جامعة الإسكندرية
ALEXANDRIA UNIVERSITY
كلية الهندسة
Faculty of Engineering

SECOND LAB
(ASSIGNMENT-2)

” Antenna ”

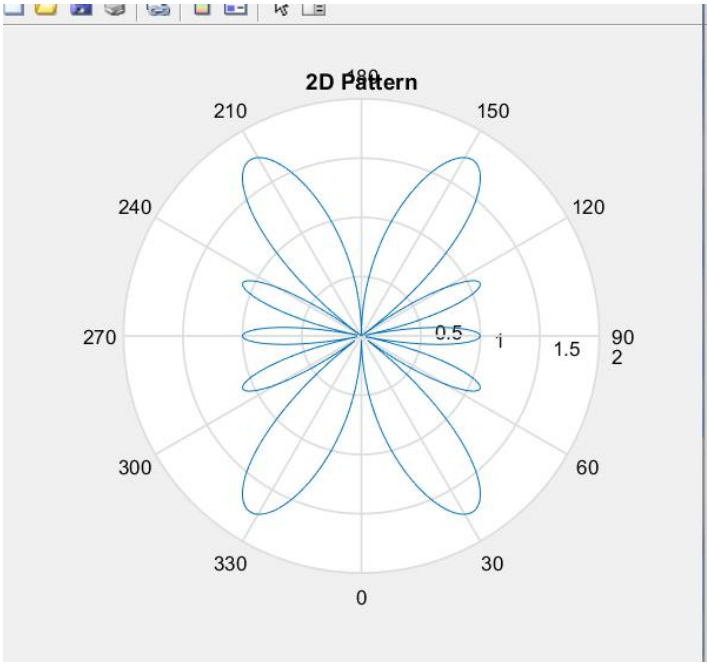
Mariam Hossam Eldeen Saleh	18011752
Ahmed Mahmoud Eldokmak	18010248
Mohamed Samir Abdelrhman	18011483

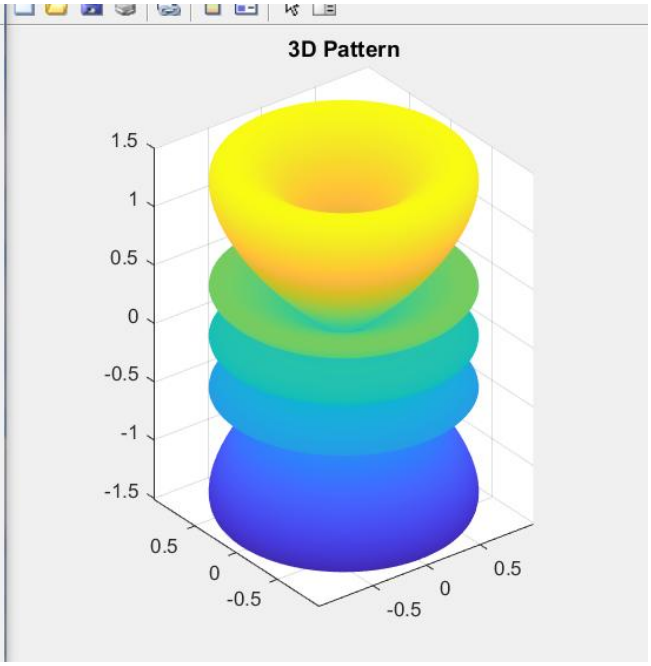
The matlab code:

Part1:linear antenna (dipole of general length)

