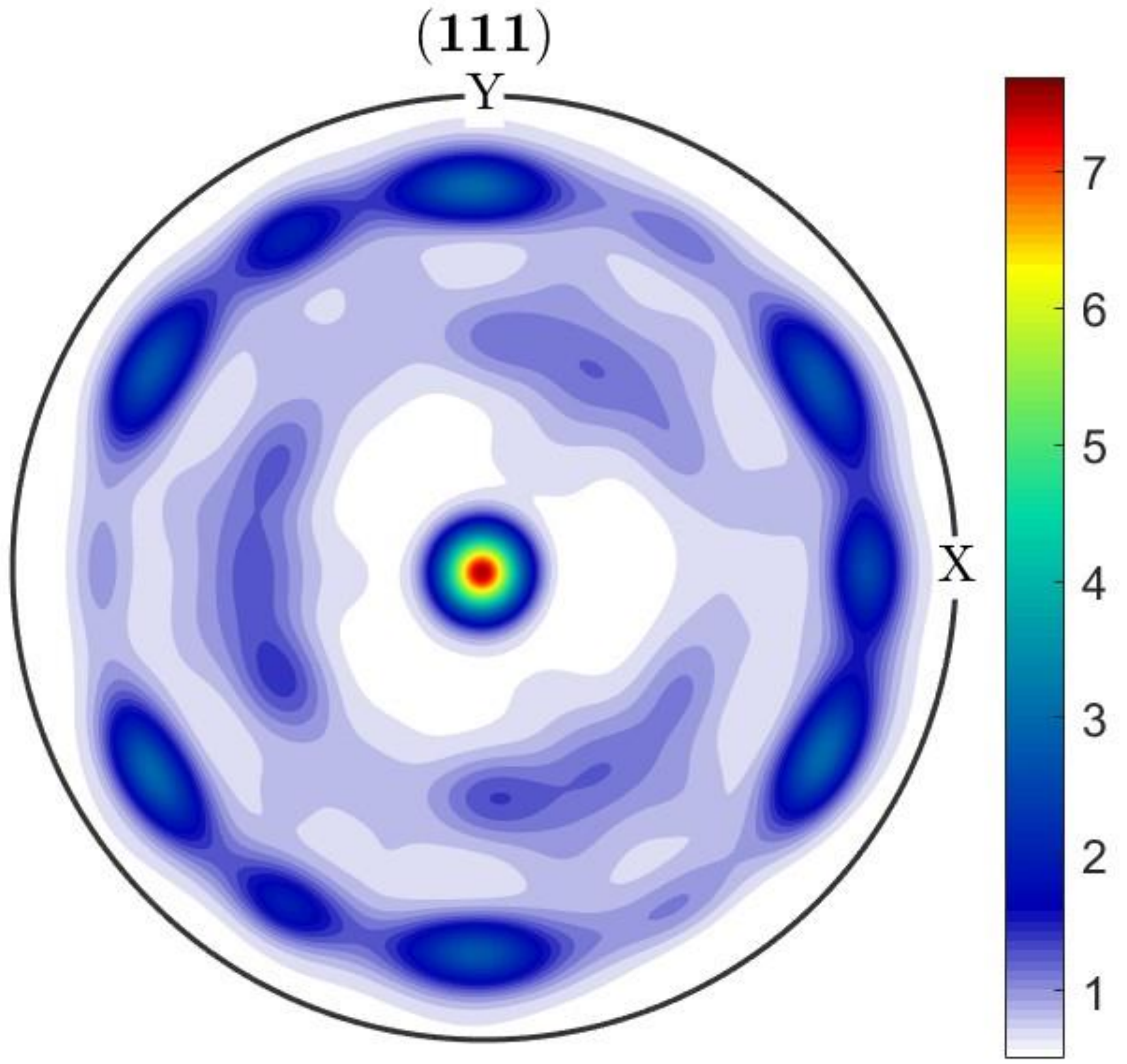
```
%% Import Script for PoleFigure Data
%
% This script was automatically created by the import wizard. You should
% run the whoole script or parts of it in order to import your data. There
% is no problem in making any changes to this script.


%% Specify Crystal and Specimen Symmetries


% crystal symmetry
CS = crystalSymmetry('m-3m', [1 1 1]);


% specimen symmetry
SS = specimenSymmetry('1');


% plotting convention
setMTEXpref('xAxisDirection','east');
setMTEXpref('zAxisDirection','outOfPlane');


%% Specify File Names


% path to files
pname = 'C:\Users\alexw\ path \xrd';


% which files to be imported
fname = [pname '\Alex.ras'];


%% Specify Miller Indice


h = { ...
Miller(1,1,1,CS),...
};
```

