**DHEEPAK,PTU**

**CODE:**

close all;

clc;

disp('PHASE CORRECTION IN DIRECT-FED REFLECTARRAY ANTENNA ');

frequency=input('Enter the Desired Frequency:(in Giga Hertz) ');

c=3\*10^8;

lambda=c/(frequency\*10^9);

rel\_permitivity=input('Enter the value of Relative Permitivity ');

wavelength\_guided=lambda/sqrt(rel\_permitivity);

patch\_size=0.5\*lambda\*1000;

row=input('Enter the no. of ROWs of array elements ');

column=input('Enter the no. of COLUMNs of array elements ');

focalpt=row\*patch\_size;

disp('Focal point is ');

disp(focalpt);

disp('Generating phase shift of Array Elements as Zeros ');

A=zeros(row,column);

disp(A);

psx= zeros(row,column);

psy=zeros(row,column);

check = mod(row,2);

if check == 0

centerx=floor(row/2);

else

centerx = floor(row/2)+1;

end

if check == 0

centery=floor(column/2);

else

centery=floor(column/2)+1;

end

posx(centerx,centerx) = 0;

%posx(:,centerx)=0;

posy(centery,centery) = 0;

disp('PSX w.r.t CENTRE ROW');

val=0;

for x=centery+1:column

val=val+patch\_size;

for y=1:row

psx(y,x)=val;

end

end

val=0;

for x=centery-1:-1:1

val=val+patch\_size;

for y=1:row

psx(y,x)=(-val);

end

end

val=0;

disp(psx);

disp('PSY w.r.t CENTRE COLUMN');

for i=centerx-1:-1:1

val=val+patch\_size;

for j=1:column

psy(i,j)=val;

end

end

val=0;

i=centerx;

for i=centerx+1:row

val=val+patch\_size;

for j=1:column

psy(i,j)=(-val);

end

end

val=0;

disp(psy);

dz = focalpt;

dy = input('Enter the angle of offset beam(Elevation) : ');

if dy~=0

theta = theta\*pi/180;

dx = input ('Enter the angle of offset beam (Azimuth ');

azimuth=azimuth\*pi/180;

else if dy == 0

dx = input ('Enter the angle of offset beam (Azimuth : ');

if dx~=0

azimuth=azimuth\*pi/180;

else

azimuth =0;

end

end

end

distance\_feed=sqrt((((psx(centerx,centery)-dx)^2)+((psy(centerx,centery)-dy)^2)+((-dz)^2)));

%dist0=sqrt((((posx(centerx,centery)-d0x)^2)+((posy(centerx,centery)-d0y)^2)+((-d0z)^2)));

%disp('Distance between Centre and Feed Element '+distance\_feed);

phase\_difference=zeros(row,column);

distance\_ele=zeros(row,column);

path\_difference=zeros(row,column);

angul\_wave\_no=2\*pi/lambda;

disp('distance\_feed');

disp(distance\_feed);

for i=1:1:row

for j=1:1:column

distance\_ele(i,j)=(sqrt((((psx(i,j)-dx)^2)+((psy(i,j)-dy)^2)+((dz)^2))));

%dist(i,j)=(sqrt((((posx(i,j)-d0x)^2)+((posy(i,j)-d0y)^2)+((d0z)^2))));

phase\_difference(i,j)=( angul\_wave\_no\*((distance\_ele(i,j)\*(1e-3))-sin(theta)\*(1e-3)\*((psx(i,j)\*cos(azimuth))+(psy(i,j)\*sin(azimuth)))))\*180/pi;

%phi(i,j)=pk\*((dist(i,j).\*1e-3)-sin(theta)\*(1e-3)\*((posx(i,j)\*cos(azimuth))+(posy(i,j)\*sin(azimuth))))\*180/pi;

if phase\_difference (i,j)>360

reminder = rem(phase\_difference(i,j),360);

integer = (phase\_difference(i,j)-reminder)/360;

phase\_difference(i,j)= phase\_difference(i,j)-integer\*360;

end

path\_difference(i,j) = distance\_ele(i,j)- distance\_feed;

end

end

disp('path\_difference:');

disp(path\_difference);

disp('Amount of Phase Shift required at each element');

phasecorrected=phase\_difference-phase\_difference(centerx,centery);

disp(phasecorrected);

%The elements that are not considered are eliminated.

