Electrical Engineering Department,

Fourth Year - Communications & Electronics.

EE 466 ANTENNA

Lab Assignment-2

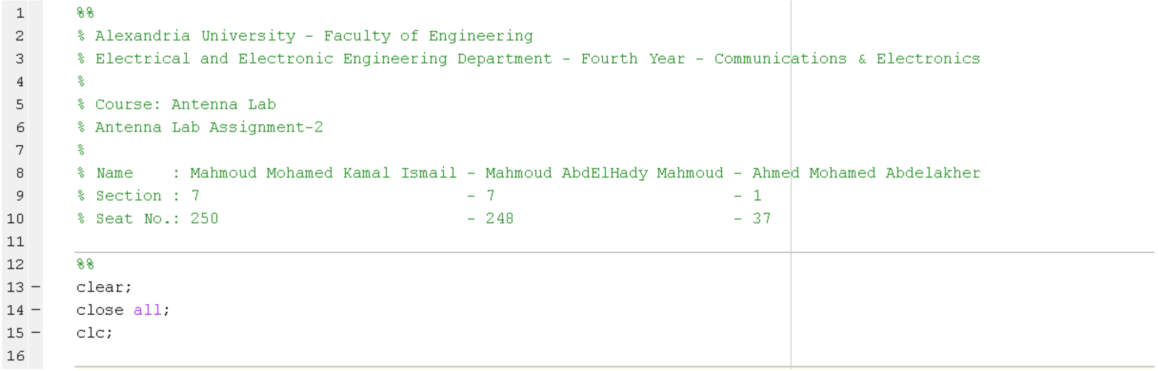
PREPARED BY	SECTION	SEAT.NO.
Mahmoud Mohamed Kamal Ismail	7	250
Mahmoud AbdElHady Mahmoud	7	248
Ahmed Mohamed Abdelakher	1	37

➢ Contents

➢ List of Figures

https://github.com/MahmoudFierro98/Antenna_Lab/

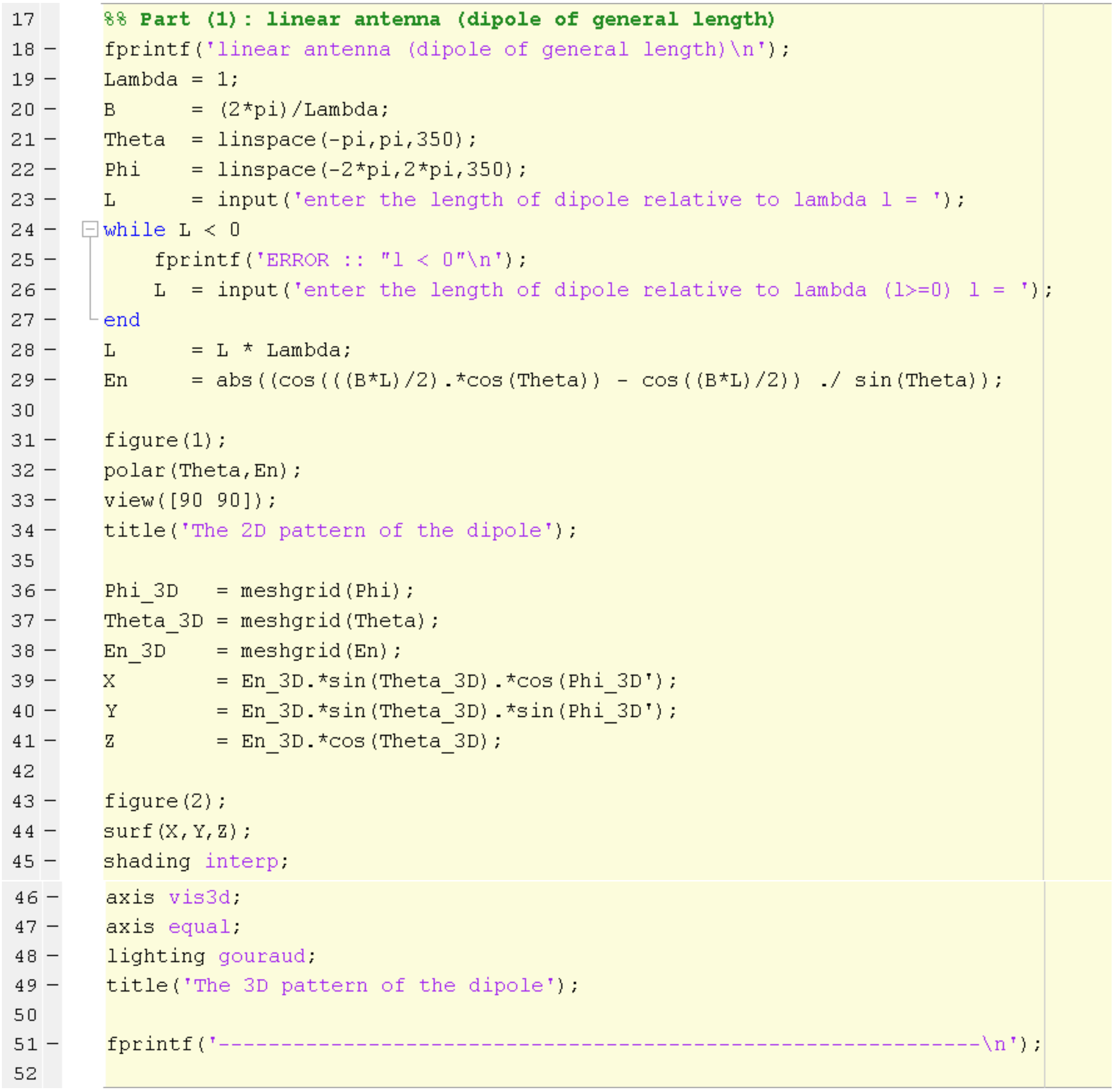
1. Code



```
%%
% Alexandria University - Faculty of Engineering
% Electrical and Electronic Engineering Department - Fourth Year - Communications & Electronics
%
% Course: Antenna Lab
% Antenna Lab Assignment-2
%
% Name    : Mahmoud Mohamed Kamal Ismail - Mahmoud AbdElHady Mahmoud - Ahmed Mohamed Abdelakher
% Section : 7                            - 7                          - 1
% Seat No.: 250                          - 248                        - 37

%%
clear;
close all;
clc;

```

Figure 1: Code – Intro.