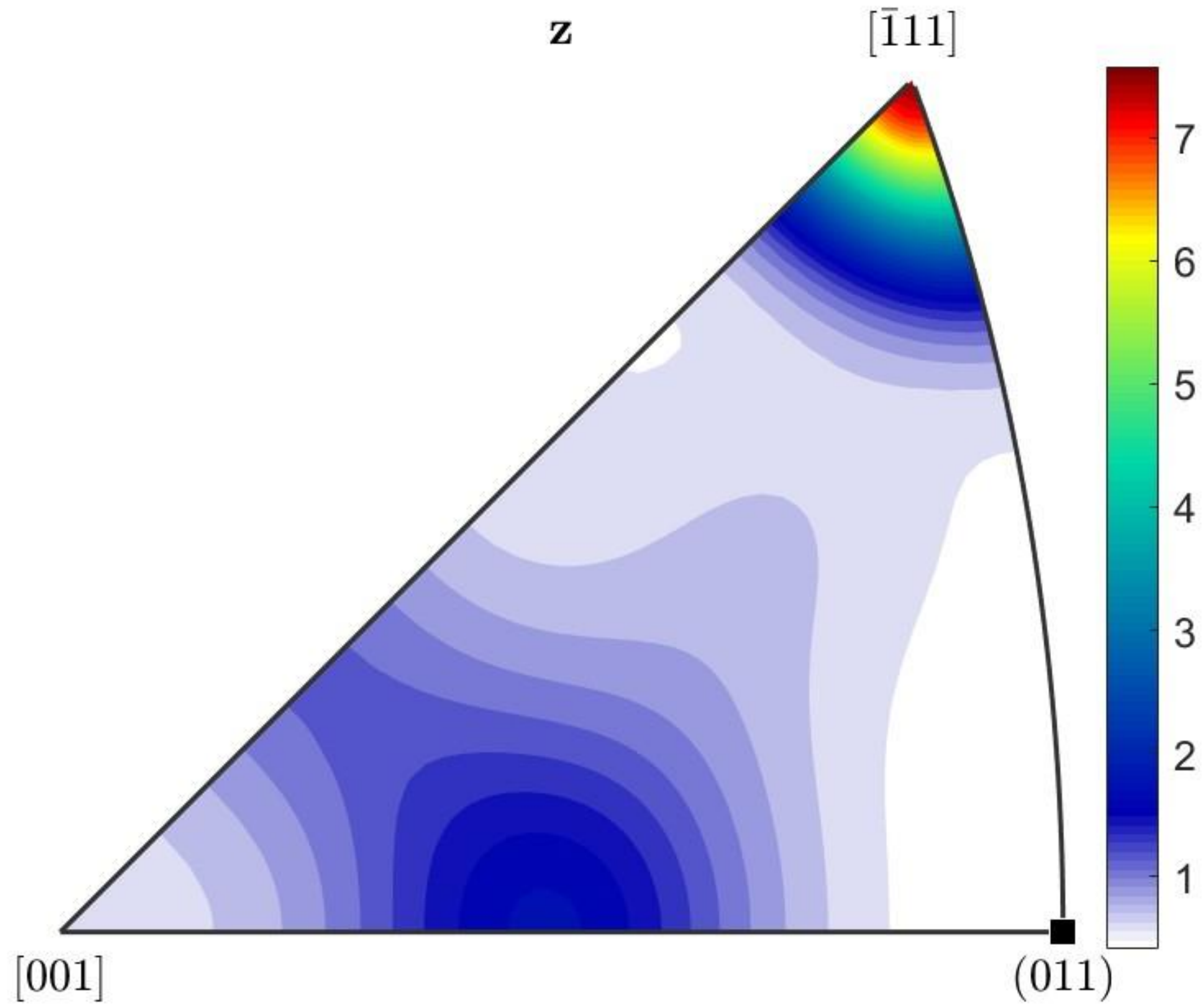
```
%% Import the Data


% create a Pole Figure variable containing the data
pf = PoleFigure.load(fname,h,CS,SS,'interface','xrd');
condition=pf.intensities>900;
pf(condition).intensities=900;
figure(1); plot(pf, 'MarkerSize',4); mtexColorbar;
figure(2); plot(pf, 'contourf'); mtexColorbar; mtexColorMap parula;


%% Orientation Distribution Function (ODF)
% PF only shows a 2D projection of the texture, so some information is
% lost.In order to correctly stock all the information, a 3D-representation
% must be calculated.


%odf=calcODF(pf,'noGhostCorrection');
odf=calcODF(pf);
figure(3); plotPDF(odf,pf.h);
%annotate([xvector,yvector,zvector], 'label', {'X', 'Y', 'Z'}, 'BackgroundColor', 'w');
colorbar;
% This one is not working: figure(4); plotDiff(pf, odf); colorbar;


%% Inverse pole figure
figure(4); plotIPDF(odf, zvector, 'antipodal');
annotate([Miller(1,1,1,CS),Miller(0,1,1,CS),Miller(1,1,0,CS)],'label',{'(111)','(011)','(110)'},'BackgroundColor','w');
colorbar;
```