figure;

imagesc(phasecorrected);

colorbar;

title('Phase difference at each Unit Cell');

xlabel('Row Elements');

ylabel('Column Elements');

zlabel('Phase Difference in degree');

figure;

surf(path\_difference);

colorbar;

title('Path difference at Unit Cell');

xlabel('Row Elements');

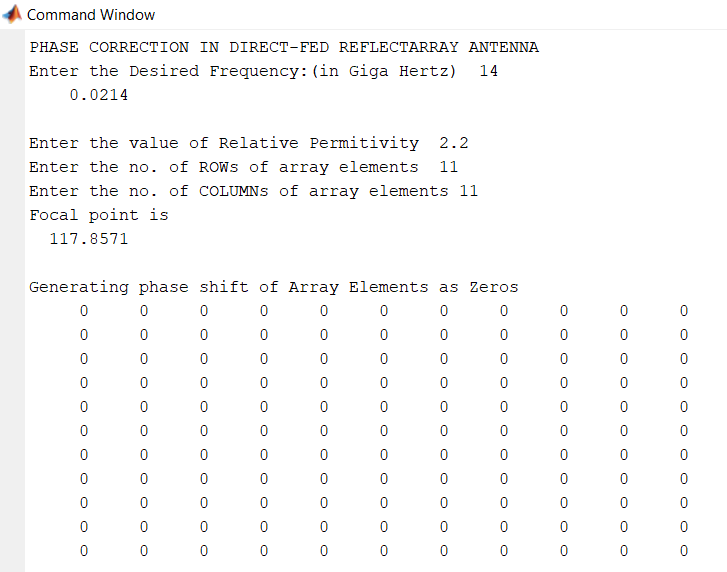
ylabel('Column Elements');

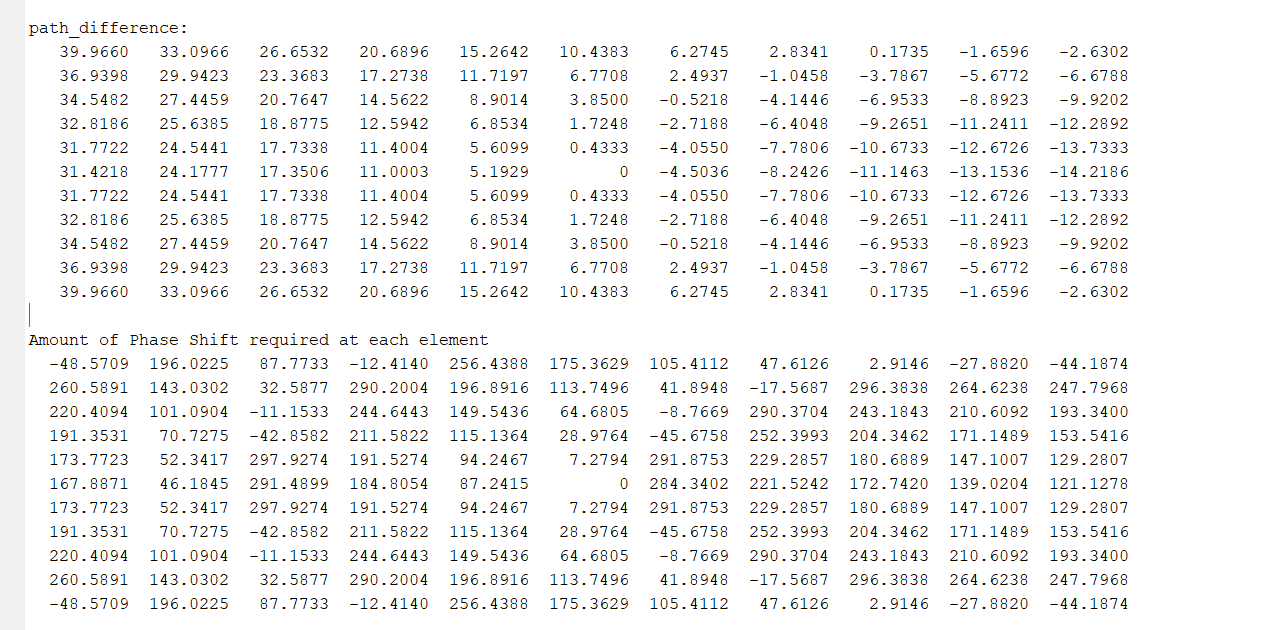
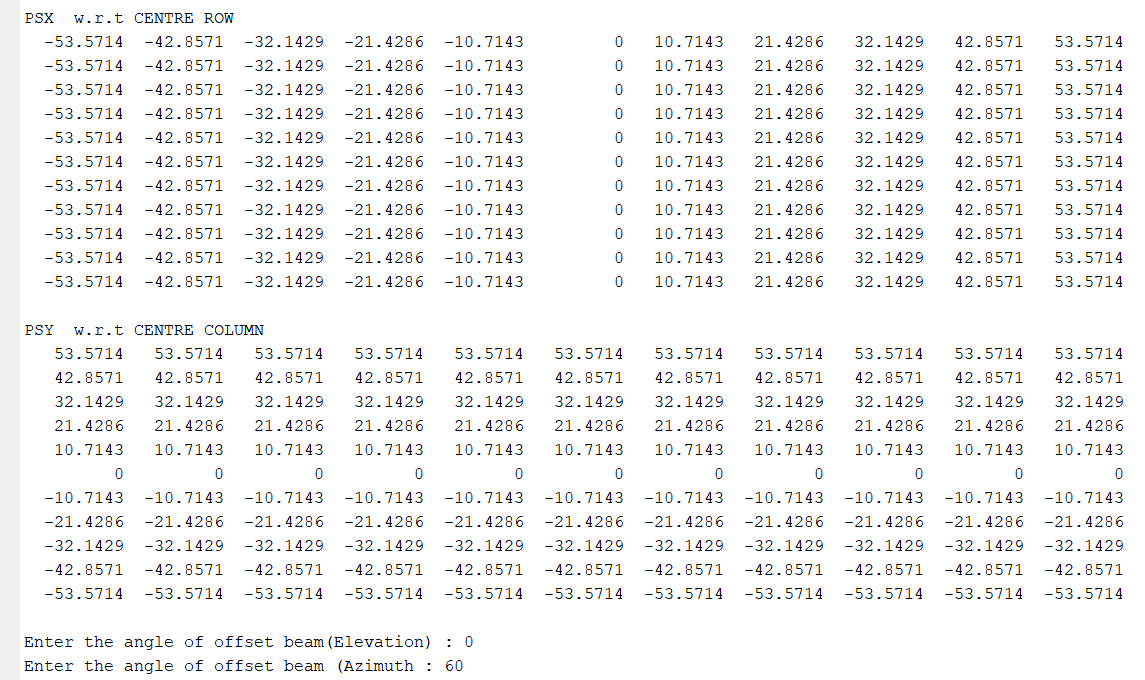
zlabel('Path Difference in mm');

