MATLAB Assignment: Radiation Pattern Analysis of Linear Antenna Array

Sourour Hammoud
2254

November 20, 2025

1 Objective

The objective of this assignment is to simulate and analyze the radiation patterns of linear antenna arrays, specifically broadside and end-fire configurations. We investigate how the number of elements (N) and element spacing (d/λ) affect the radiation characteristics, directivity, and beamwidth of these arrays.

2 Theory

2.1 Array Factor

For a linear array of N isotropic elements with uniform amplitude and progressive phase shift, the array factor is given by:

$$AF(\theta) = \frac{\sin(N\psi/2)}{N\sin(\psi/2)} \tag{1}$$

where:

$$\psi = kd\cos\theta + \Phi \tag{2}$$

and:

- $k = 2\pi/\lambda$ (wave number)
- d = element spacing
- θ = observation angle from the array axis
- Φ = progressive phase shift between elements

2.2 Broadside Array

A broadside array has:

- Uniform spacing: $d = \lambda/2$
- Uniform phase excitation: $\Phi = 0$
- Maximum radiation perpendicular to the array axis ($\theta = 90$)
- Therefore: $\psi = \pi\cos(\theta)$

The array factor becomes:

$$F_{AFbroadside}(\theta) = \frac{\sin(N\pi\cos(\theta)/2)}{N\sin(\pi\cos(\theta)/2)} \tag{3}$$

2.3 End-Fire Array

An end-fire array has:

- Phase shift: $\Phi = -\pi/2$ (or equivalently $d = \lambda/4$)
- Maximum radiation along the array axis ($\theta = 0$ or 180)
- Progressive phase shift between elements
- Therefore: $\psi = \frac{\pi}{2}(\cos(\theta) - 1)$

  The array factor becomes:

$$F_{AFendfire}(\theta) = \frac{\sin(N\pi(\cos(\theta) - 1)/2)}{N\sin(\pi(\cos(\theta) - 1)/2)} \tag{4}$$

3 Broadside Array Analysis

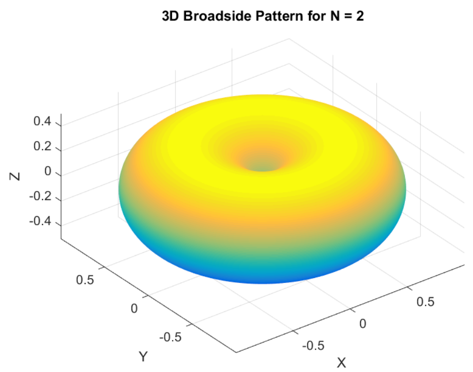
3.1 Varying Number of Elements (N) with Fixed $d/\lambda = 0.5$

3.1.1 Theory

With fixed spacing at half-wavelength ($d/\lambda = 0.5$), increasing the number of elements:

- Narrows the main beam (reduces beamwidth)
- Increases directivity
- Creates more nulls and side lobes
- Maintains the broadside radiation pattern