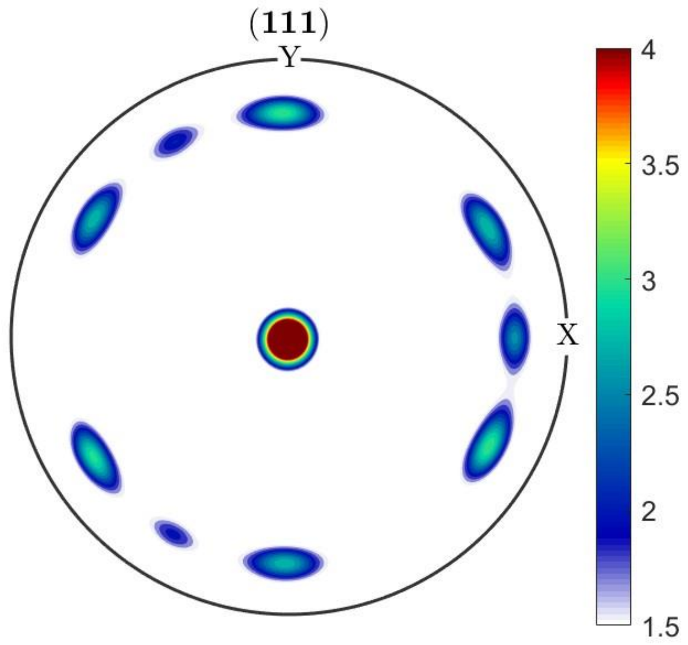
```
>> Cu20240502_111
 0 |  0.81
 1 |  0.51
 2 |  0.47
 3 |  0.46
 4 |  0.45
 5 |  0.44
 6 |  0.43
 I''m going to apply ghost
correction.
 Uniform portion fixed to c0 =
0.14
 0 |  0.84
 1 |  0.52
 2 |  0.48
 3 |  0.47
 4 |  0.46
 5 |  0.45
 6 |  0.44
```