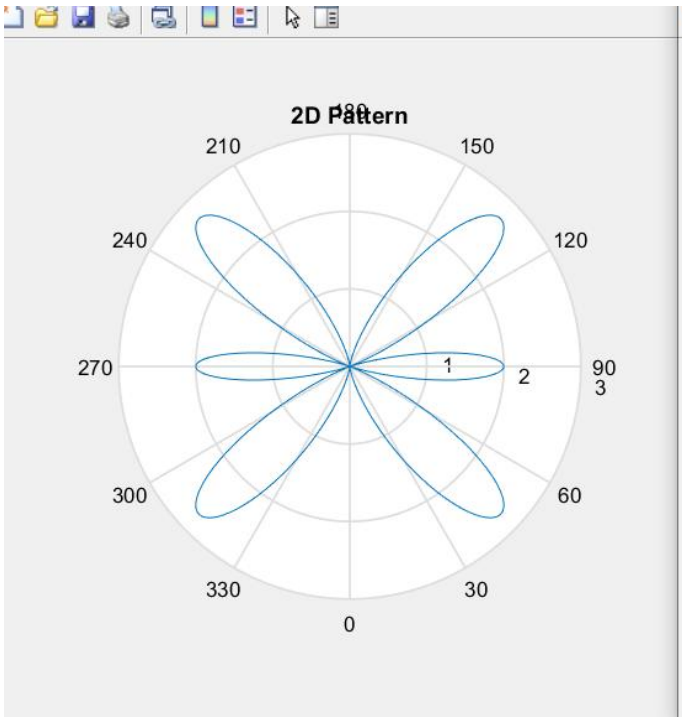
```
Lambda_      = 1;
Betta        = (2*pi)/Lambda_;
Theta        = linspace(-pi,pi,360);
Phi          = linspace(-2*pi,2*pi,360);
L            = input('( Linear antenna )¥nEnter The Lenght Of Dipole L, (L >1) : ');
L            = L * Lambda_;
En           = abs((cos( ( (Betta*L)/2 ).*cos(Theta)) - cos( ( (Betta*L)/2 ))./ sin(Theta));
% 2D Pattern
figure(1);
polar(Theta,En);
view([90,90]);
title('2D Pattern');
% 3D pattern
phi_3D       = meshgrid(Phi);
Theta_3D     = meshgrid(Theta);
En_3D        = meshgrid(En);
X            = En_3D .* sin(Theta_3D) .* cos(phi_3D');
Y            = En_3D .* sin(Theta_3D) .* sin(phi_3D');
Z            = En_3D .* cos(Theta_3D);
figure(2);
surf(X,Y,Z);
shading interp;
axis vis3d;
axis equal;
lighting gouraud;
title('3D Pattern');
```