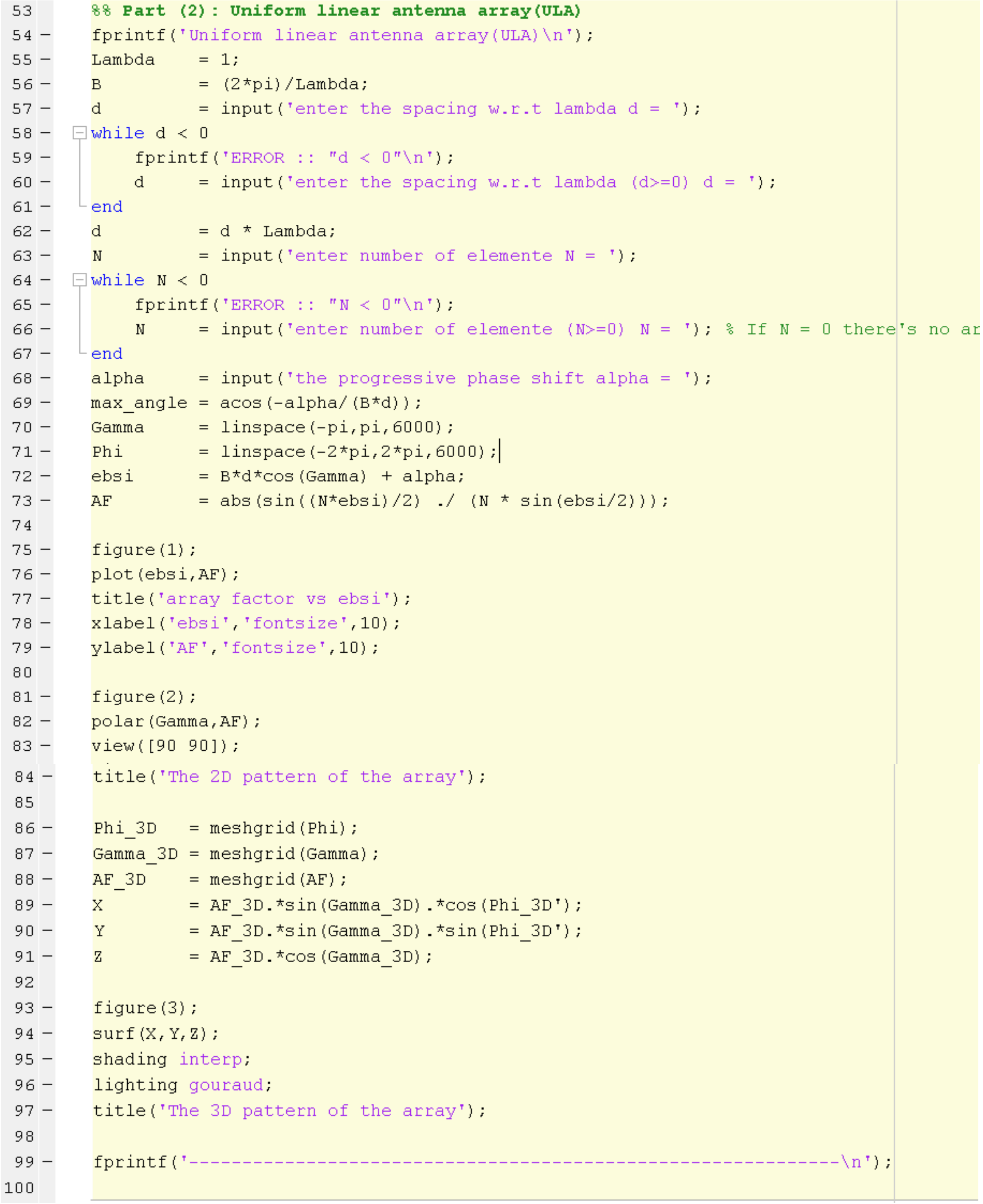
1.1. Part 1: linear antenna (dipole of general length)



```
%% Part (1): linear antenna (dipole of general length)
fprintf('linear antenna (dipole of general length)\n');
Lambda = 1;
B      = (2*pi)/Lambda;
Theta  = linspace(-pi,pi,350);
Phi    = linspace(-2*pi,2*pi,350);
L      = input('enter the length of dipole relative to lambda l = ');
while L < 0
    fprintf('ERROR :: "l < 0"\n');
    L  = input('enter the length of dipole relative to lambda (l>=0) l = ');
end
L      = L * Lambda;
En     = abs((cos(((B*L)/2).*cos(Theta)) - cos((B*L)/2)) ./ sin(Theta));

figure(1);
polar(Theta,En);
view([90 90]);
title('The 2D pattern of the dipole');

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('The 3D pattern of the dipole');

fprintf('------------------------------------------------------------\n');

```

Figure 2: Code - Part 1: linear antenna (dipole of general length).

1.2. Part 2: Uniform linear antenna array (ULA)

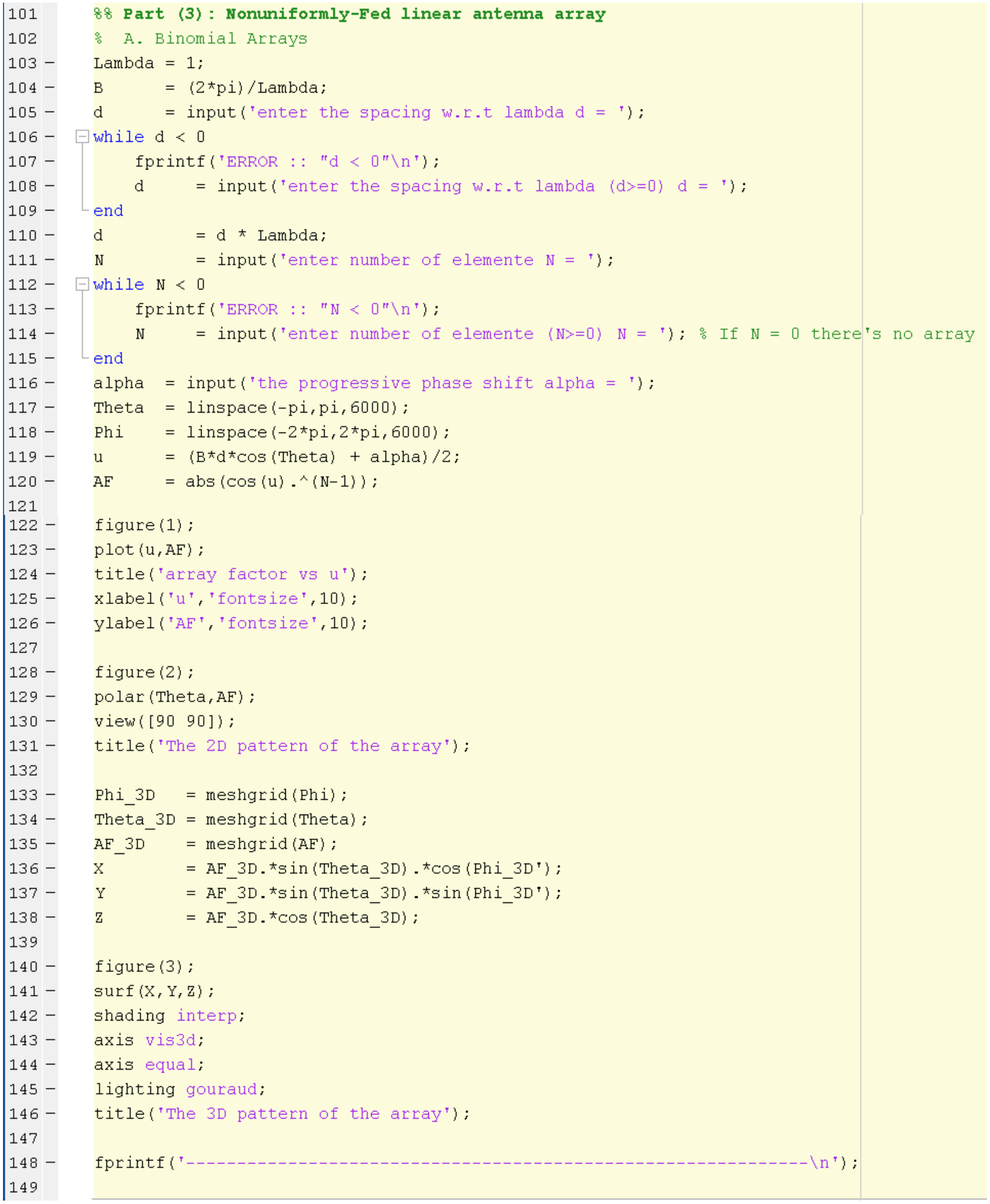


```
%% Part (2): Uniform linear antenna array(ULA)
fprintf('Uniform linear antenna array(ULA)\n');
Lambda     = 1;
B          = (2*pi)/Lambda;
d          = input('enter the spacing w.r.t lambda d = ');
while d < 0
    fprintf('ERROR :: "d < 0"\n');
    d      = input('enter the spacing w.r.t lambda (d>=0) d = ');
end
d          = d * Lambda;
N          = input('enter number of elemente N = ');
while N < 0
    fprintf('ERROR :: "N < 0"\n');
    N      = input('enter number of elemente (N>=0) N = '); % If N = 0 there's no ar
end
alpha      = input('the progressive phase shift alpha = ');
max_angle = acos(-alpha/(B*d));
Gamma      = linspace(-pi,pi,6000);
Phi        = linspace(-2*pi,2*pi,6000);
ebsi       = B*d*cos(Gamma) + alpha;
AF         = abs(sin((N*ebsi)/2) ./ (N * sin(ebsi/2)));

figure(1);
plot(ebsi,AF);
title('array factor vs ebsi');
xlabel('ebsi','fontsize',10);
ylabel('AF','fontsize',10);

figure(2);
polar(Gamma,AF);
view([90 90]);
title('The 2D pattern of the array');

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;
lighting gouraud;
title('The 3D pattern of the array');

fprintf('------------------------------------------------------------\n');