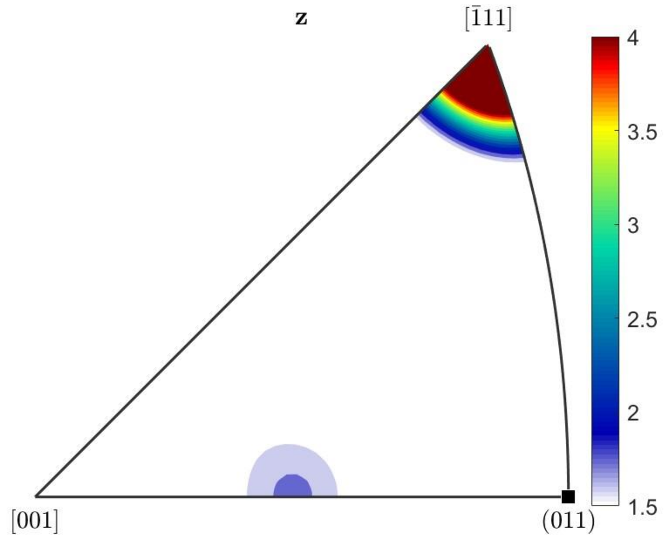








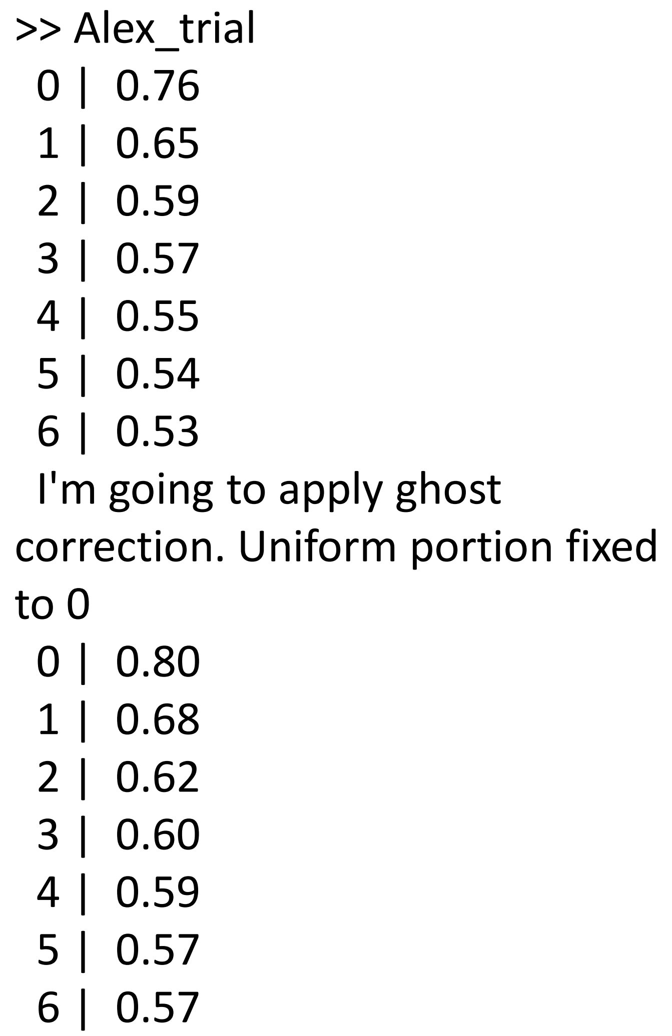




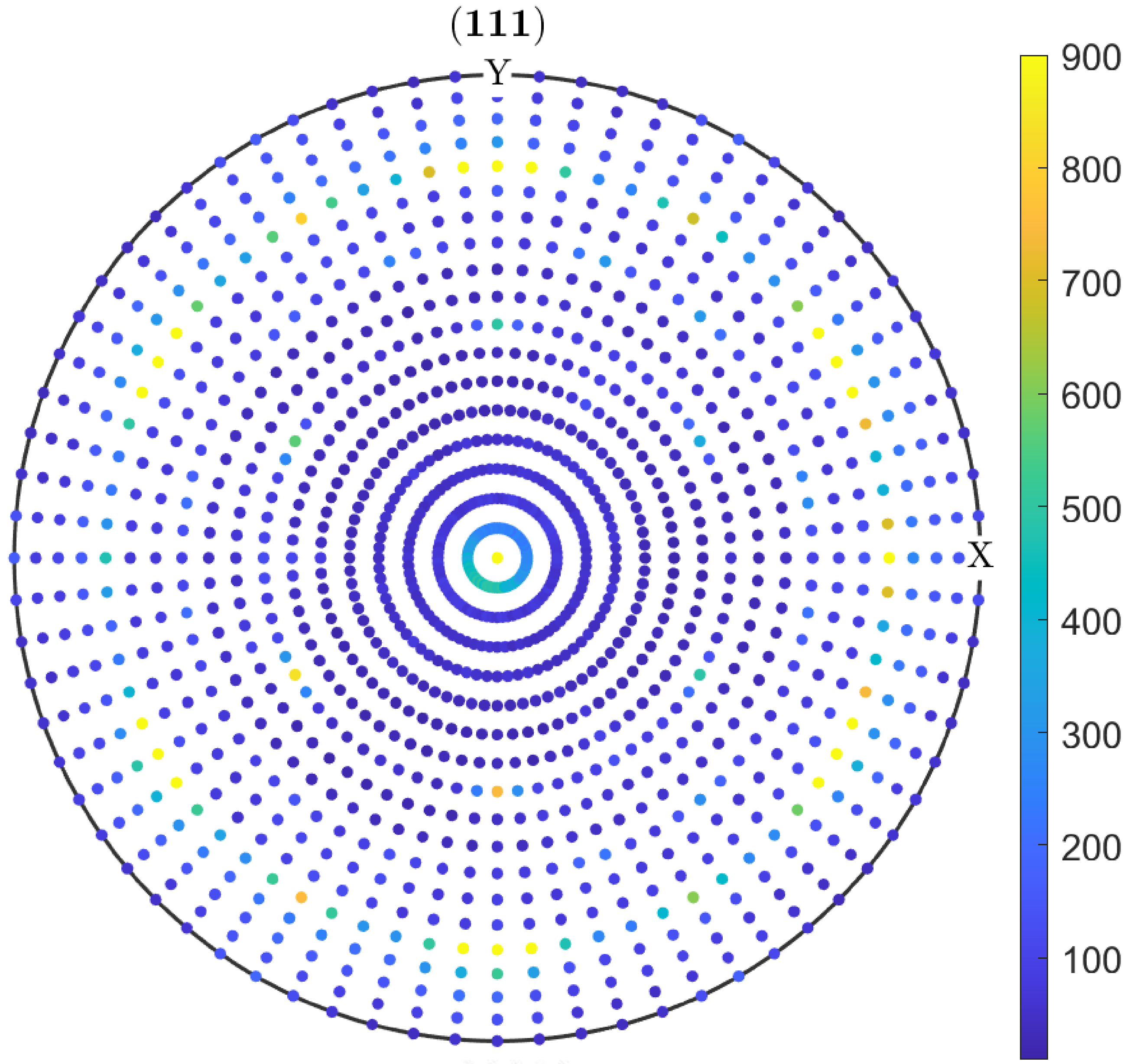
Coworker Data (slides 4-6)