3.1.2 Radiation Patterns



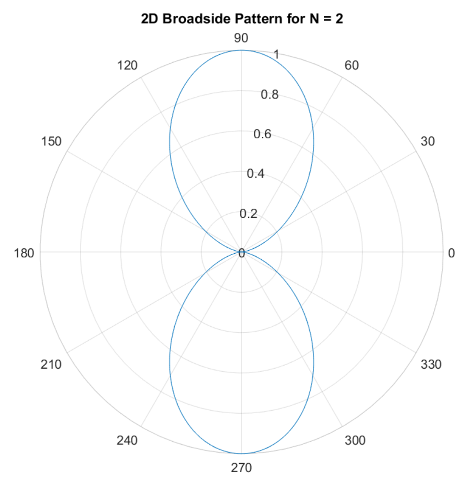


Figure 1: Broadside array radiation patterns for N=2

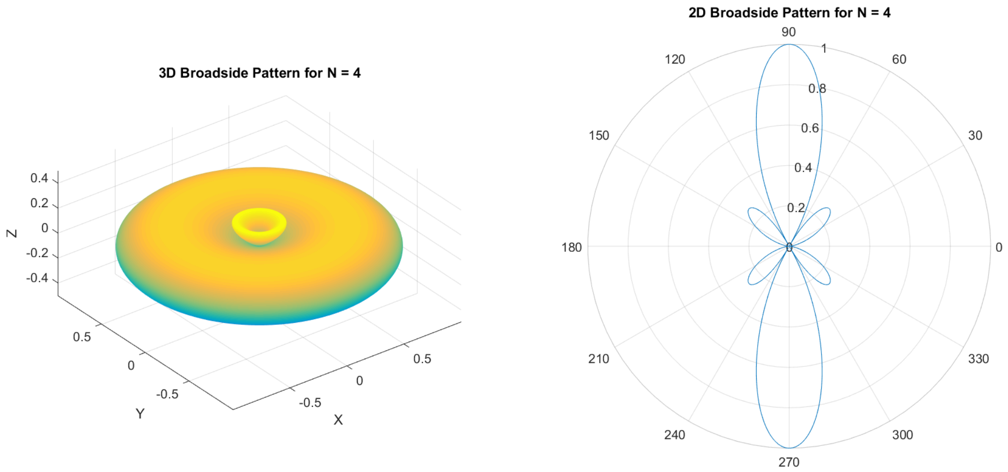


Figure 2: Broadside array radiation patterns for N=4

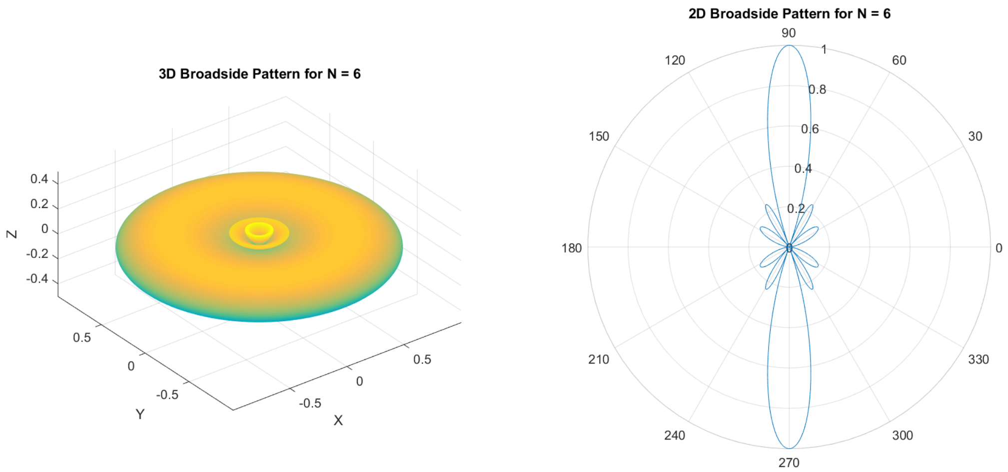


Figure 3: Broadside array radiation patterns for N=6

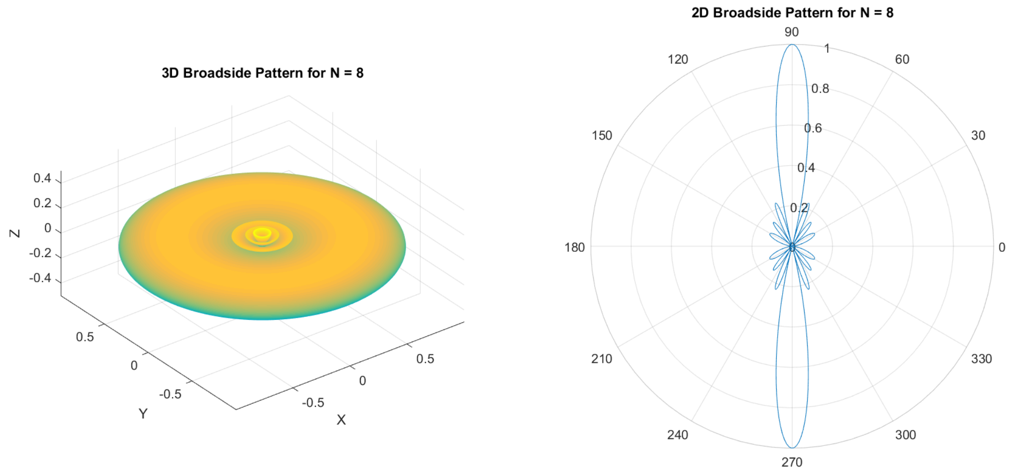


Figure 4: Broadside array radiation patterns for N=8

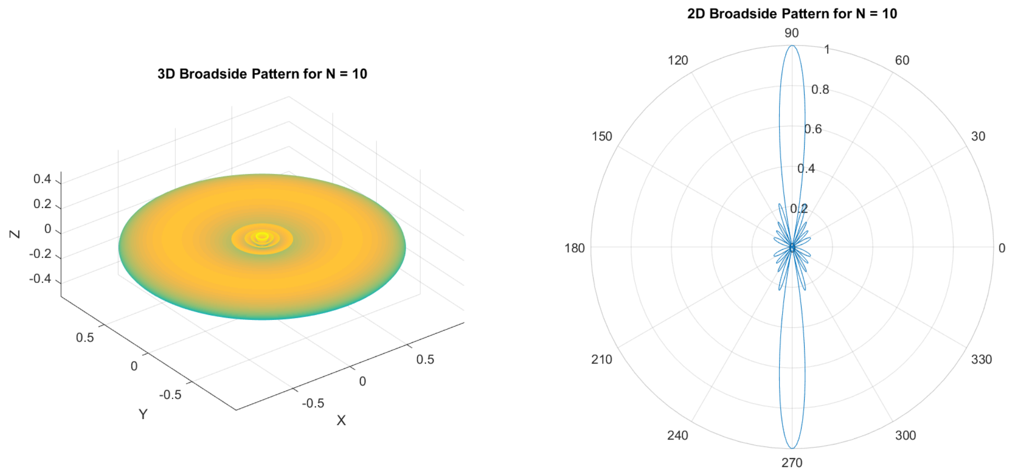


Figure 5: Broadside array radiation patterns for N=10

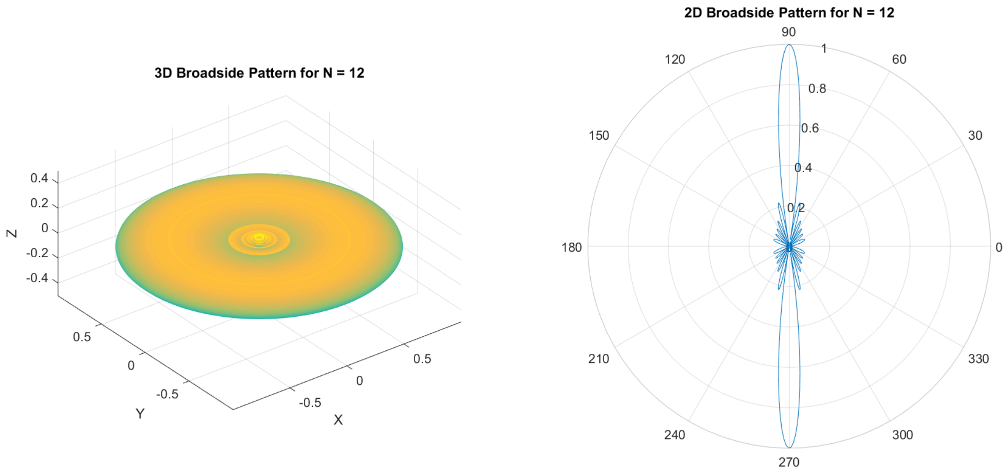


Figure 6: Broadside array radiation patterns for N=12

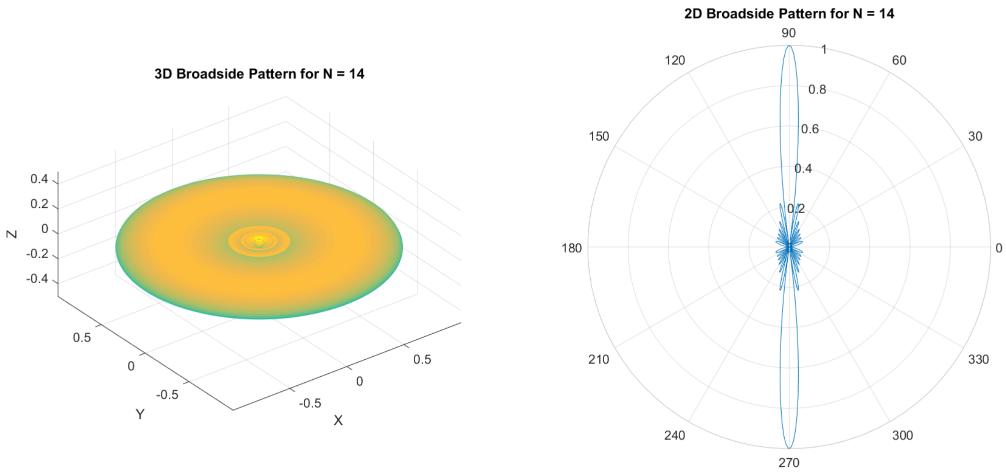


Figure 7: Broadside array radiation patterns for N=14

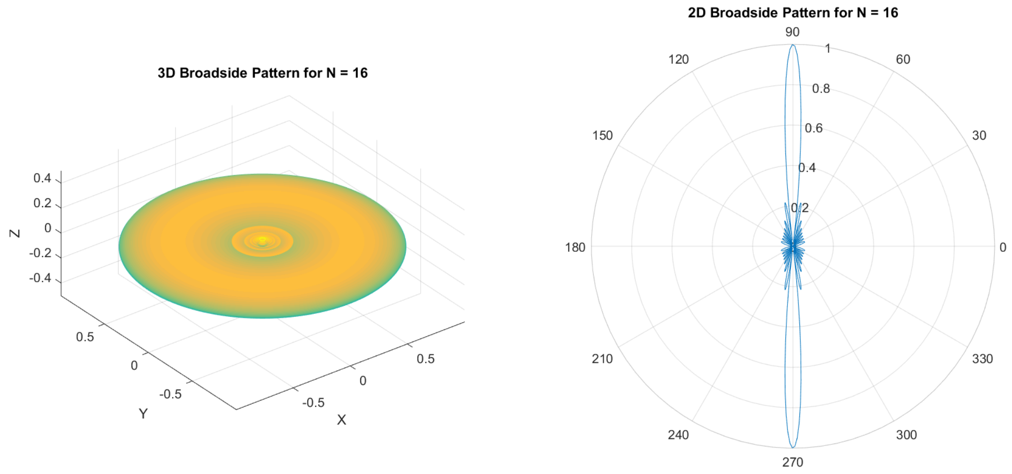


Figure 8: Broadside array radiation patterns for N=16

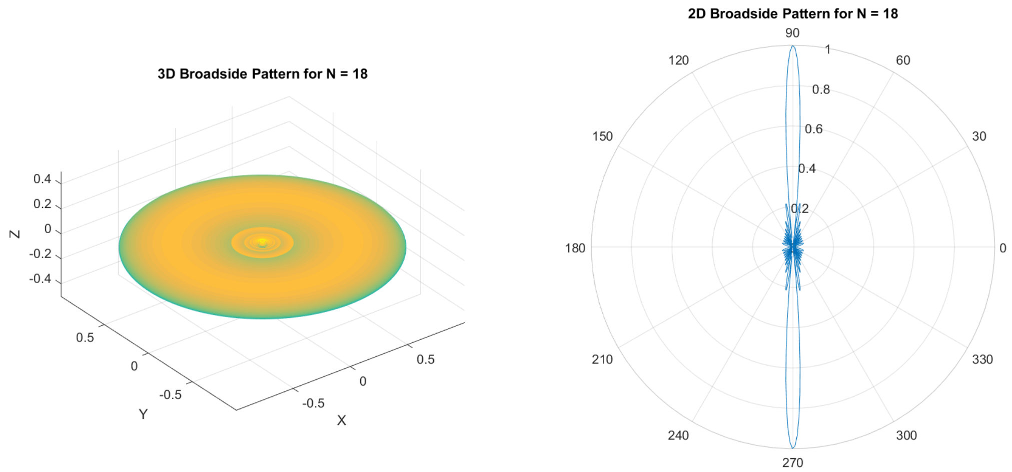


Figure 9: Broadside array radiation patterns for N=18

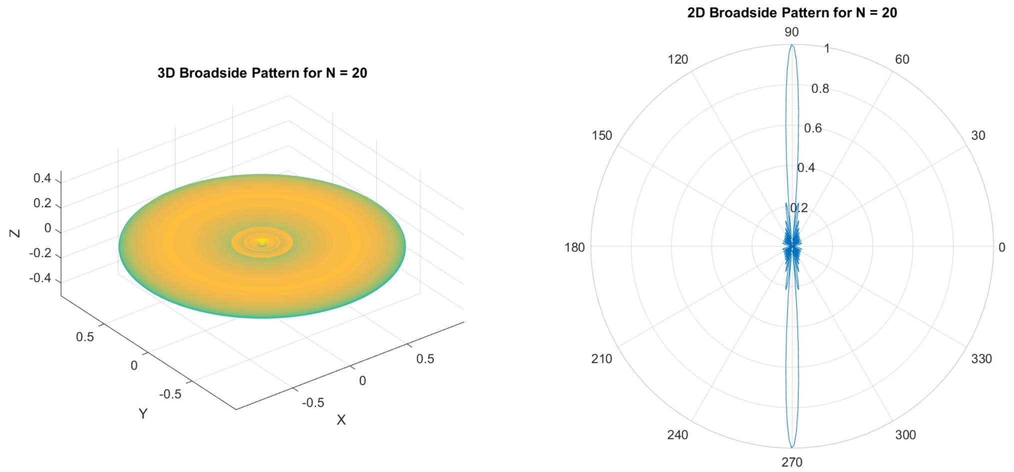


Figure 10: Broadside array radiation patterns for N=20

Analysis:

- $N = 1$: Single omnidirectional element
