

```

1  clc;
2  clear all;
3  % Speed of light and lambda calculation
4  C = 2.99792458e8;
5  freq_Hz = 1420e6;
6  lambda_m = C/freq_Hz;
7
8  %read an image and capture its size, flip and take fft and fftshift. plot its real, imag and
   absolute value
9  H = imread("/home/dsp/Downloads/friendlyVRI-master/models/compact.png");
10 subplot(4,4,1)
11 contour(H);
12 title('input image');
13 n = size(H);
14 nx = n(1);
15 ny = n(2);
16 modelImgArr = flipud(H);
17 modelFFTarr = fftshift(fft2(modelImgArr));
18 subplot(4,4,2)
19 contour(real(modelFFTarr))
20 title('real')
21 subplot(4,4,3)
22 contour(imag(modelFFTarr))
23 title('imag');
24 subplot(4,4,4)
25 contour(abs(modelFFTarr))
26 title('abs');
27
28 %Assumed resolution in arcsec
29 pixScaleImg_asec = 1;
30 pixScaleImg_lam = deg2rad(pixScaleImg_asec/3600.0);
31 fftScale_lam = 1.0/pixScaleImg_lam;
32 pixScaleFFTX_lam = 2.0*fftScale_lam/nx;
33 pixScaleFFTY_lam = 2.0*fftScale_lam/ny;
34 uvMaskArr = zeros(nx,ny);
35
36 % Calculate the hour angles
37 sampRate_deg = 60 * 15.0 / 3600.0;
38 sampRate_hr = 60 / 3600.0;
39 hastart= -1;
40 haend= 1;
41 nSamps = ((haend - (hastart))/sampRate_hr +1);
42 haArr_hr = linspace(hastart, haend, nSamps);
43 haArr_rad = deg2rad(haArr_hr * 15.0);
44
45 %Provide the parameters to get xyz coordinate and uv plane values(2 antennas)
46
47 lat1 = -30.312906;
48 lat1 = deg2rad(lat1);
49 %% For different positions of Antennas
50 %test1
51 % eastArr_m = [0 100];
52 % northArr_m = [0 0];
53 % UpArr_m = [0 0];
54 %Test2
55 % eastArr_m = [0 -100];
56 % northArr_m = [0 0];
57 % UpArr_m = [0 0];
58 %Test3
59 % eastArr_m = [0 0];
60 % northArr_m = [0 100];
61 % UpArr_m = [0 0];
62 %Test4
63 % eastArr_m = [0 0];
64 % northArr_m = [0 -100];

```

```

65 % UpArr_m = [0 0];
66 %Test5
67 % eastArr_m = [0 70.71];
68 % northArr_m = [0 70.71];
69 % UpArr_m = [0 0];
70 %Test6
71 % eastArr_m = [0 -70.71];
72 % northArr_m = [0 -70.71];
73 % UpArr_m = [0 0];
74 %Test7
75 % eastArr_m = [0 -70.71];
76 % northArr_m = [0 70.71];
77 % UpArr_m = [0 0];
78 %Test8
79 eastArr_m = [0 70.71];
80 northArr_m = [0 -70.71];
81 UpArr_m = [0 0];
82 %Plot the actual positions of Antenna
83 subplot(4,4,5)
84 plot3(eastArr_m,northArr_m,UpArr_m,'o')
85 xlabel('E')
86 ylabel('N')
87 zlabel('up')
88 title('ant pos')
89
90 nAnt = length(eastArr_m);
91 nBase = nAnt*((nAnt-1)/2);
92
93     xArr_m = -northArr_m*sin(lat1);
94     yArr_m = eastArr_m;
95     zArr_m = northArr_m*cos(lat1);
96
97 %Calculation of baseline
98 Bx_m = zeros(nBase)
99 By_m = zeros(nBase)
100 Bz_m = zeros(nBase)
101
102     n = 1;
103     for i = 1: nAnt
104         for j = i+1 : nAnt
105             Bx_m(n) = xArr_m(j) - xArr_m(i);
106             By_m(n) = yArr_m(j) - yArr_m(i);
107             Bz_m(n) = zArr_m(j) - zArr_m(i);
108             n += 1;
109         end
110     end
111 % Calculate vector of baseline lengths
112 lBase_m = sqrt(Bx_m.^2.0 + By_m.^2.0 + Bz_m.^2.0);
113
114 %Angle of declination
115 dec_deg = 20.0;
116 dec_rad = deg2rad(-20);
117 %Calculation of u,v plane
118 for i = 1: nBase
119     u_m(i, :) = (Bx_m(i) * sin(haArr_rad) +
120                 By_m(i) * cos(haArr_rad));
121     v_m(i, :) = (-Bx_m(i) * sin(dec_rad) *
122                 cos(haArr_rad) +
123                 By_m(i) * sin(dec_rad) *
124                 sin(haArr_rad) +
125                 Bz_m(i) * cos(dec_rad));
126 end
127
128 %Normalization with lambda
129 uArr_lam = u_m./lambda_m;