```

Figure 3: Code - Part 2: Uniform linear antenna array (ULA).

1.3. Part 3: Nonuniformly-Fed linear antenna array

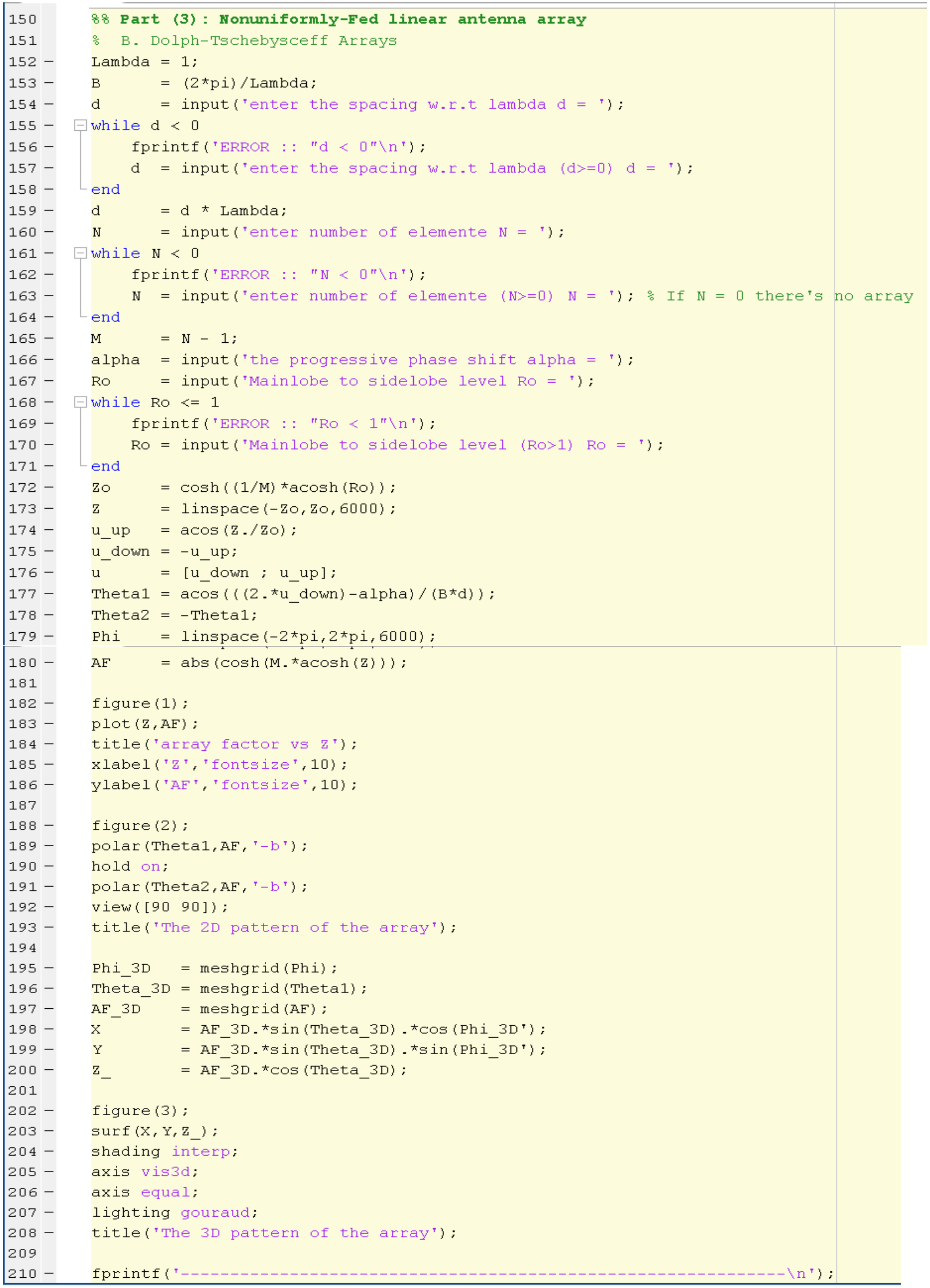
A. Binomial Arrays



```
%% Part (3): Nonuniformly-Fed linear antenna array
%  A. Binomial Arrays
Lambda = 1;
B      = (2*pi)/Lambda;
d      = input('enter the spacing w.r.t lambda d = ');
while d < 0
    fprintf('ERROR :: "d < 0"\n');
    d     = input('enter the spacing w.r.t lambda (d>=0) d = ');
end
d          = d * Lambda;
N          = input('enter number of elemente N = ');
while N < 0
    fprintf('ERROR :: "N < 0"\n');
    N     = input('enter number of elemente (N>=0) N = '); % If N = 0 there's no array
end
alpha  = input('the progressive phase shift alpha = ');
Theta  = linspace(-pi,pi,6000);
Phi    = linspace(-2*pi,2*pi,6000);
u      = (B*d*cos(Theta) + alpha)/2;
AF     = abs(cos(u).^(N-1));

figure(1);
plot(u,AF);
title('array factor vs u');
xlabel('u','fontsize',10);
ylabel('AF','fontsize',10);

figure(2);
polar(Theta,AF);
view([90 90]);
title('The 2D pattern of the array');

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('The 3D pattern of the array');

fprintf('-------------------------------------------------------------\n');

```

Figure 4: Code - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

B. Dolph-Tschebyscff Arrays



```matlab
%% Part (3): Nonuniformly-Fed linear antenna array
%  B. Dolph-Tschebyscff Arrays
Lambda = 1;
B      = (2*pi)/Lambda;
d      = input('enter the spacing w.r.t lambda d = ');
while d < 0
    fprintf('ERROR :: "d < 0"\n');
    d  = input('enter the spacing w.r.t lambda (d>=0) d = ');
end
d      = d * Lambda;
N      = input('enter number of elemente N = ');
while N < 0
    fprintf('ERROR :: "N < 0"\n');
    N  = input('enter number of elemente (N>=0) N = '); % If N = 0 there's no array
end
M      = N - 1;
alpha  = input('the progressive phase shift alpha = ');
Ro     = input('Mainlobe to sidelobe level Ro = ');
while Ro <= 1
    fprintf('ERROR :: "Ro < 1"\n');
    Ro = input('Mainlobe to sidelobe level (Ro>1) Ro = ');
end
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)/(B*d));
Theta2 = -Theta1;
Phi    = linspace(-2*pi,2*pi,6000);
AF     = abs(cosh(M.*acosh(Z)));

figure(1);
plot(Z,AF);
title('array factor vs Z');
xlabel('Z','fontsize',10);
ylabel('AF','fontsize',10);

figure(2);
polar(Theta1,AF,'-b');
hold on;
polar(Theta2,AF,'-b');
view([90 90]);
title('The 2D pattern of the array');

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('The 3D pattern of the array');

fprintf('------------------------------------------------------------\n');
```

Figure 5: Code - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebyscff Arrays).