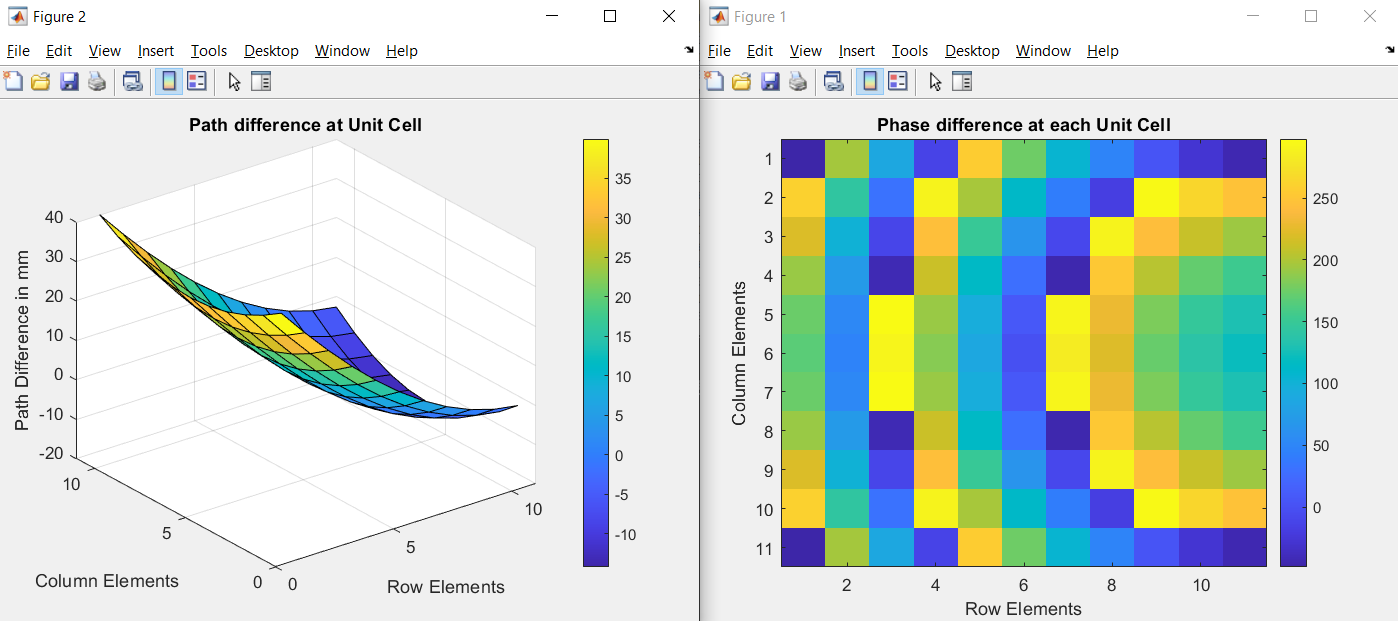
**OUTPUT::: CASE1 for arbitrary values**