```

```

130 vArr_lam = v_m./lambda_m;
131 %wArr_lam = w_m./lambda_m;
132
133 subplot(4,4,6)
134 plot3([uArr_lam,-uArr_lam],[vArr_lam,-vArr_lam],'o');
135 xlabel('u')
136 ylabel('v')
137 title('ha +_1 de -20 l=21cm lat=-30')
138
139 %Gridding
140 u_lam = uArr_lam(:);
141 v_lam = vArr_lam(:);
142
143 u_pixt = (u_lam+fftScale_lam)/pixScaleFFTX_lam;
144 v_pixt = (v_lam+fftScale_lam)/pixScaleFFTY_lam;
145 u2_pixt = (-u_lam+fftScale_lam)/pixScaleFFTX_lam;
146 v2_pixt = (-v_lam+fftScale_lam)/pixScaleFFTY_lam;
147
148 u_pix = fix(u_pixt);
149 v_pix = fix(v_pixt);
150 u2_pix = fix(u2_pixt);
151 v2_pix = fix(v2_pixt);
152
153 subplot(4,4,7)
154 plot([u_pix,u2_pix],[v_pix,v2_pix],'o');
155 xlabel('u');
156 ylabel('v');
157 title('translated pix')
158
159 %Masking
160 for j = 1: length(u_pix)
161 uvMaskArr(v_pix(j), u_pix(j)) = 1;
162 uvMaskArr(v2_pix(j), u2_pix(j)) = 1;
163 end
164
165 subplot(4,4,8)
166 contour(uvMaskArr);
167 title('uvMaskArr');
168
169 %Observed FFT
170 obsFFTTarr = modelFFTTarr.*uvMaskArr;
171
172 %subplot(4,4,9)
173 %contour(obsFFTTarr);
174 %title('obsFFTTarr');
175
176 subplot(4,4,9)
177 contour(real(obsFFTTarr));
178 title('real obsFFT');
179
180 subplot(4,4,10)
181 contour(imag(obsFFTTarr));
182 title('imag obsFFT');
183
184 subplot(4,4,11)
185 contour(abs(obsFFTTarr));
186 title('abs obsFFT');
187
188 %beamArr = ifftshift(ifft2(uvMaskArr));
189
190 %Observed image
191
192 obsImgArr = ifft2(ifftshift(obsFFTTarr));
193 %subplot(4,4,13)
194 %contour(obsImgArr);

```

```
195 %title('obsImgArr');
196
197 subplot(4,4,13)
198 contour(real(obsImgArr));
199 title('real obsimg');
200
201 subplot(4,4,14)
202 contour(imag(obsImgArr));
203 title('imag obsimg');
204
205 subplot(4,4,15)
206 contour(abs(obsImgArr));
207 title('abs obsimg');
208 imshow(obsImgArr);
```