- $N = 2 - 5$: Beam begins to narrow, first side lobes appear
- $N = 8 - 12$: Well-defined main beam with multiple side lobes
- $N = 15 - 20$: Very narrow main beam, high directivity, many side lobes

3.1.3 MATLAB Code

```
% Take d = \lambda/2 > \beta * d = pi
% We vary N
theta = linspace(0, 2*pi, 500); % theta from 0 to 2*pi
phi = linspace(0, 2*pi, 500); % phi from 0 to 2*pi
[Theta, Phi] = meshgrid(theta, phi);
beta_d = pi;
N = 2:2:20;
for i = 1:length(N)
% Array factor calculation for broadside array
numerator_3D = sin(N(i) * 0.5 * beta_d * cos(Theta));
denominator_3D = N(i) * sin(0.5 * beta_d * cos(Theta));
numerator = sin(N(i) * 0.5 * beta_d * cos(theta));
denominator = N(i) * sin(0.5 * beta_d * cos(theta));
F_AF_3D = numerator_3D ./ denominator_3D;
F_AF = numerator ./ denominator;
% Convert to Cartesian coordinates for 3D plotting
X = abs(F_AF_3D) .* sin(Theta) .* cos(Phi);
Y = abs(F_AF_3D) .* sin(Theta) .* sin(Phi);
Z = abs(F_AF_3D) .* cos(Theta);
% Create a new figure for both 3D and 2D plots
```

```
fig = figure;
% Plot the radiation pattern in 3D
subplot(1, 2, 1);
surf(X, Y, Z);