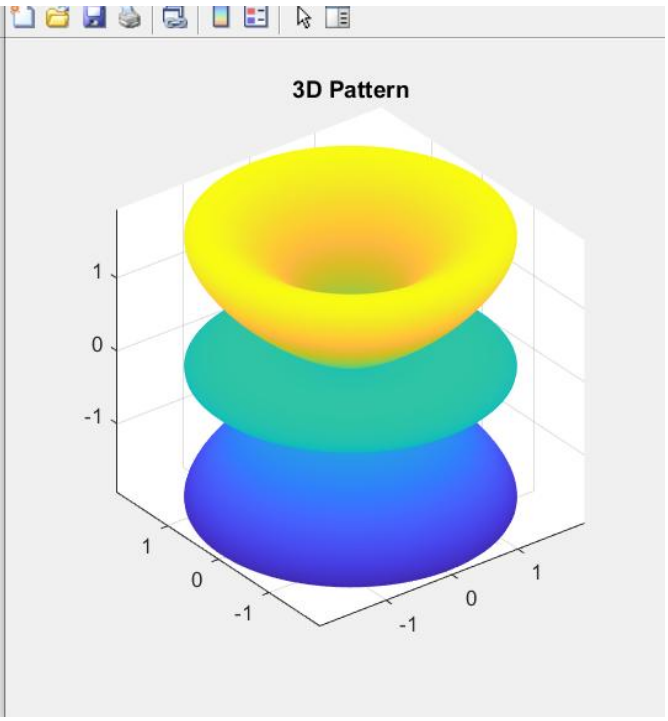
For L = 5/2





For L = 3

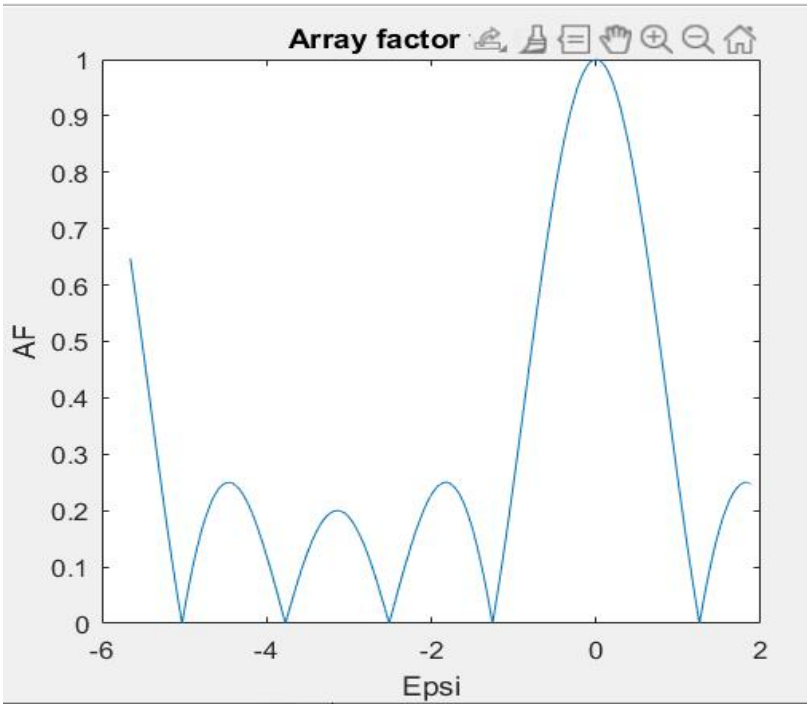


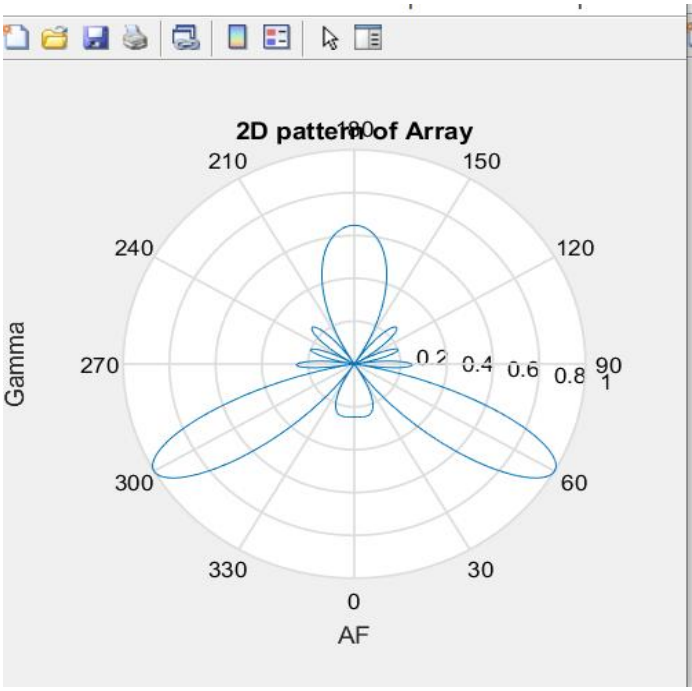


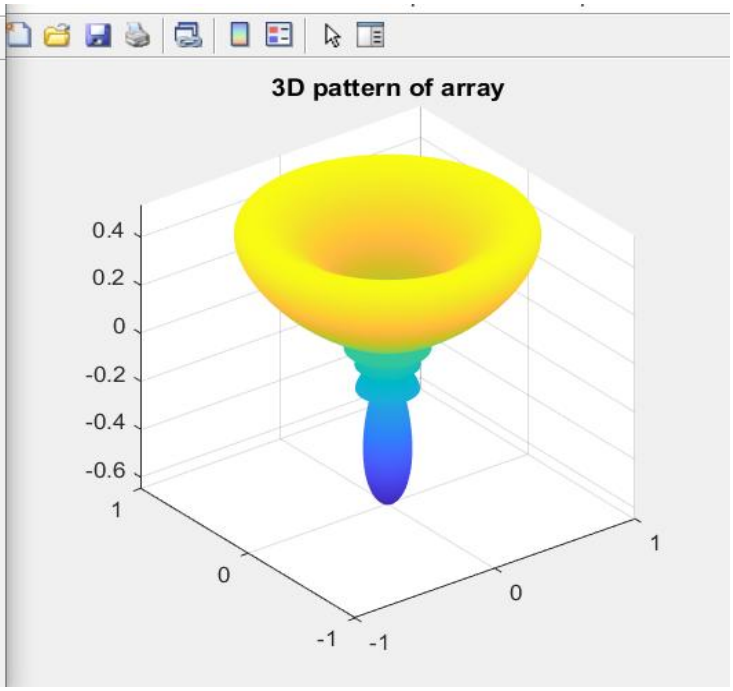
Part2:Uniform linear antenna array(ULA)

```
Lambda_      = 1;
Betta        = (2*pi)/Lambda_;
d       = input('Enter the spacing (d >=0 ) : ');
d       = d * Lambda_ ;
N       = input('Enter the number of elements N (N>=0) : ');
Alpha   = input('Enter the progressive phase shift: ');
Gamma        = linspace(-pi,pi,6000);
Phi          = linspace(-2*pi,2*pi,6000);
Epsi         = Betta*d*cos(Gamma) + Alpha;
AF           = abs( sin((N*Epsi)/2) ./ (N*sin(Epsi/2)) );
% 2D pattern
figure(1);
plot(Epsi,AF);  title('Array factor vs Epsi');
xlabel('Epsi'); ylabel('AF');
figure(2);
polar(Gamma,AF); view([90,90]);
xlabel('Gamma'); ylabel('AF');
% 3D pattern
Phi_3D     = meshgrid(Phi);
Gamma_3D   = meshgrid(Gamma);
AF_3D      = meshgrid(AF);
X          = AF_3D.*sin(Gamma_3D).*cos(Phi_3D');
Y          = AF_3D.*sin(Gamma_3D).*sin(Phi_3D');
Z          = AF_3D.*cos(Gamma_3D);
figure(3);
surf(X,Y,Z);
shading interp;
axis vis3d;
lighting gouraud;
title('3D pattern of array');
```