**offset beam(Elevation) : 0**

**offset beam (Azimuth) :60**

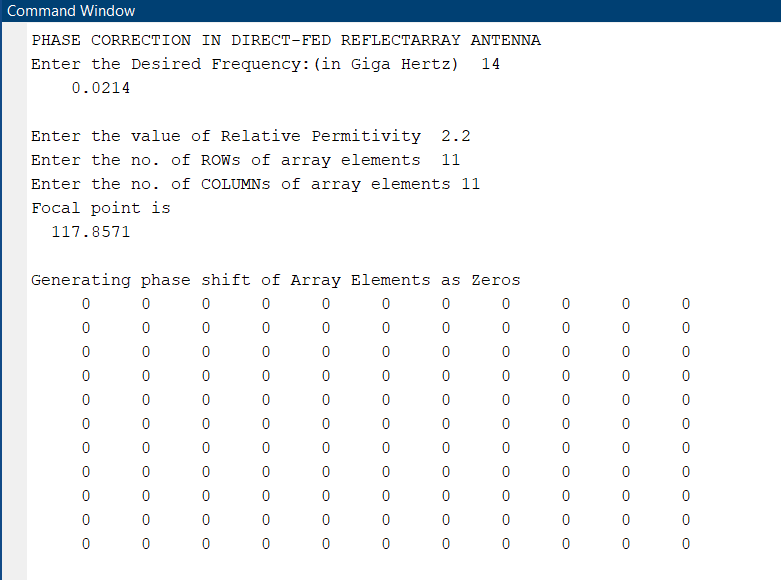
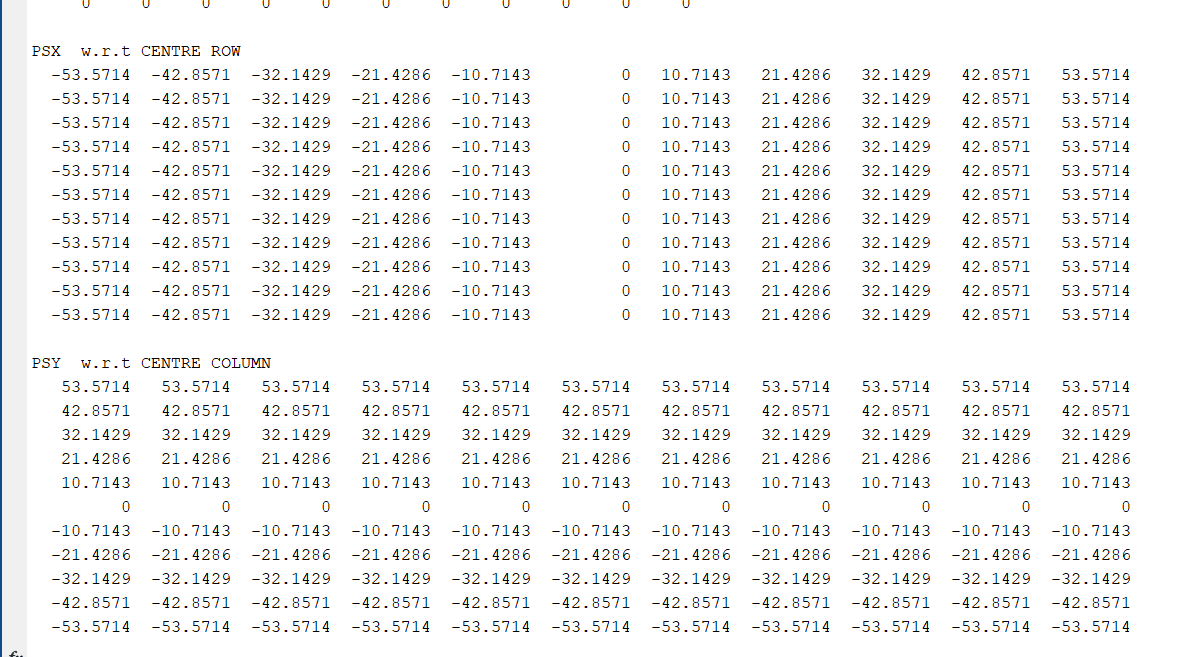
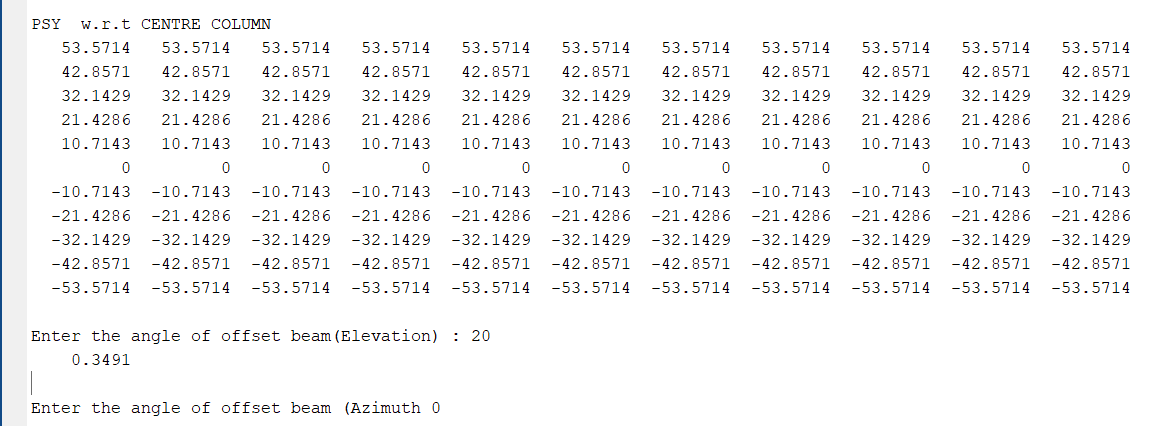
****