shading interp;
title(['3D␣Broadside␣Pattern␣for␣N␣=␣', num2str(N(i))]);
xlabel('X'); ylabel('Y'); zlabel('Z');
axis equal;
axis([-1 1 -1 1 -0.5 0.5] * max(abs(F_AF_3D(:)))); % Set axis limits tomake differences noticeable
% Plot the radiation pattern in 2D
subplot(1, 2, 2);
polarplot(theta, abs(F_AF)); % 2D pattern for a fixed phi slice
title(['2D␣Broadside␣Pattern␣for␣N␣=␣', num2str(N(i))]);
% Maximize the figure
set(fig, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
% Save the figure as a PNG file
saveas(fig, ['Broadside_Pattern_N_', num2str(N(i)), '.png']);
% Close the figure to avoid excessive open windows
close(fig);

end
```

Listing 1: Broadside array simulation with varying N

3.2 Varying Element Spacing (d/λ) with Fixed $N = 8$

3.2.1 Theory

With fixed number of elements ($N = 8$), varying the spacing affects:

- $d/\lambda < 0.5$: Wider beamwidth, lower directivity
- $d/\lambda = 0.5$: Optimal spacing for broadside, no grating lobes
- $d/\lambda > 0.5$: Narrower beam but grating lobes may appear
- $d/\lambda \geq 1$: Significant grating lobes (undesirable)