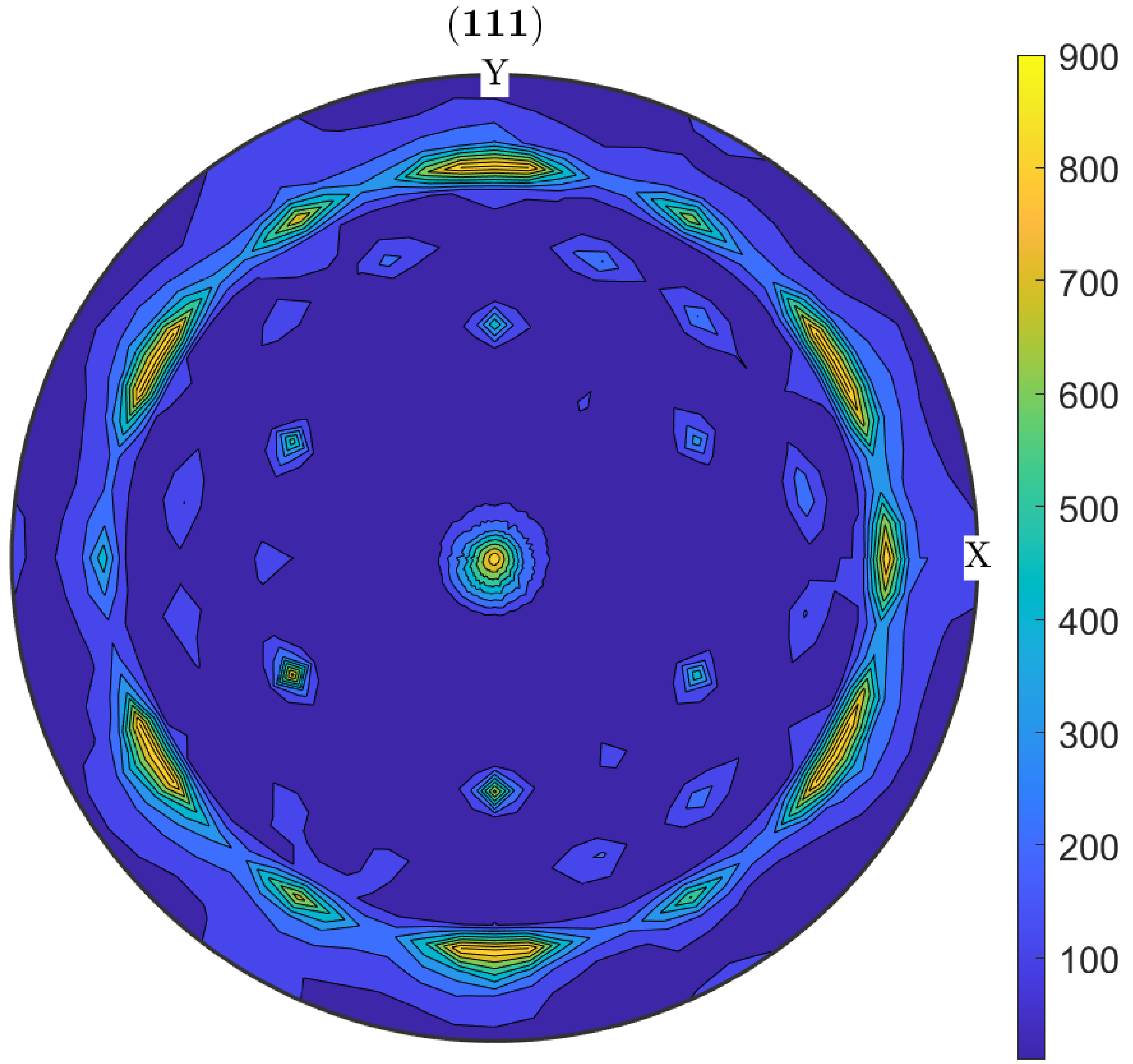
```
%% Import Script for PoleFigure Data
%
% This script was automatically created by the import wizard. You should
% run the whoole script or parts of it in order to import your data. There
% is no problem in making any changes to this script.


%% Specify Crystal and Specimen Symmetries


% crystal symmetry
CS = crystalSymmetry('m-3m', [1 1 1]);


% specimen symmetry
SS = specimenSymmetry('1');


% plotting convention
setMTEXpref('xAxisDirection','east');
setMTEXpref('zAxisDirection','outOfPlane');


%% Specify File Names


% path to files
pname = 'C:\Users\path\MATLAB';


% which files to be imported
fname = [pname '\Alex.ras'];


%% Specify Miller Indice


h = { ...
Miller(1,1,1,CS),...
};
```