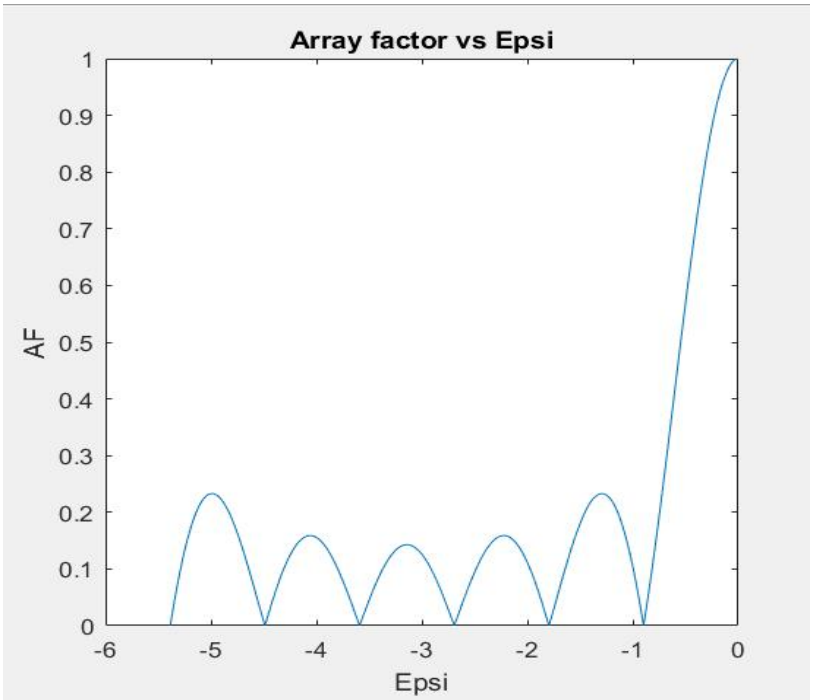
For L=3/5, N=5, α = -3Π/5

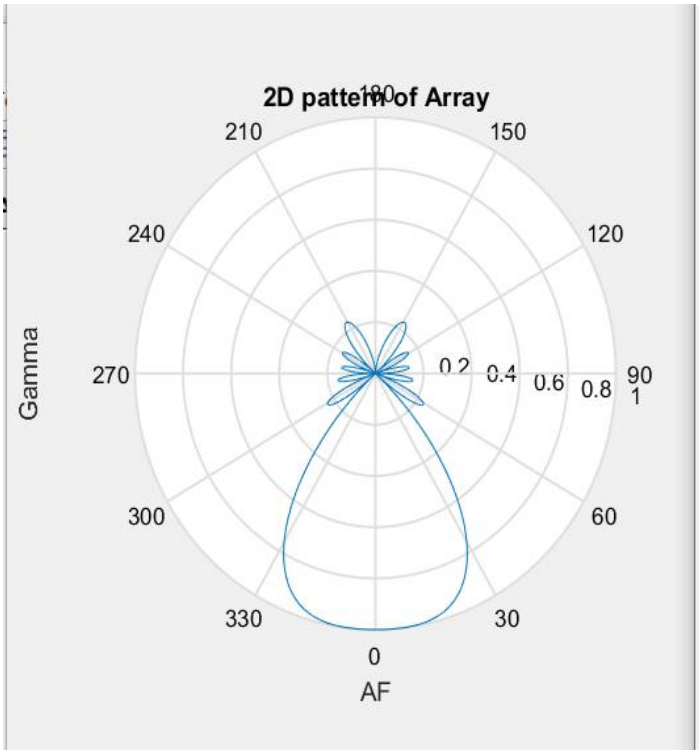


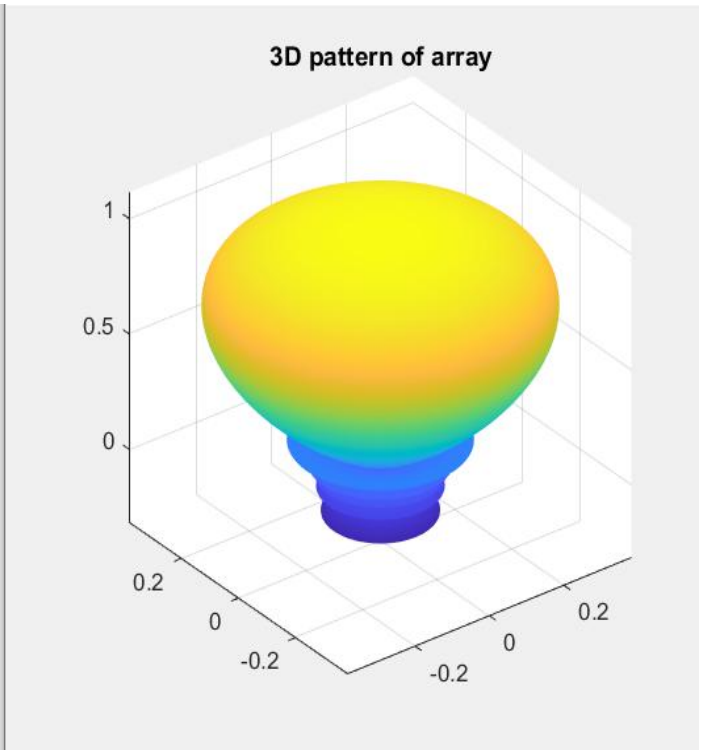




For L=6/14, N=7, α = -6Π/7



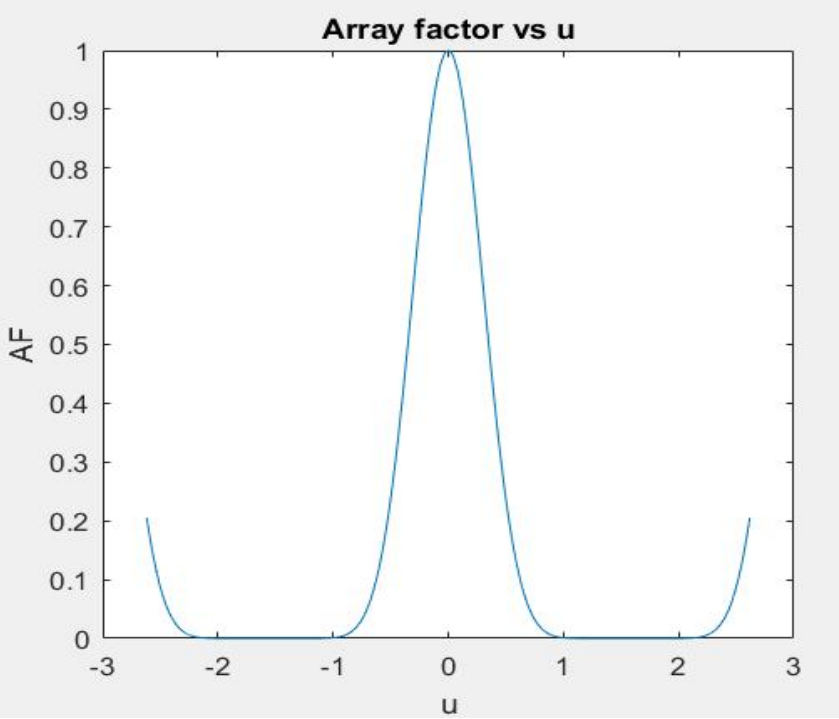


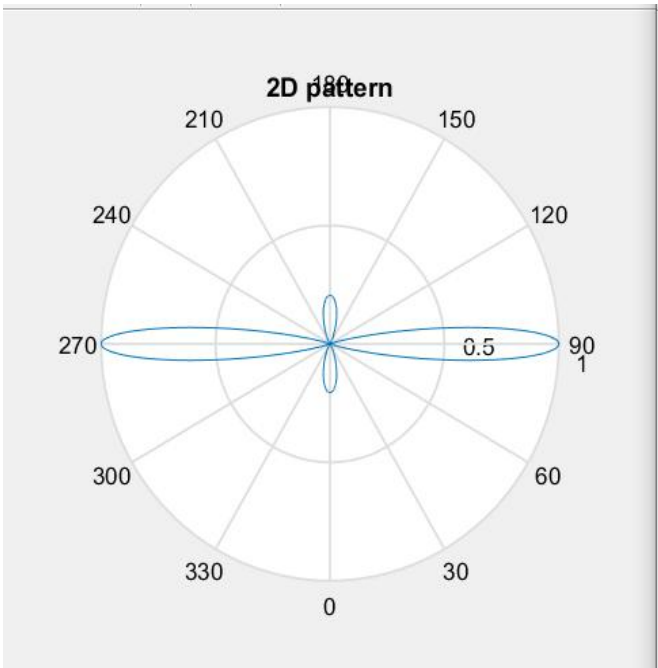


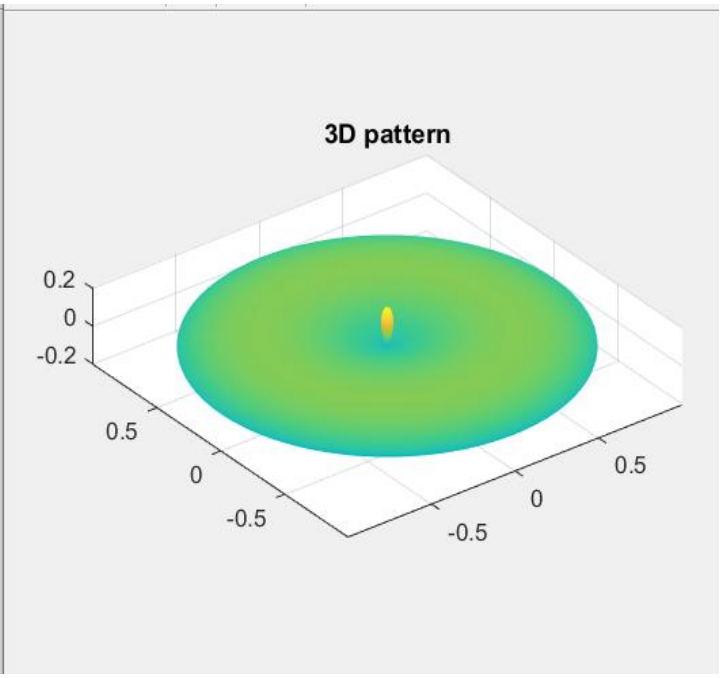
Part3:Nonuniformly-Fed linear antenna array A.Binomial Arrays

```
Lambda_     = 1;
Betta       = (2*pi) / Lambda_;
d       = input('Enter the spacing d (d>=0) : ');
N       = input('Enter the number of elements N (N>=0) : ');
Alpha   = input('Enter the progressive phase shift: ');
Theta       = linspace(-pi,pi,6000);
Phi         = linspace(-2*pi,2*pi,6000);
U           = (Betta*d*cos(Theta) + Alpha) / 2;
AF          = abs(cos(U).^(N-1));
% 2D pattern
figure(1);
plot(U,AF); title('Array factor vs u');
xlabel('u'); ylabel('AF');
figure(2);
polar(Theta, AF);
view([90 90]);
title('2D pattern');
% 3D pattern
Phi_3D   = meshgrid(Phi);
Theta_3D = meshgrid(Theta);
AF_3D    = meshgrid(AF);
X        = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
Y        = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
Z        = AF_3D.*cos(Theta_3D);
figure(3);
surf(X,Y,Z);
shading interp;
axis vis3d;
axis equal;
lighting gouraud;
title('3D pattern');
```