2. Examples

2.1. Part 1: linear antenna (dipole of general length)

Example 1: $l = \frac{3\lambda}{2}$

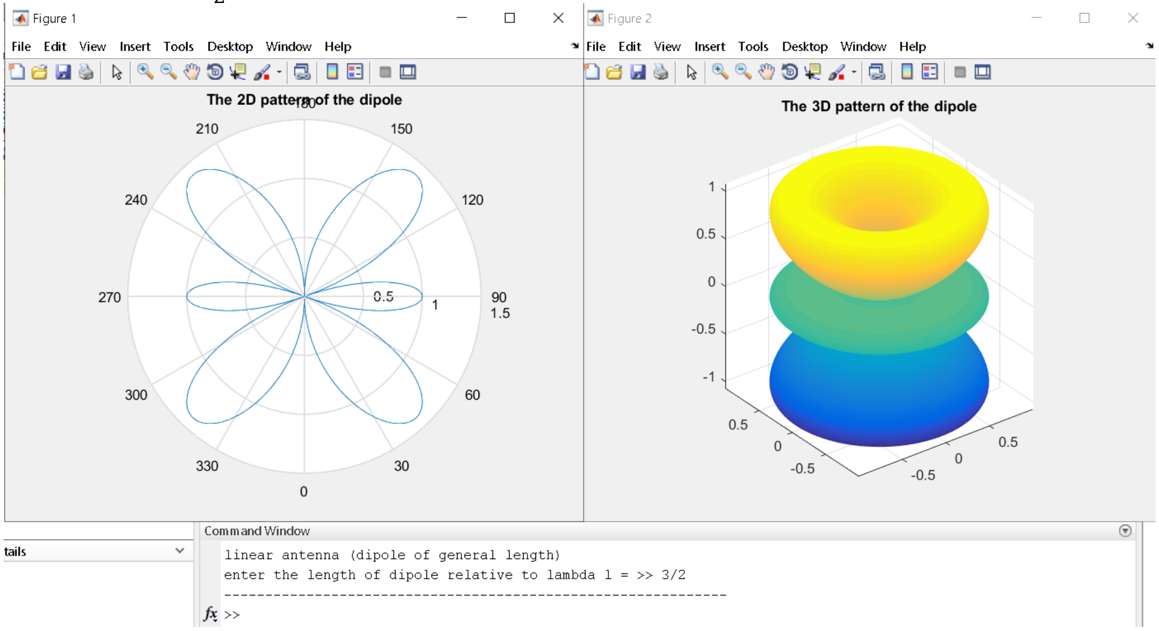


Figure 1
File Edit View Insert Tools Desktop Window Help
The 2D pattern of the dipole
180
210
150
240
120
270
90
1.5
0.5
1
300
60
330
30
0
Figure 2
File Edit View Insert Tools Desktop Window Help
The 3D pattern of the dipole
1
0.5
0
-0.5
-1
0.5
0
-0.5
0.5
0
-0.5
Command Window
tails
fx >>

```
linear antenna (dipole of general length)
enter the length of dipole relative to lambda l = >> 3/2
-------------------------------------------------------------
```

Figure 6: Example 1 - Part 1: linear antenna (dipole of general length).

Example 2: $l = 2\lambda$

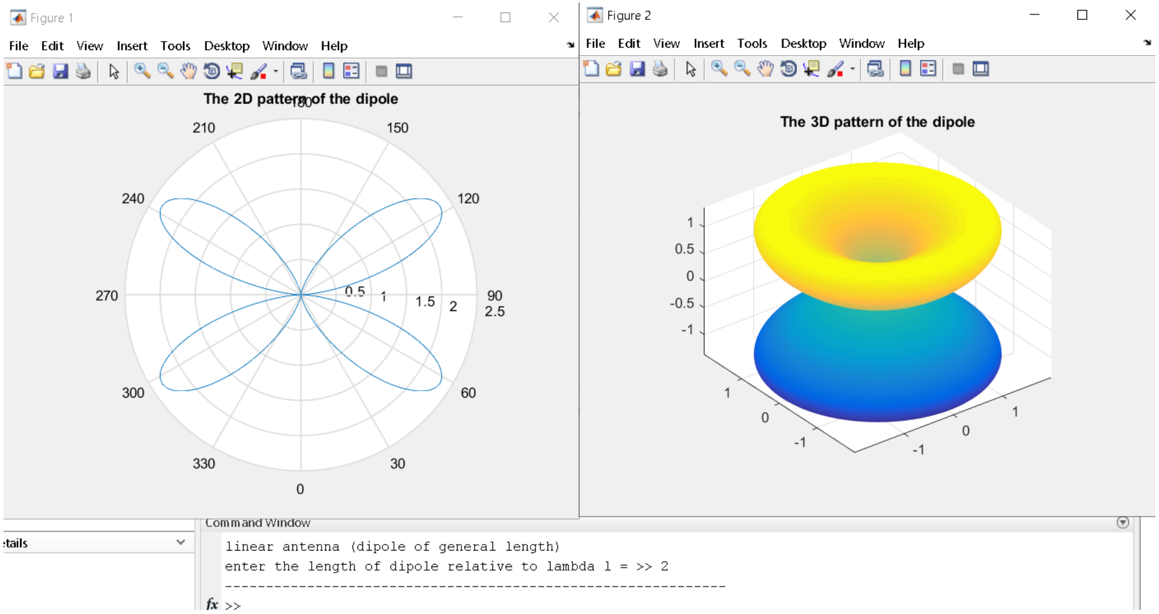


Figure 1
File Edit View Insert Tools Desktop Window Help
The 2D pattern of the dipole
180
210
150
240
120
270
90
2.5
0.5
1
1.5
2
300
60
330
30
0
Figure 2
File Edit View Insert Tools Desktop Window Help
The 3D pattern of the dipole
1
0.5
0
-0.5
-1
1
0
-1
1
0
-1
Command Window
tails
fx >>

```
linear antenna (dipole of general length)
enter the length of dipole relative to lambda l = >> 2
-------------------------------------------------------------
```

Figure 7: Example 2 - Part 1: linear antenna (dipole of general length).

Example 3: $Errors$

$if\ l < 0$, we will ask you again $l = ?$, then I put it = 0.5.