3.2.2 Radiation Patterns

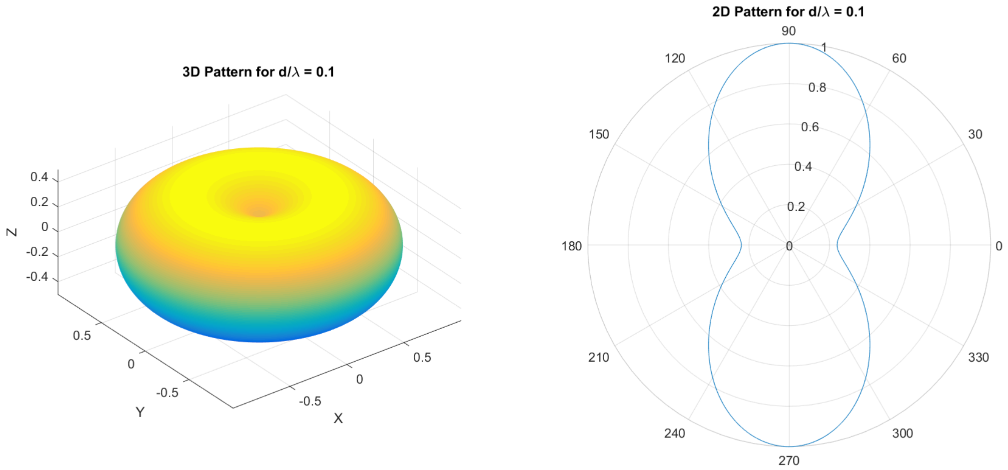


Figure 11: Broadside array radiation patterns for varying d/λ=0.1

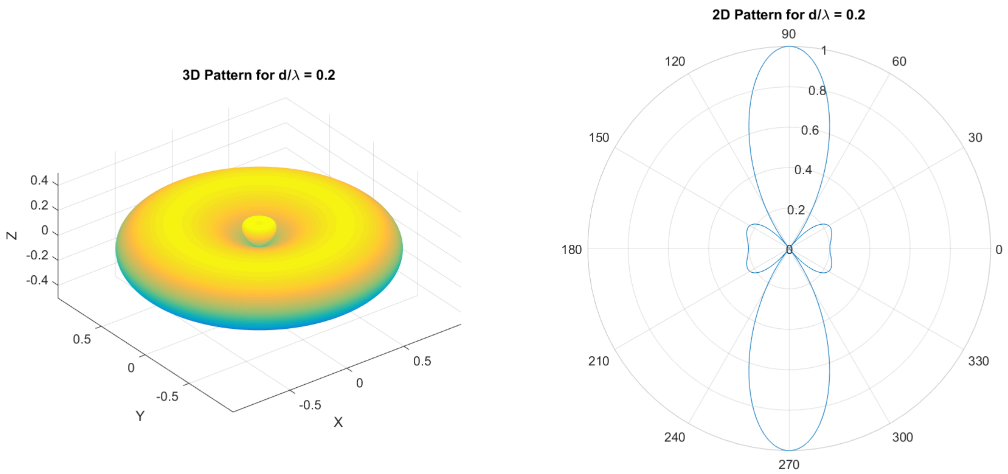


Figure 12: Broadside array radiation patterns for varying d/λ=0.2

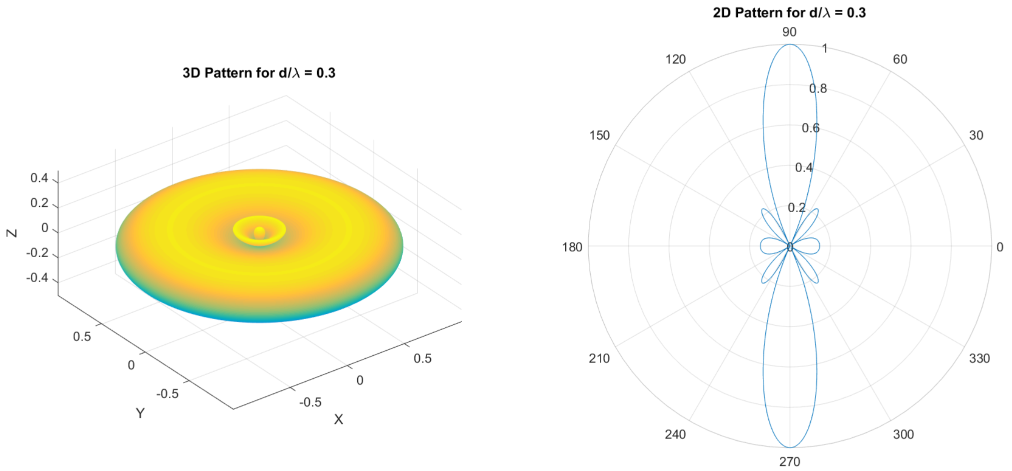


Figure 13: Broadside array radiation patterns for varying d/λ=0.3

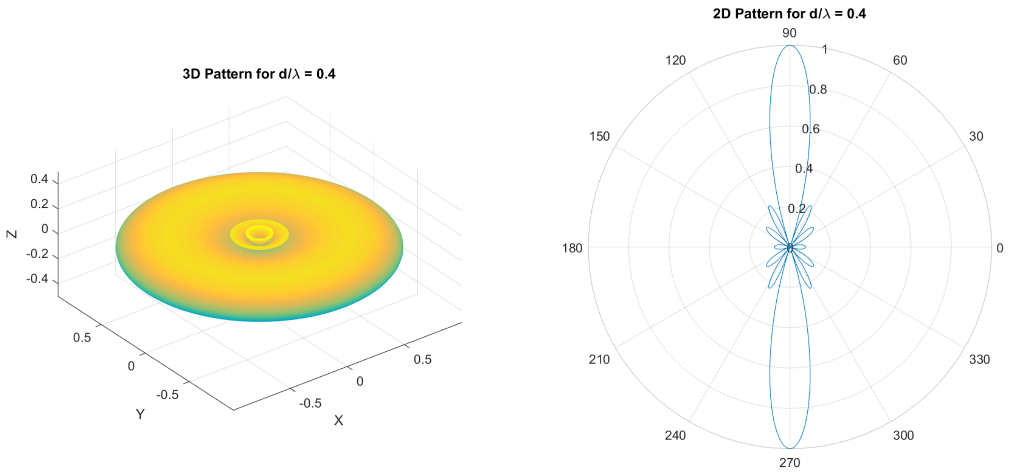


Figure 14: Broadside array radiation patterns for varying d/λ=0.4

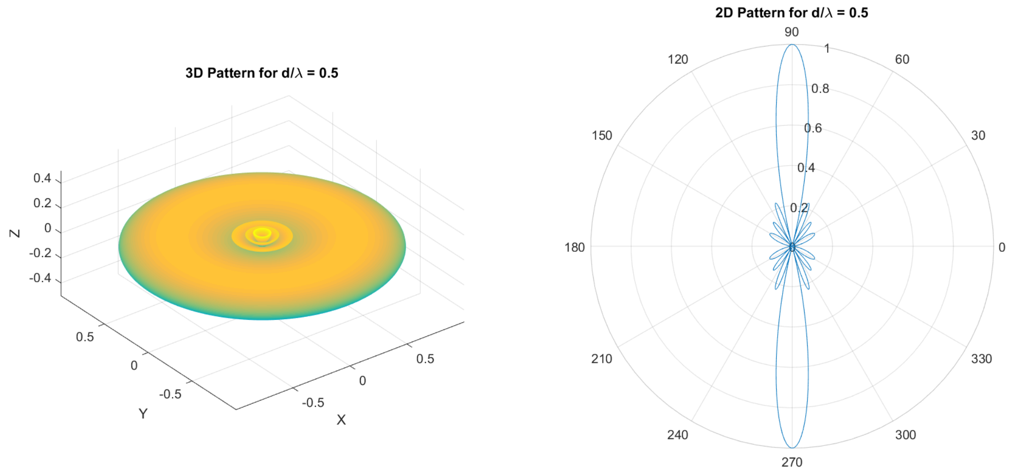


Figure 15: Broadside array radiation patterns for varying d/λ=0.5

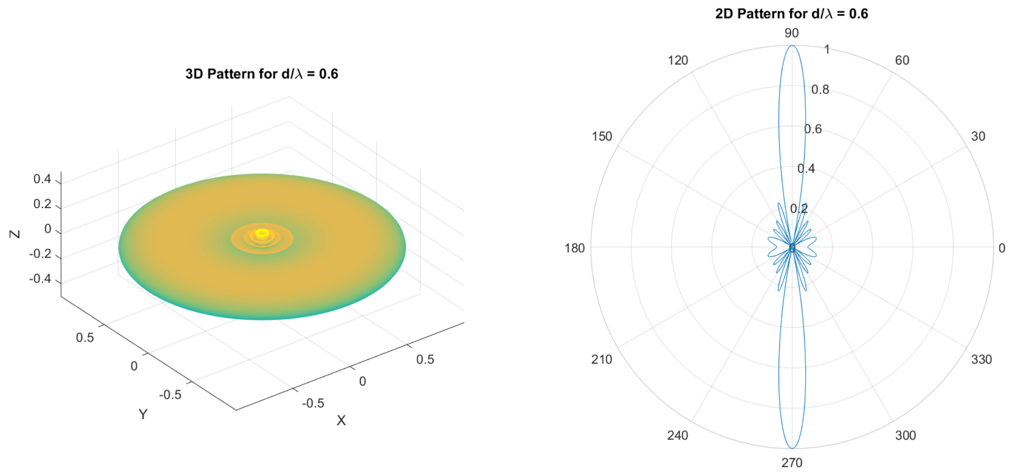


Figure 16: Broadside array radiation patterns for varying d/λ=0.6

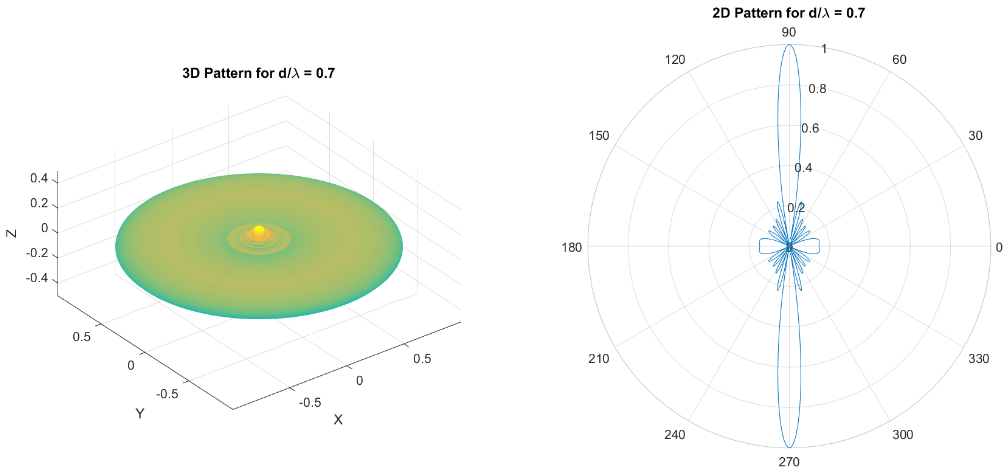