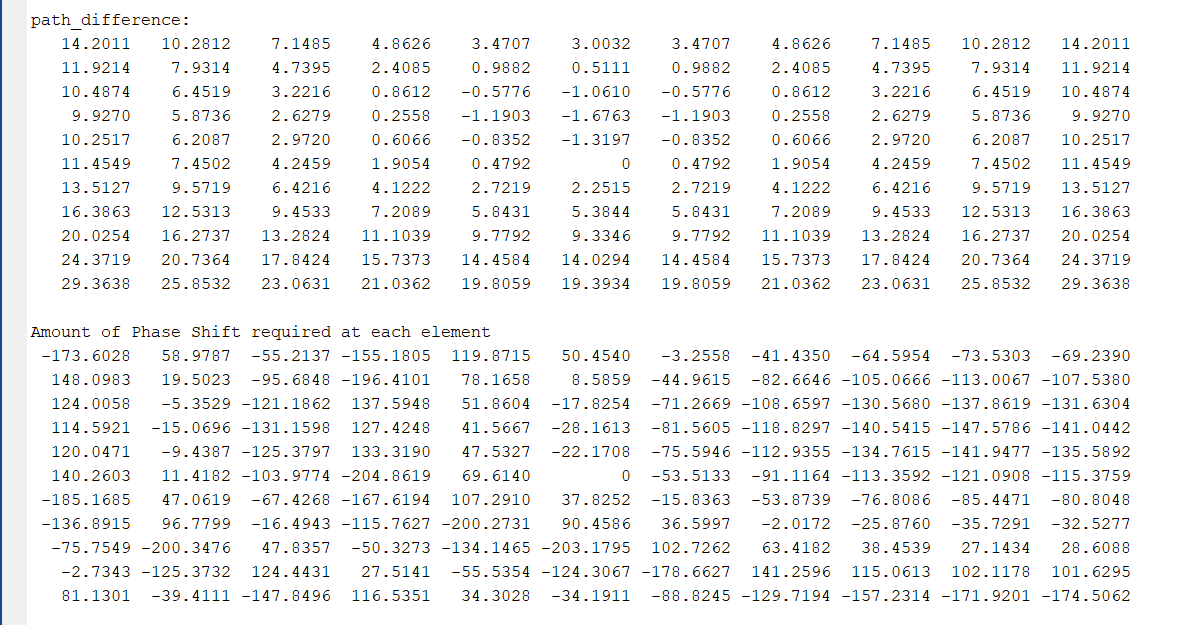
****

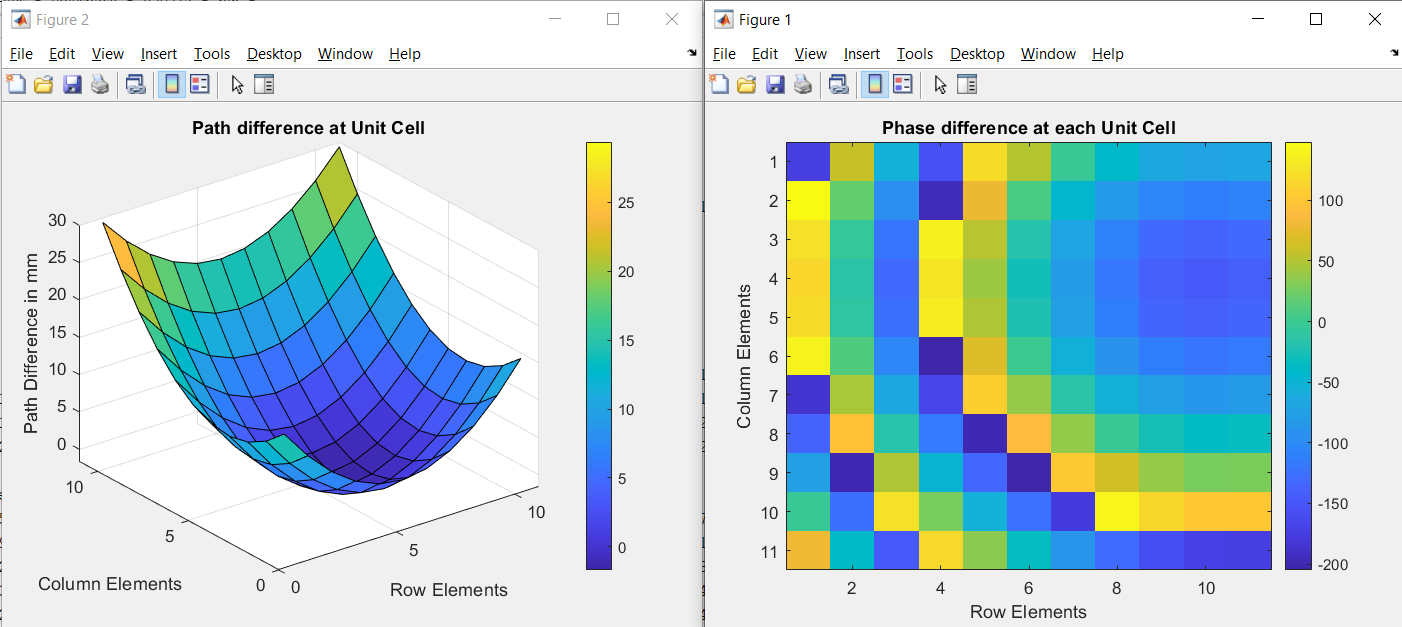
****

**OUTPUT::: CASE2 