And what happen if you put $l = 0$

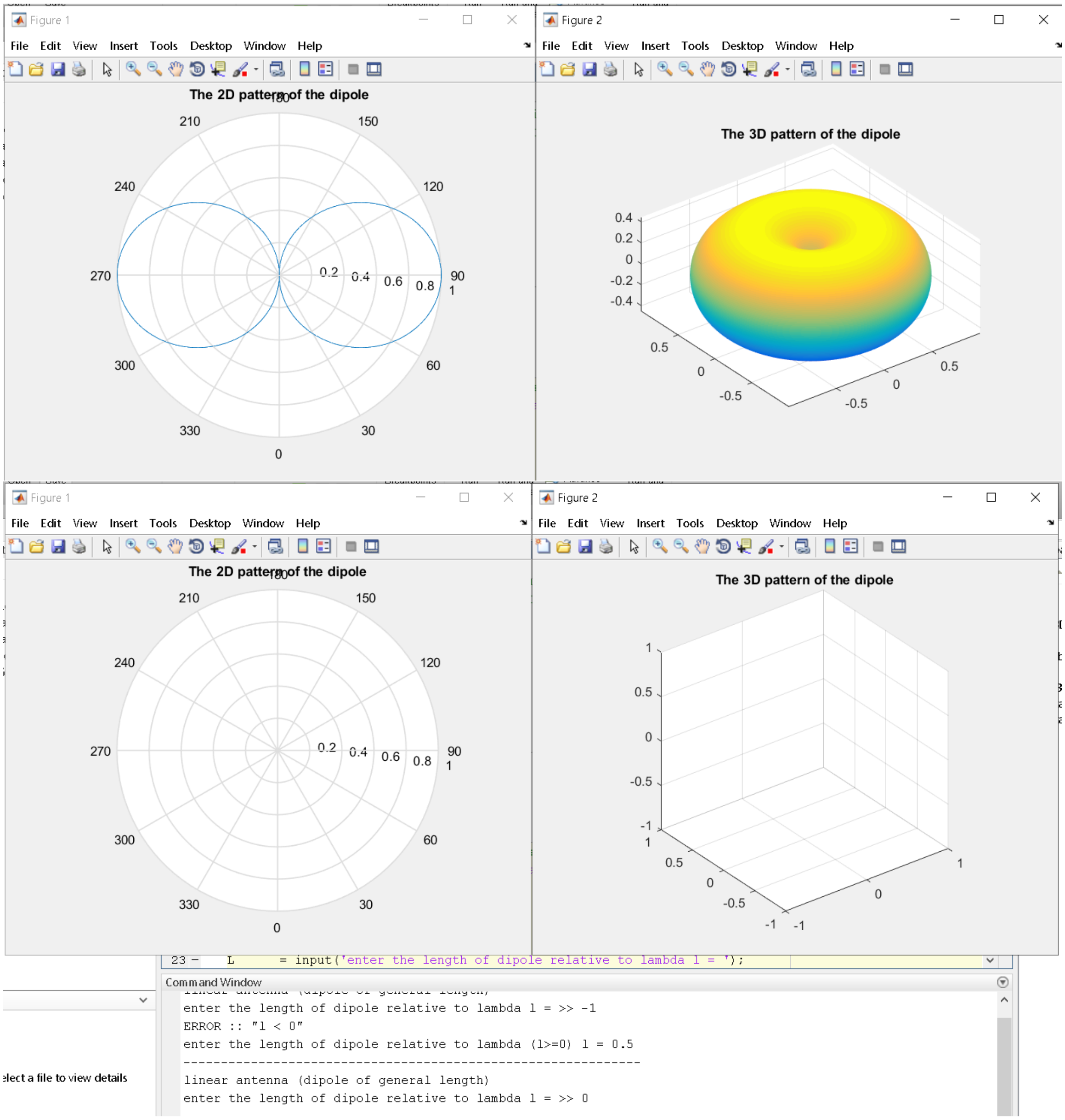


Figure 1
File Edit View Insert Tools Desktop Window Help
The 2D pattern of the dipole
180
210
150
240
120
270
90
300
60
330
30
0
0.2
0.4
0.6
0.8
1
Figure 2
The 3D pattern of the dipole
0.4
0.2
0
-0.2
-0.4
0.5
0
-0.5
0.5
0
-0.5
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
-1
0
1
Command Window
elect a file to view details

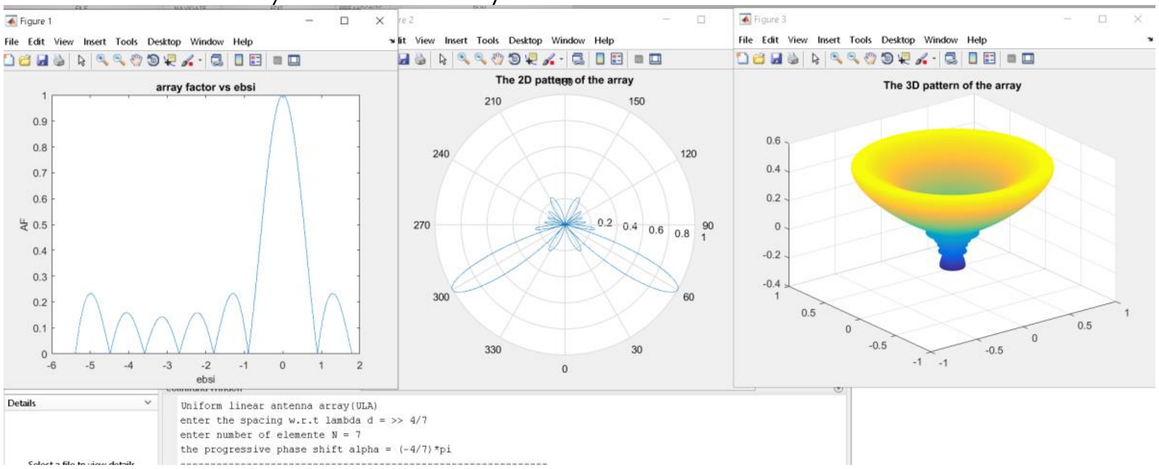
```
L        = input('enter the length of dipole relative to lambda l = ');
```

```
linear antenna (dipole of general length)
enter the length of dipole relative to lambda l = >> -1
ERROR :: "l < 0"
enter the length of dipole relative to lambda (l>=0) l = 0.5
------------------------------------------------------------
linear antenna (dipole of general length)
enter the length of dipole relative to lambda l = >> 0
```

Figure 8: Example 3 (Errors) - Part 1: linear antenna (dipole of general length).

2.2. Part 2: Uniform linear antenna array (ULA)

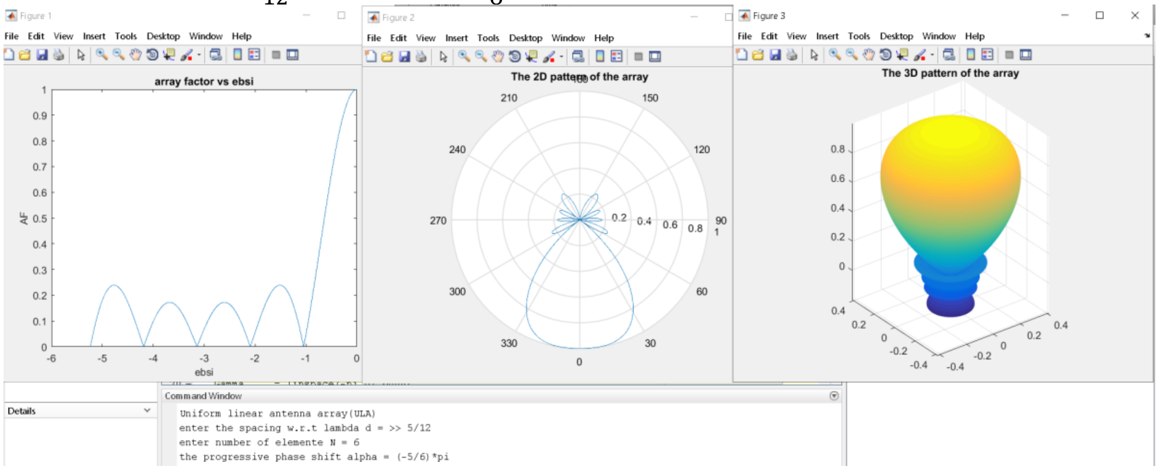
Example 1: $d = \frac{4\lambda}{7}, \mathrm{N} = 7, \alpha = \frac{-4\pi}{7}$



```
Uniform linear antenna array(ULA)
enter the spacing w.r.t lambda d = >> 4/7
enter number of elemente N = 7
the progressive phase shift alpha = (-4/7)*pi
```

Figure 9: Example 1 - Part 2: Uniform linear antenna array (ULA).