Figure 17: Broadside array radiation patterns for varying d/λ=0.7

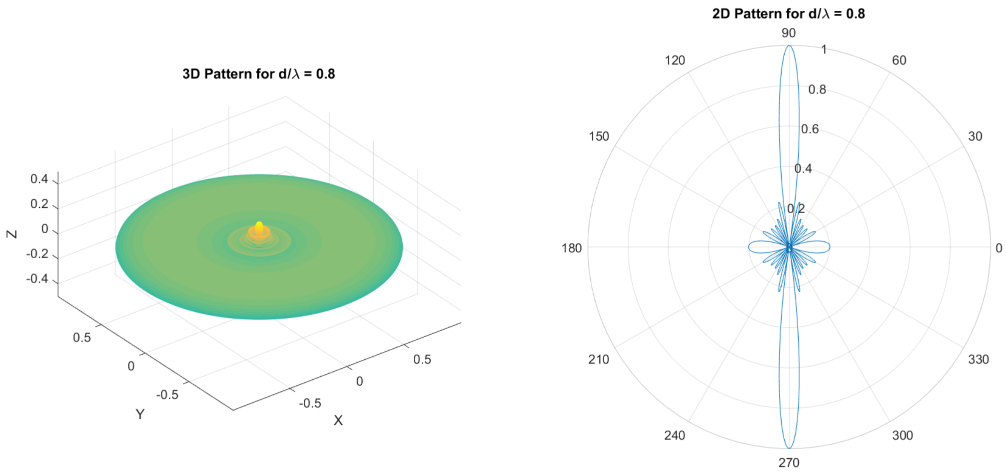


Figure 18: Broadside array radiation patterns for varying d/λ=0.8

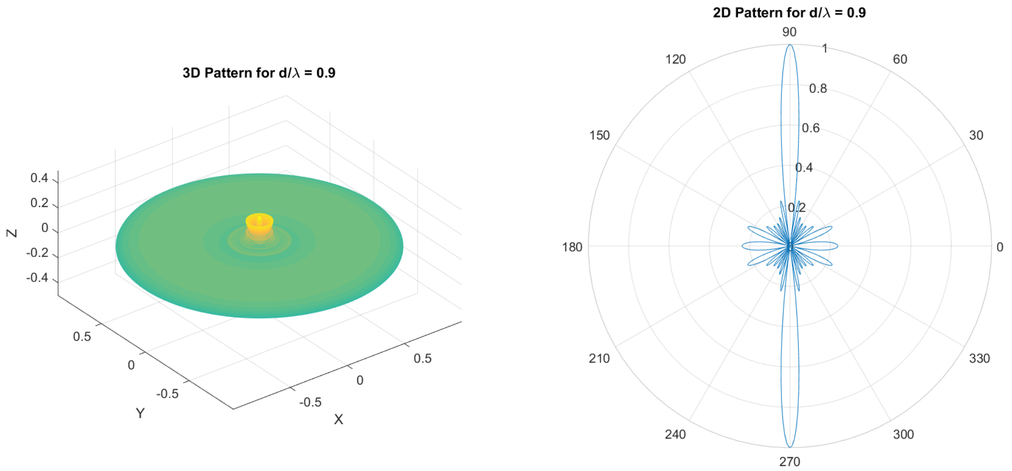


Figure 19: Broadside array radiation patterns for varying d/λ=0.9

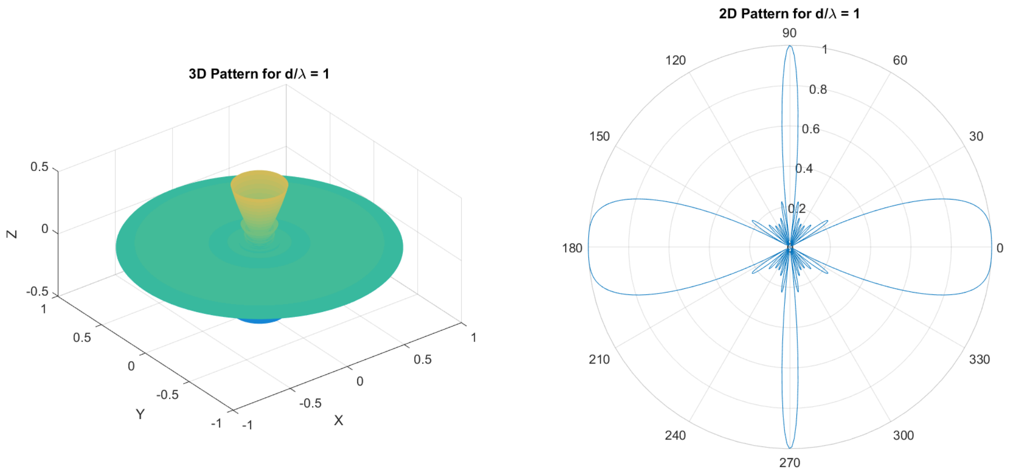


Figure 20: Broadside array radiation patterns for varying d/λ=1

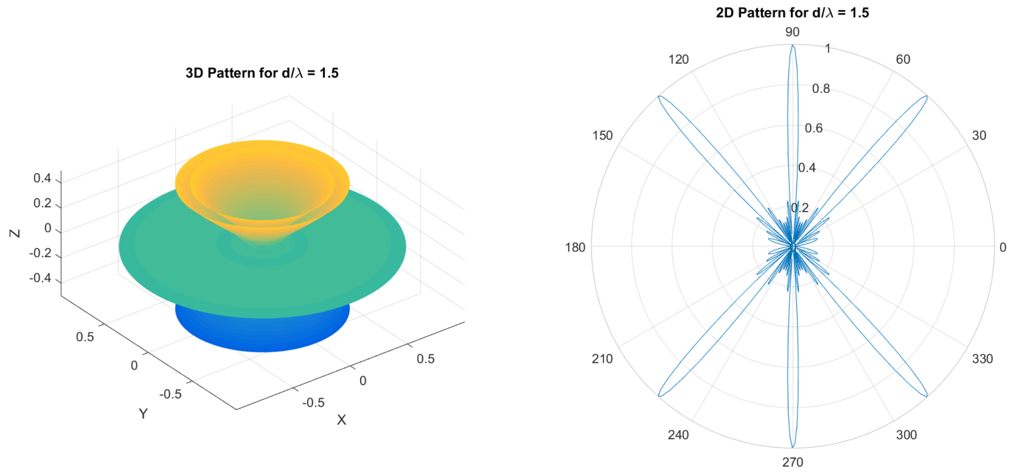


Figure 21: Broadside array radiation patterns for varying d/λ=1.5

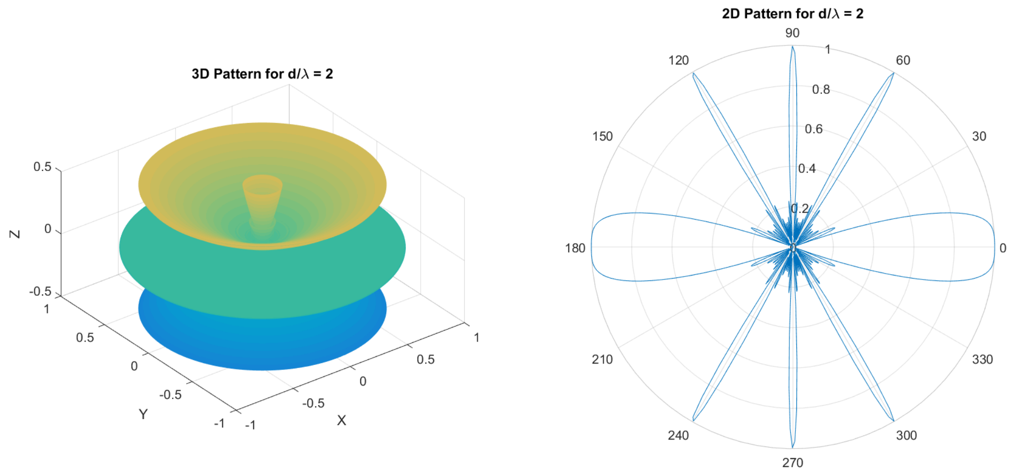


Figure 22: Broadside array radiation patterns for varying d/λ=2

Analysis:

- $d/\lambda = 0.1 - 0.3$: Very wide beam, low directivity