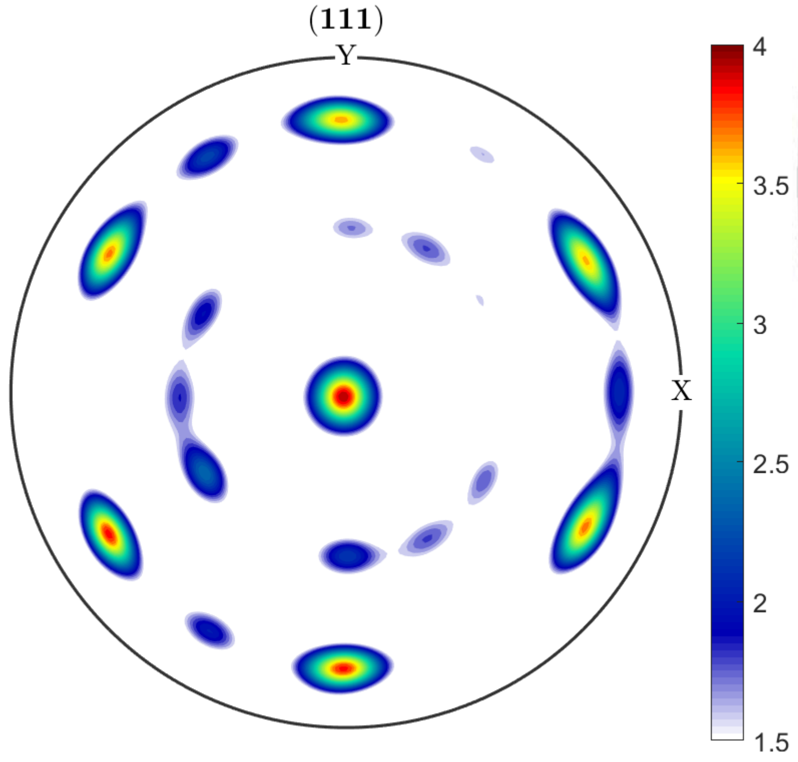
```
%% Import the Data


% create a Pole Figure variable containing the data
pf = PoleFigure.load(fname,h,CS,SS,'interface','xrd');
condition=pf.intensities>900;
pf(condition).intensities=900;
figure(1); plot(pf, 'MarkerSize',4); mtexColorbar;
figure(2); plot(pf, 'contourf'); mtexColorbar; mtexColorMap parula;



%% Orientation Distribution Function (ODF)
% PF only shows a 2D projection of the texture, so some information is
% lost.In order to correctly stock all the information, a 3D-representation
% must be calculated.


%odf=calcODF(pf,'noGhostCorrection');
odf=calcODF(pf);
figure(3); plotPDF(odf,pf.h);
%annotate([xvector,yvector,zvector], 'label', {'X', 'Y', 'Z'}, 'BackgroundColor', 'w');
colorbar;
% This one is not working: figure(4); plotDiff(pf, odf); colorbar;


%% Inverse pole figure
figure(4); plotIPDF(odf, zvector, 'antipodal');
annotate([Miller(1,1,1,CS),Miller(0,1,1,CS),Miller(1,1,0,CS)],'label',{'(111)','(011)','(110)'},'BackgroundColor','w');
colorbar;
```