Example 2: $d = \frac{5\lambda}{12}, \mathrm{N} = 6, \alpha = \frac{-5\pi}{6}$



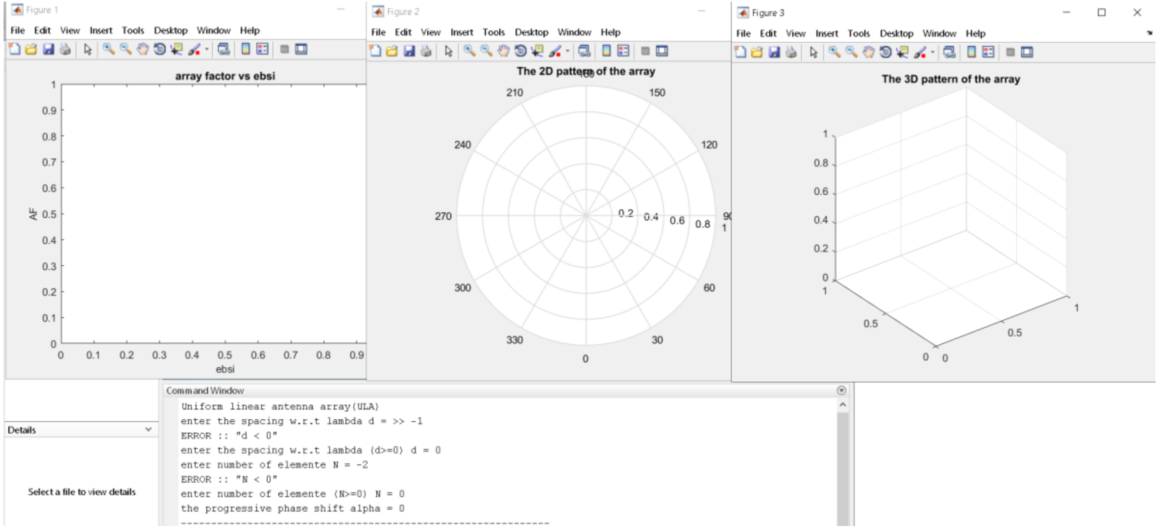
```
Uniform linear antenna array(ULA)
enter the spacing w.r.t lambda d = >> 5/12
enter number of elemente N = 6
the progressive phase shift alpha = (-5/6)*pi
```

Figure 10: Example 2 - Part 2: Uniform linear antenna array (ULA).

Example 3: $Errors$

$if\ d < 0\ or\ N < 0$, we will ask you again.

And what happen if you put $all\ pramaters = 0$



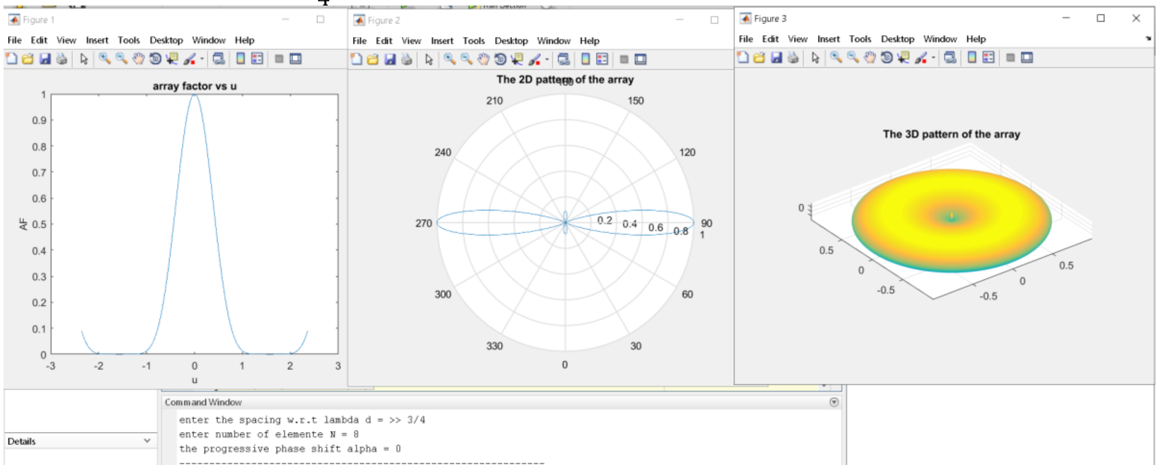
```
Uniform linear antenna array(ULA)
enter the spacing w.r.t lambda d = >> -1
ERROR :: "d < 0"
enter the spacing w.r.t lambda (d>=0) d = 0
enter number of elemente N = -2
ERROR :: "N < 0"
enter number of elemente (N>=0) N = 0
the progressive phase shift alpha = 0
-------------------------------------------------------------
```

Figure 11: Example 3 (Errors) - Part 2: Uniform linear antenna array (ULA).

2.3. Part 3: Nonuniformly-Fed linear antenna array

A. Binomial Arrays

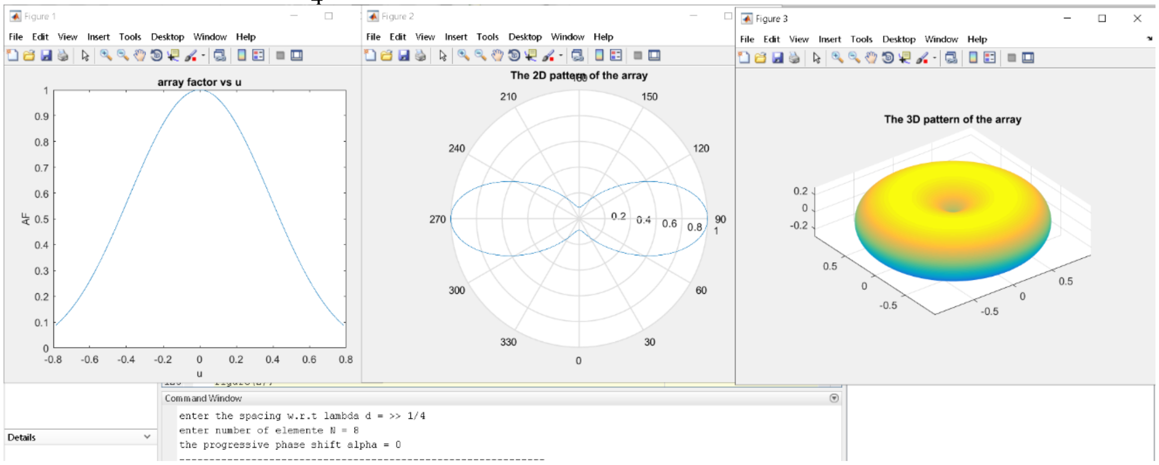
Example 1: $d = \frac{3\lambda}{4}, \mathrm{N} = 8, \alpha = 0$



```
enter the spacing w.r.t lambda d = >> 3/4
enter number of elemente N = 8
the progressive phase shift alpha = 0
-------------------------------------------------------------
```