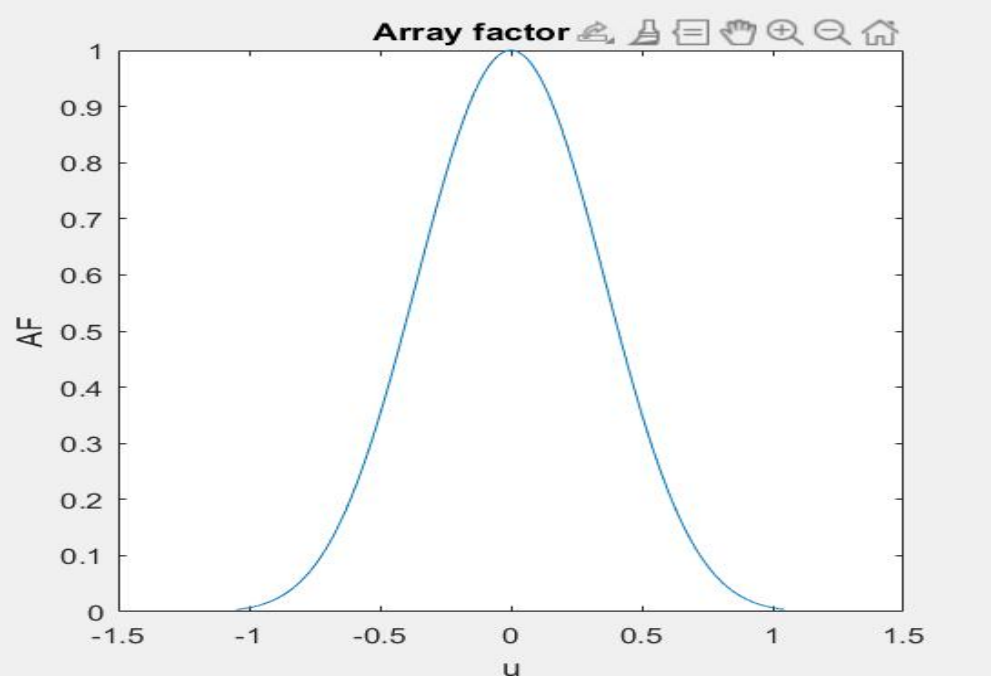
For L=5/6, N=12, α = 0

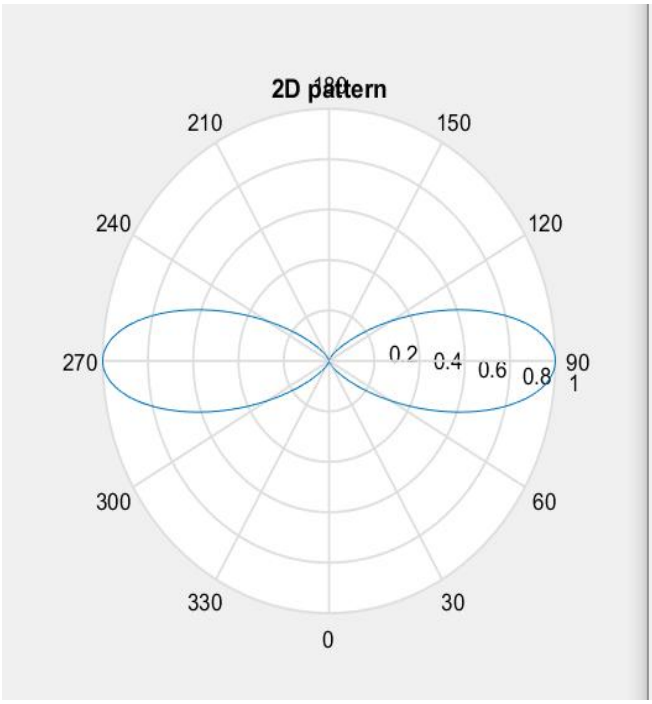


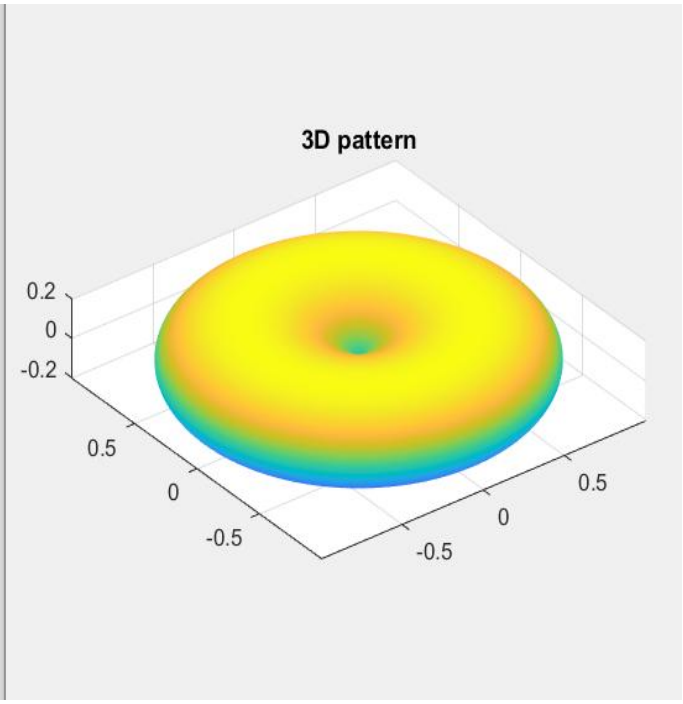




For L=2/6, N=9, α = 0


Array factor
AF
u
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5


2D pattern
180
210
150
240
120
270
90
300
60
330
30
0
0.2
0.4
0.6
0.8
1

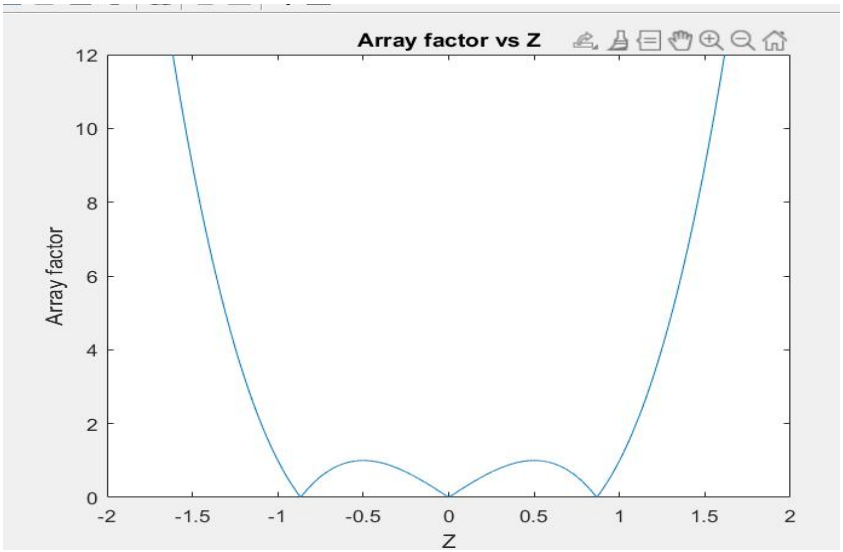

3D pattern
0.2
0
-0.2
0.5
0
-0.5
0.5
0
-0.5

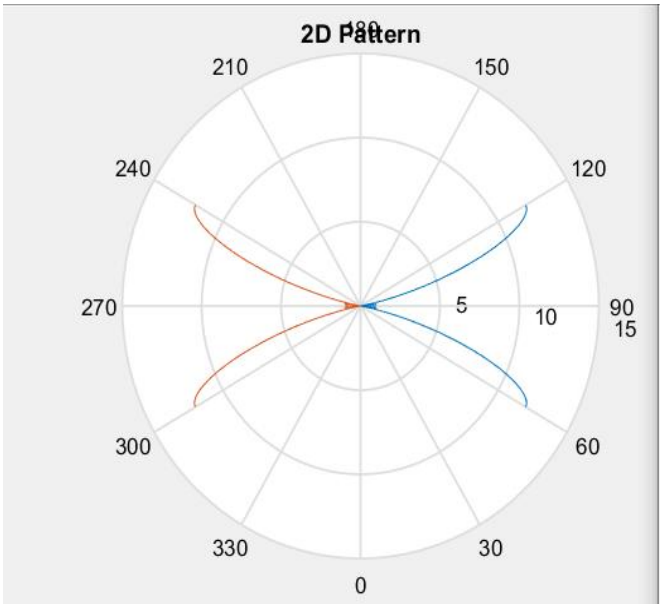
B. Dolph-Tshebysceff Array

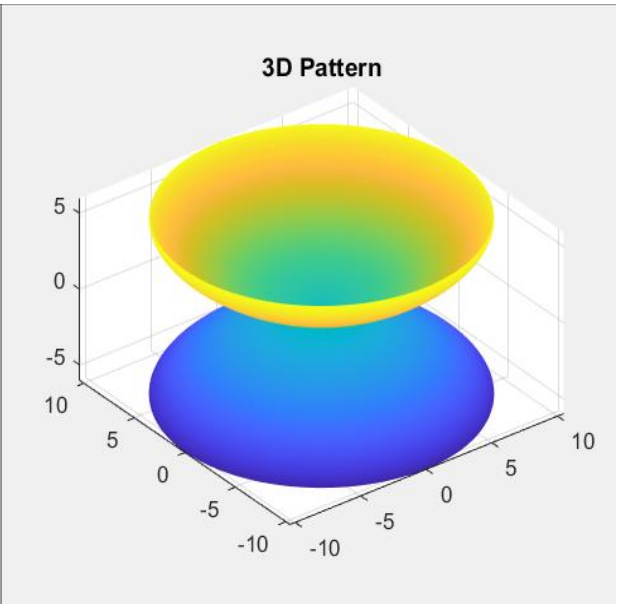
```
Lambda_     = 1;
Betta       = (2*pi) / Lambda_;
d       = input('Enter the spacing d (d>=0) : ');
N       = input('Enter the number of elements N (N>=0) : ');
Alpha   = input('Enter the progressive phase shift: ');
M       = N - 1;
Ro      = input('Mainlobe to sidelobe level Ro (Ro>1) : ');
Zo      = cosh((1/M)*acosh(Ro));
Z       = linspace(-Zo,Zo,6000);
U_up    = acos(Z./Zo);      U_down  = -U_up;
U       = [U_down ; U_up];
Theta1      = acos(((2.*U_down)-Alpha)/(Betta*d));
Theta2      = -Theta1;
Phi         = linspace(-2*pi,2*pi,6000);
AF          = abs(cosh(M.*acosh(Z)));
% 2D Pattern
figure(1);
plot(Z,AF); title('Array factor vs Z');
xlabel('Z'); ylabel('Array factor');
figure(2);
polar(Theta1,AF);
hold on;
polar(Theta2, AF);
view([90 90]);
title('2D Pattern')
% 3D Pattern
Phi_3D   = meshgrid(Phi);
Theta_3D = meshgrid(Theta1);
AF_3D    = meshgrid(AF);
X        = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
Y        = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
Z        = AF_3D.*cos(Theta_3D);
```

```
figure(3);
surf(X,Y,Z);
shading interp;
axis vis3d;
axis equal;
lighting gouraud;
title('3D Pattern')
```

For L=1, N=4, α = -Π, Ro=12







For L=1/2, N=5, α = -Π, Ro=12