for arbitrary values**

**offset beam(Elevation) : 20**

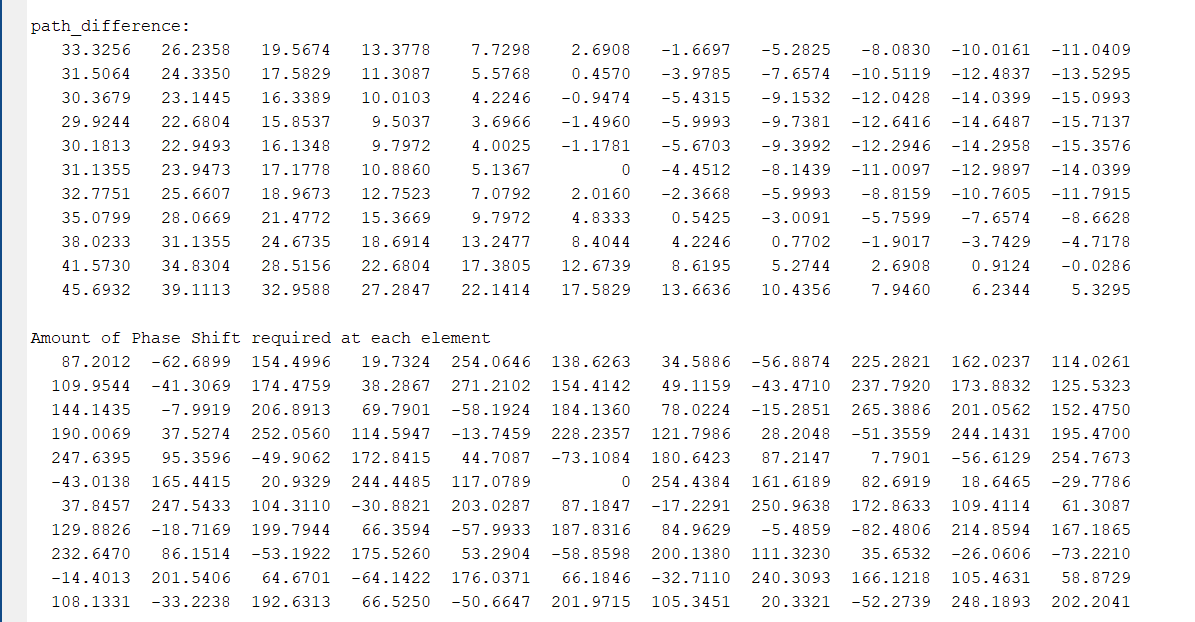