Figure 12: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

Example 2: $d = \frac{\lambda}{4}, \mathrm{N} = 8, \alpha = 0$



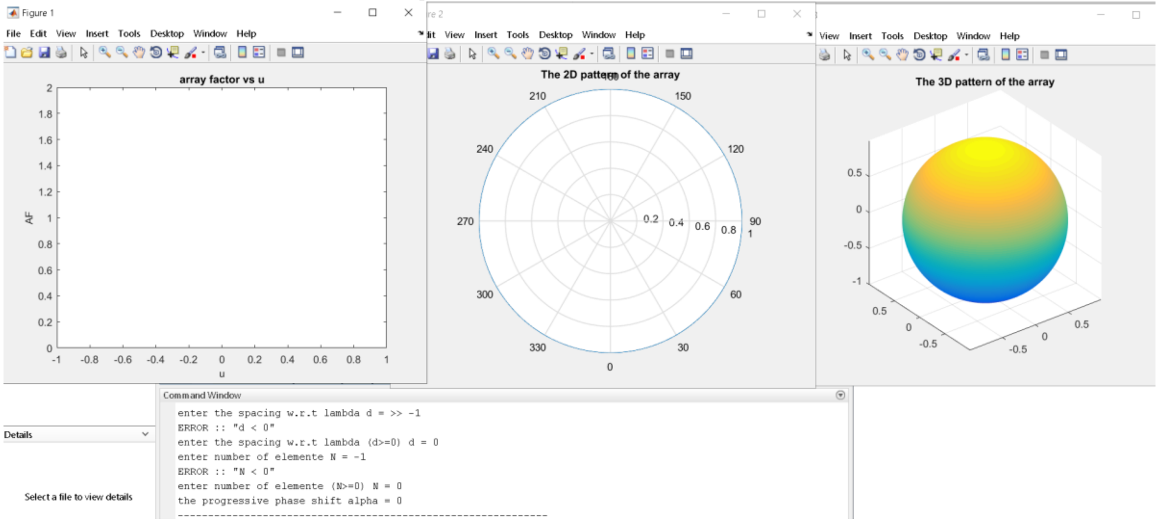
```
enter the spacing w.r.t lambda d = >> 1/4
enter number of elemente N = 8
the progressive phase shift alpha = 0
--------------------------------------------------------------
```

Figure 13: Example 2 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

Example 3: $Errors$

$if\ d < 0\ or\ N < 0$, we will ask you again.

And what happen if you put $all\ pramaters = 0$

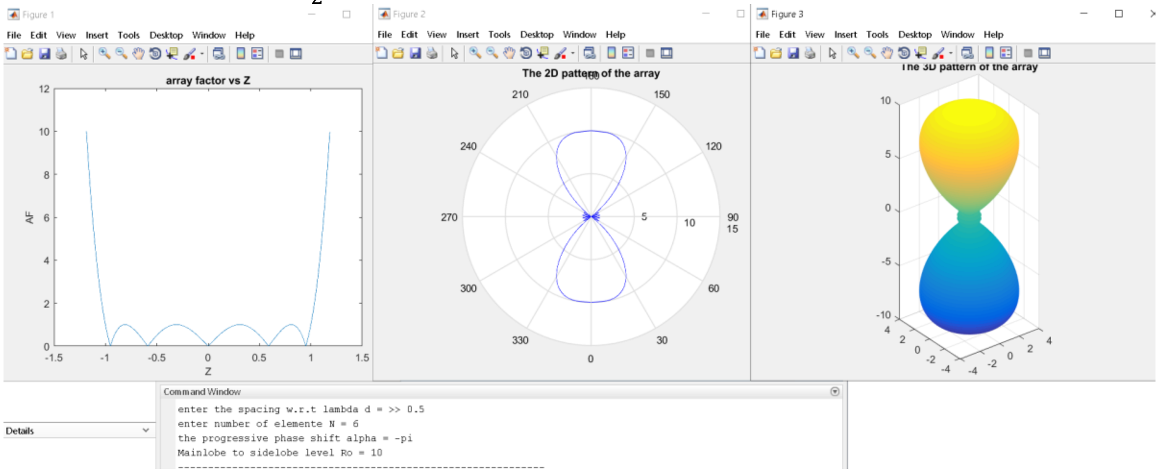


```
enter the spacing w.r.t lambda d = >> -1
ERROR :: "d < 0"
enter the spacing w.r.t lambda (d>=0) d = 0
enter number of elemente N = -1
ERROR :: "N < 0"
enter number of elemente (N>=0) N = 0
the progressive phase shift alpha = 0
--------------------------------------------------------------
```

Figure 14: Example 3 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

B. Dolph-Tschebysceff Arrays

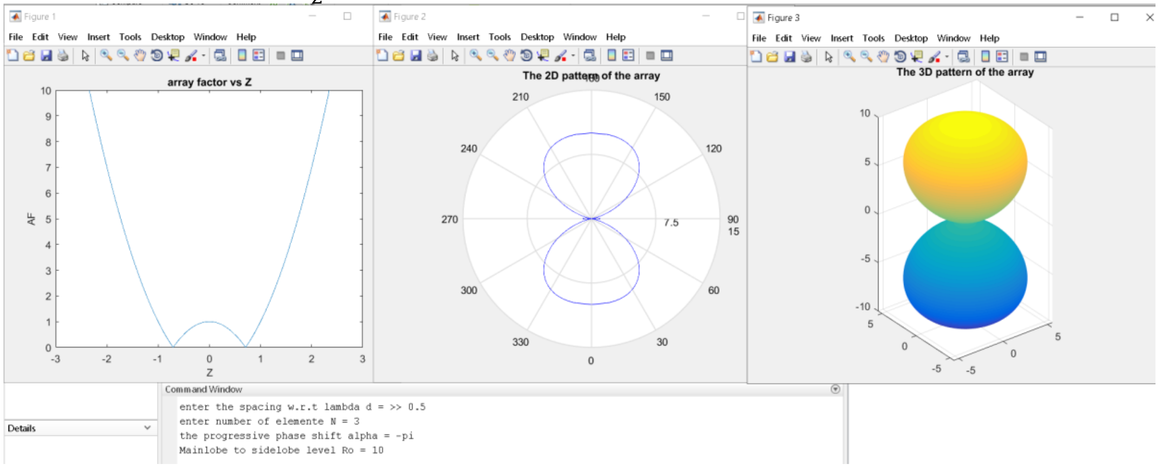
Example 1: $d = \frac{\lambda}{2}, \mathrm{N} = 6, \alpha = -\pi, R_0 = 10.$



```
enter the spacing w.r.t lambda d = >> 0.5
enter number of elemente N = 6
the progressive phase shift alpha = -pi
Mainlobe to sidelobe level Ro = 10
-------------------------------------------------------------
```