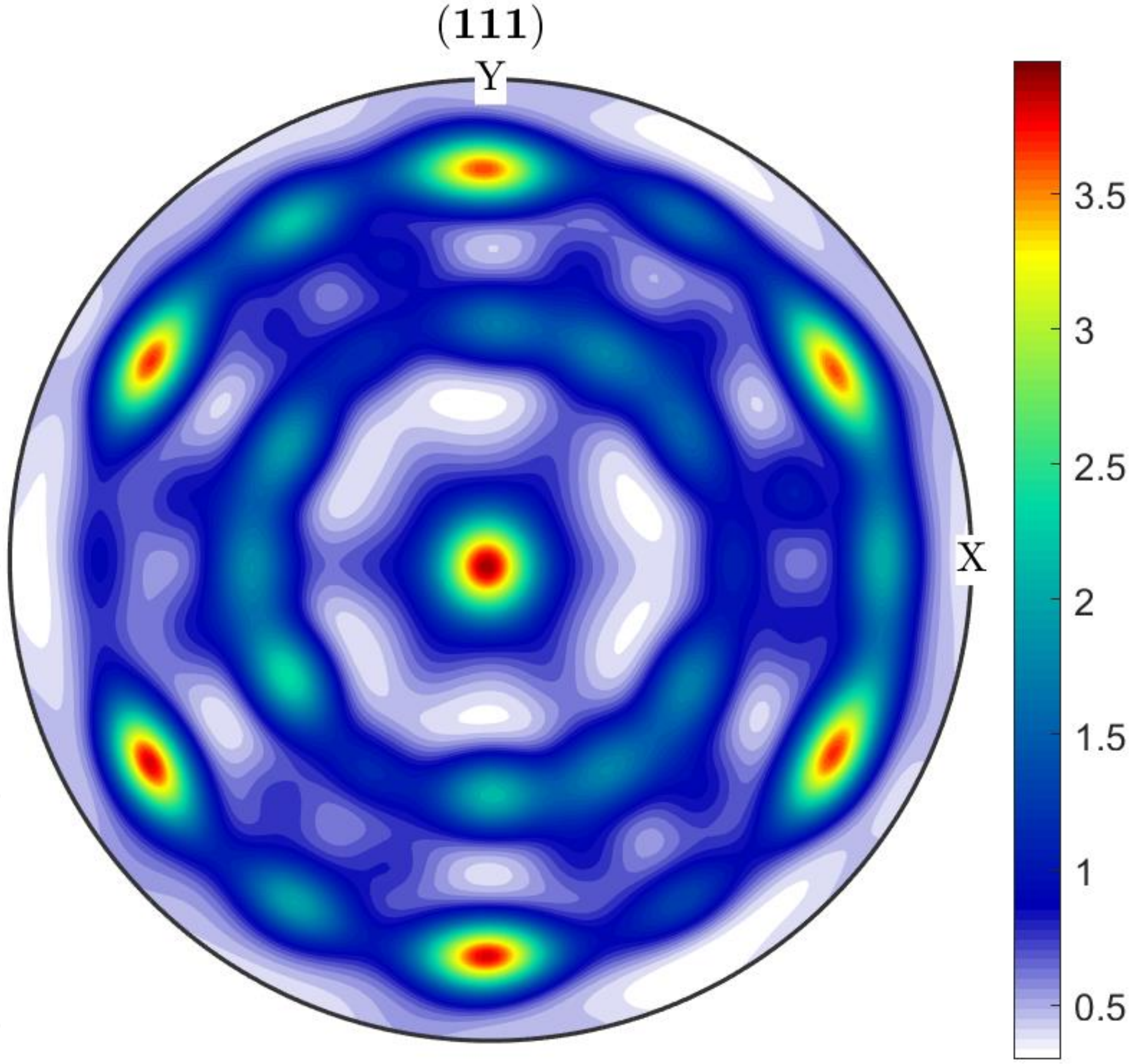


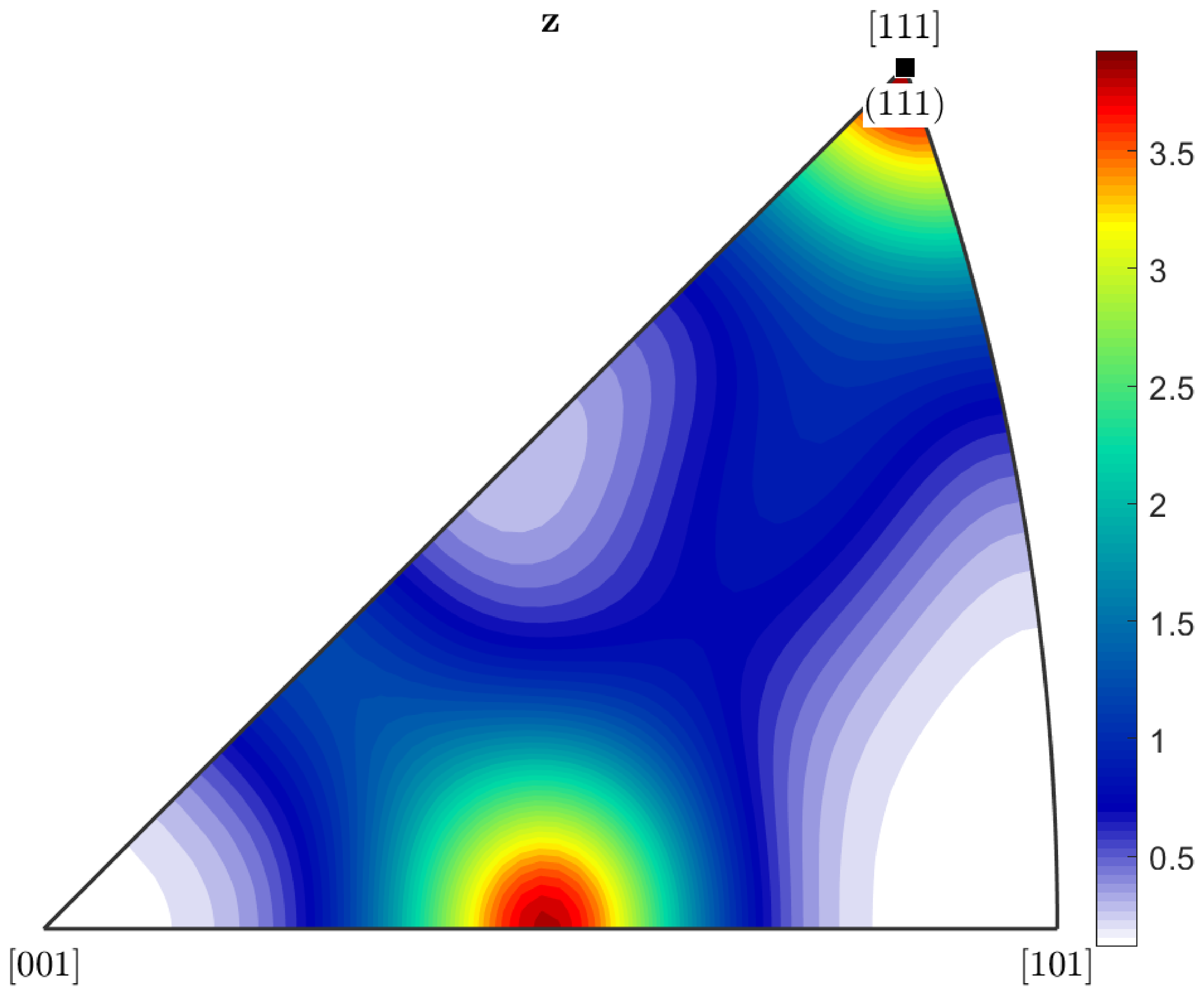
```
>> Alex_trial
  0 |  0.76
  1 |  0.65
  2 |  0.59
  3 |  0.57
  4 |  0.55
  5 |  0.54
  6 |  0.53
  I'm going to apply ghost
correction. Uniform portion fixed
to 0
  0 |  0.80
  1 |  0.68
  2 |  0.62
  3 |  0.60
  4 |  0.59