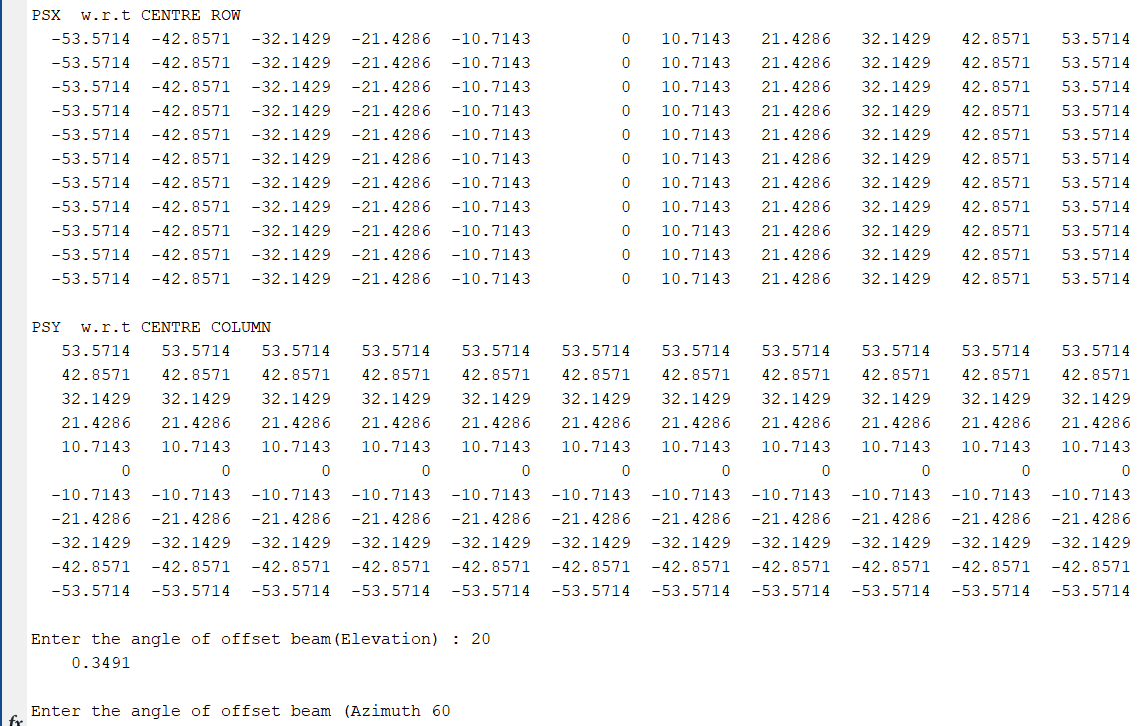
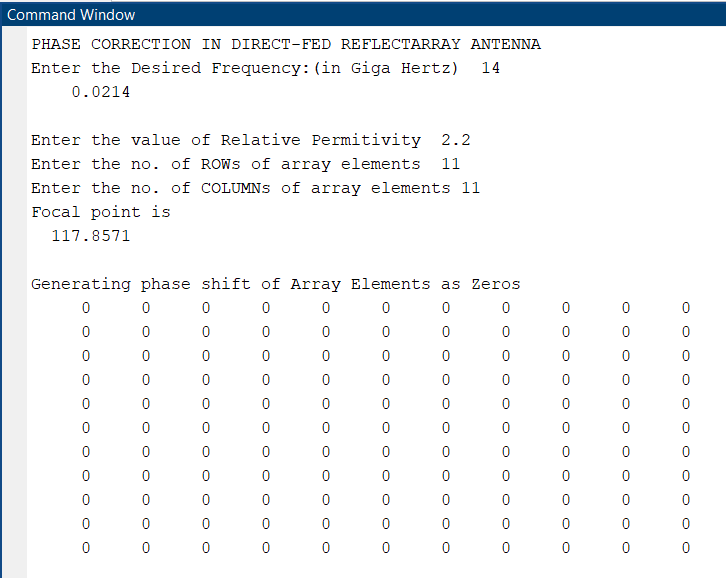
**offset beam (Azimuth) :0******