Figure 15: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

Example 2: $d = \frac{\lambda}{2}, \mathrm{N} = 3, \alpha = -\pi, R_0 = 10.$



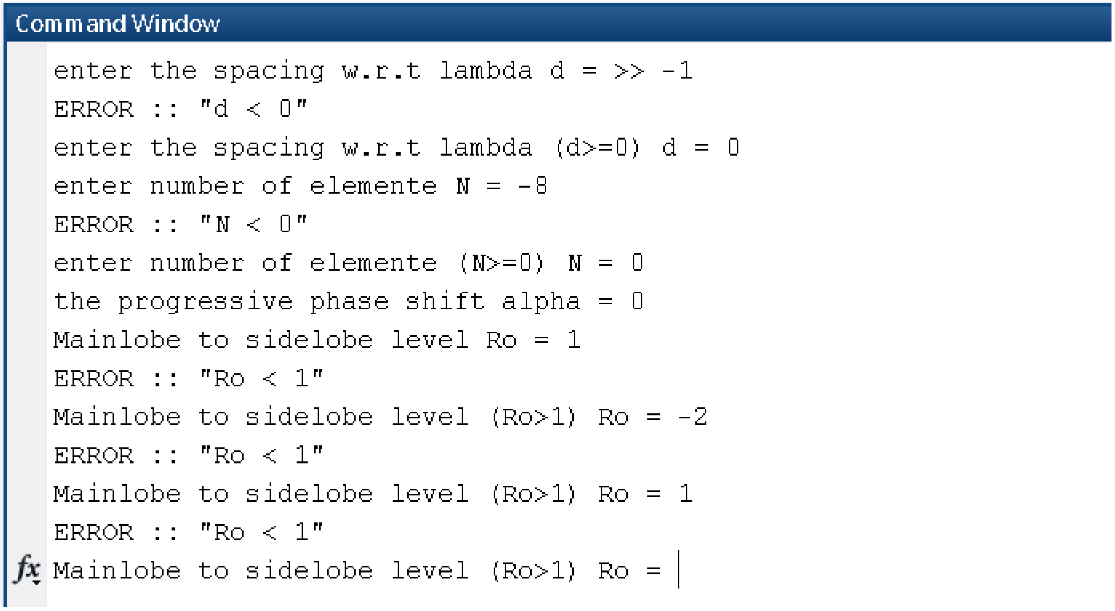
```
enter the spacing w.r.t lambda d = >> 0.5
enter number of elemente N = 3
the progressive phase shift alpha = -pi
Mainlobe to sidelobe level Ro = 10
```

Figure 16: Example 2 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

Example 3: $Errors$

$if\ d < 0\ or\ N < 0$, we will ask you again.

$Always\ R_0 > 1$.



```
enter the spacing w.r.t lambda d = >> -1
ERROR :: "d < 0"
enter the spacing w.r.t lambda (d>=0) d = 0
enter number of elemente N = -8
ERROR :: "N < 0"
enter number of elemente (N>=0) N = 0
the progressive phase shift alpha = 0
Mainlobe to sidelobe level Ro = 1
ERROR :: "Ro < 1"
Mainlobe to sidelobe level (Ro>1) Ro = -2
ERROR :: "Ro < 1"
Mainlobe to sidelobe level (Ro>1) Ro = 1
ERROR :: "Ro < 1"
fx Mainlobe to sidelobe level (Ro>1) Ro =
```

Figure 17: Example 3 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

3. Copy of Code

```
%%
% Alexandria University - Faculty of Engineering
% Electrical and Electronic Engineering Department -
Fourth Year - Communications & Electronics
%
% Course: Antenna Lab
% Antenna Lab Assignment-2
%
% Name    : Mahmoud Mohamed Kamal Ismail - Mahmoud
AbdElHady Mahmoud - Ahmed Mohamed Abdelakher
% Section : 7                           - 7
- 1
% Seat No.: 250                         - 248
- 37

%%
clear;
close all;
clc;

%% Part (1): linear antenna (dipole of general length)
fprintf('linear antenna (dipole of general length)\n');
Lambda = 1;
B      = (2*pi)/Lambda;
Theta  = linspace(-pi,pi,350);
Phi    = linspace(-2*pi,2*pi,350);
L      = input('enter the length of dipole relative to
lambda l = ');
while L < 0
    fprintf('ERROR :: "l < 0"\n');
    L  = input('enter the length of dipole relative to
lambda (l>=0) l = ');
end
L      = L * Lambda;
En     = abs((cos(((B*L)/2).*cos(Theta)) -
cos((B*L)/2)) ./ sin(Theta));

figure(1);
polar(Theta,En);
view([90 90]);
title('The 2D pattern of the dipole');
```

```
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('The 3D pattern of the dipole');

fprintf('----------------------------------------------
---------------\n');

%% Part (2): Uniform linear antenna array(ULA)
fprintf('Uniform linear antenna array(ULA)\n');
Lambda     = 1;
B          = (2*pi)/Lambda;
d          = input('enter the spacing w.r.t lambda d = 
');
while d < 0
    fprintf('ERROR :: "d < 0"\n');
    d      = input('enter the spacing w.r.t lambda 
(d>=0) d = ');
end
d          = d * Lambda;
N          = input('enter number of elemente N = ');
while N < 0
    fprintf('ERROR :: "N < 0"\n');
    N      = input('enter number of elemente (N>=0) N = 
'); % If N = 0 there's no array
end
alpha      = input('the progressive phase shift alpha = 
');
max_angle = acos(-alpha/(B*d));
Gamma      = linspace(-pi,pi,6000);
Phi        = linspace(-2*pi,2*pi,6000);
ebsi       = B*d*cos(Gamma) + alpha;
AF         = abs(sin((N*ebsi)/2) ./ (N * sin(ebsi/2)));
```

```
figure(1);
plot(ebsi,AF);
title('array factor vs ebsi');
xlabel('ebsi','fontsize',10);
ylabel('AF','fontsize',10);

figure(2);
polar(Gamma,AF);
view([90 90]);
title('The 2D pattern of the array');

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;
lighting gouraud;
title('The 3D pattern of the array');

fprintf('----------------------------------------------
---------------\n');

%% Part (3): Nonuniformly-Fed linear antenna array
%  A. Binomial Arrays
Lambda = 1;
B      = (2*pi)/Lambda;
d      = input('enter the spacing w.r.t lambda d = ');
while d < 0
    fprintf('ERROR :: "d < 0"\n');
    d     = input('enter the spacing w.r.t lambda
(d>=0) d = ');
end
d          = d * Lambda;
N          = input('enter number of elemente N = ');
while N < 0
    fprintf('ERROR :: "N < 0"\n');
    N     = input('enter number of elemente (N>=0) N =
'); % If N = 0 there's no array
end
```

```
alpha  = input('the progressive phase shift alpha = ');
Theta  = linspace(-pi,pi,6000);
Phi    = linspace(-2*pi,2*pi,6000);
u      = (B*d*cos(Theta) + alpha)/2;
AF     = abs(cos(u).^(N-1));

figure(1);
plot(u,AF);
title('array factor vs u');
xlabel('u','fontsize',10);
ylabel('AF','fontsize',10);

figure(2);
polar(Theta,AF);
view([90 90]);
title('The 2D pattern of the array');

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('The 3D pattern of the array');

fprintf('---------------------------------------------
---------------\n');

%% Part (3): Nonuniformly-Fed linear antenna array
%  B. Dolph-Tschebysceff Arrays
Lambda = 1;
B      = (2*pi)/Lambda;
d      = input('enter the spacing w.r.t lambda d = ');
while d < 0
    fprintf('ERROR :: "d < 0"\n');
    d  = input('enter the spacing w.r.t lambda (d>=0) d
= ');
```

```
end
d       = d * Lambda;
N       = input('enter number of elemente N = ');
while N < 0
    fprintf('ERROR :: "N < 0"\n');
    N   = input('enter number of elemente (N>=0) N = ');
% If N = 0 there's no array
end
M       = N - 1;
alpha   = input('the progressive phase shift alpha = ');
Ro      = input('Mainlobe to sidelobe level Ro = ');
while Ro <= 1
    fprintf('ERROR :: "Ro < 1"\n');
    Ro = input('Mainlobe to sidelobe level (Ro>1) Ro =
');
end
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)/(B*d));
Theta2 = -Theta1;
Phi     = linspace(-2*pi,2*pi,6000);
AF      = abs(cosh(M.*acosh(Z)));

figure(1);
plot(Z,AF);
title('array factor vs Z');
xlabel('Z','fontsize',10);
ylabel('AF','fontsize',10);

figure(2);
polar(Theta1,AF,'-b');
hold on;
polar(Theta2,AF,'-b');
view([90 90]);
title('The 2D pattern of the array');

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('The 3D pattern of the array');

fprintf('----------------------------------------------
---------------\n');
```