  5 |  0.57
  6 |  0.57
```

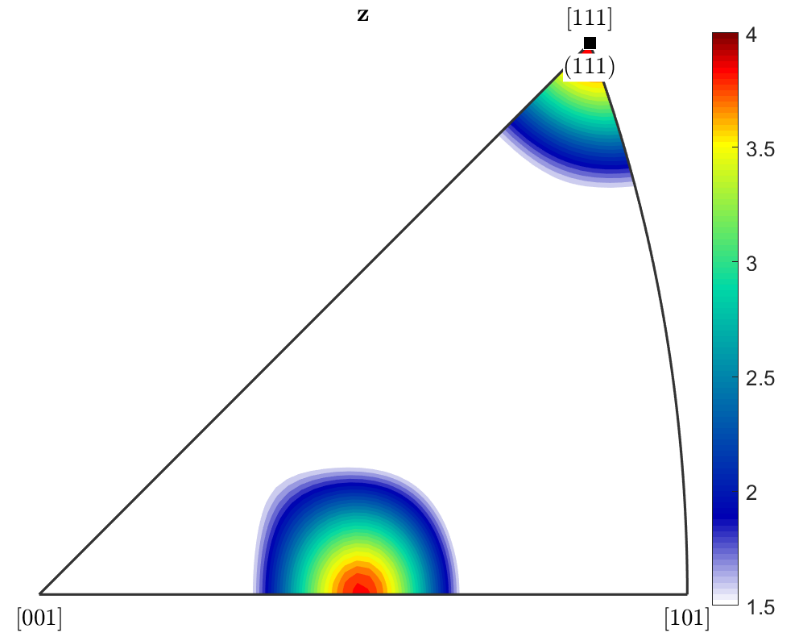

(111)
Y
X
900
800
700
600
500
400
300
200
100


(111)
Y
X
900
800
700
600
500
400
300
200
100


(111)
Y
X
4
3.5
3
2.5
2
1.5


(111)
Y
X
3.5
3
2.5
2
1.5
1
0.5


z
[111]
(111)
[001]
[101]
3.5
3
2.5
2
1.5
1
0.5


z
[111]
(111)
[001]
[101]
4
3.5
3
2.5
2
1.5