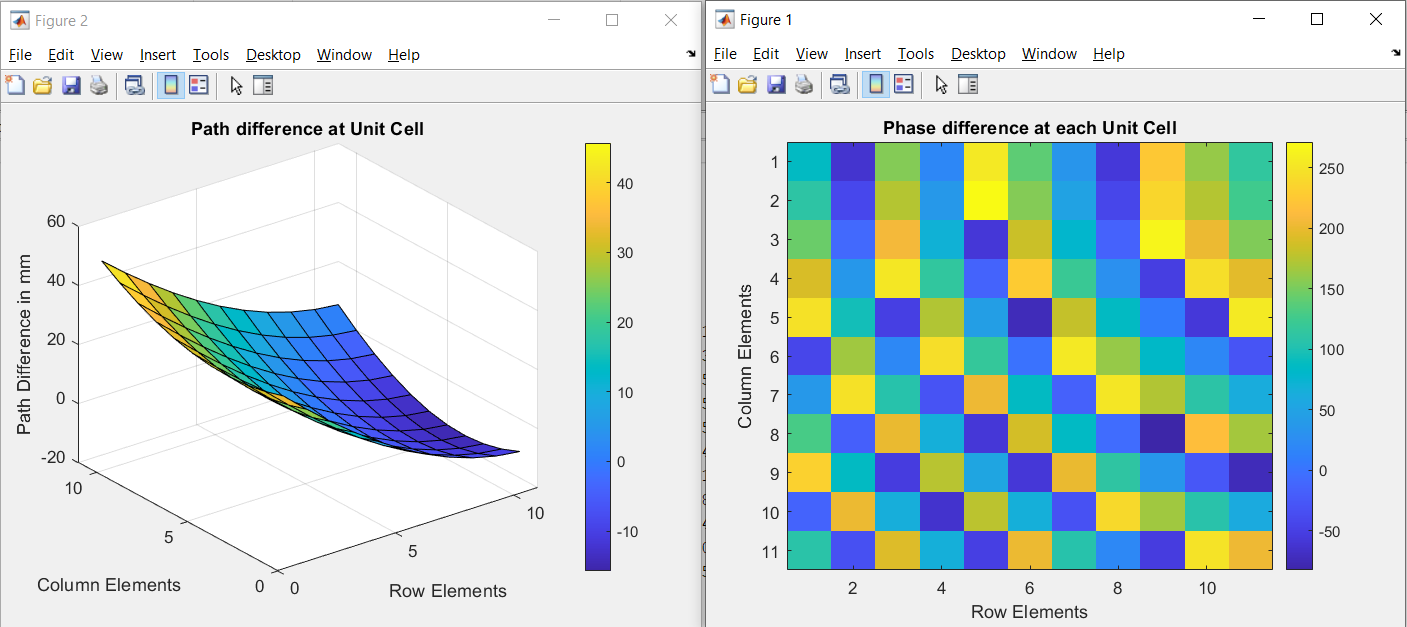


****

**OUTPUT::: CASE2 for arbitrary values**

**offset beam(Elevation) : 20**

**offset beam (Azimuth) :60** 

****