- $d/\lambda = 0.4 - 0.6$: Optimal broadside performance
- $d/\lambda = 0.7 - 0.9$: Beam narrowing continues
- $d/\lambda = 1.0 - 2.0$: Grating lobes emerge, pattern deteriorates

3.2.3 MATLAB Code

```
% Take N = 8, vary d / lambda
theta = linspace(0, 2*pi, 500); % theta from 0 to 2*pi
phi = linspace(0, 2*pi, 500); % phi from 0 to 2*pi
[Theta, Phi] = meshgrid(theta, phi);
N = 8;
d_lambda_values = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,
   1, 1.5, 2]; % Vary d/lambda from 0.1 to 1
for k = 1:length(d_lambda_values)
% Calculate beta_d from d/lambda
d_lambda = d_lambda_values(k);
beta_d = 2 * pi * d_lambda; % Beta * d = 2 * pi * (d / lambda)
% Array factor calculation for broadside array
numerator_3D = sin(N * 0.5 * beta_d * cos(Theta));
denominator_3D = N * sin(0.5 * beta_d * cos(Theta));
numerator = sin(N * 0.5 * beta_d * cos(theta));
denominator = N * sin(0.5 * beta_d * cos(theta));
F_AF_3D = numerator_3D ./ denominator_3D;
F_AF = numerator ./ denominator;
% Convert to Cartesian coordinates for 3D plotting
X = abs(F_AF_3D) .* sin(Theta) .* cos(Phi);
Y = abs(F_AF_3D) .* sin(Theta) .* sin(Phi);
Z = abs(F_AF_3D) .* cos(Theta);
% Create a new figure for both 3D and 2D plots
fig = figure;
% Plot the radiation pattern in 3D
subplot(1, 2, 1);
surf(X, Y, Z);
shading interp;
title(['3D Pattern for d/\lambda = ', num2str(d_lambda)]);
xlabel('X'); ylabel('Y'); zlabel('Z');
axis equal;
axis([-1 1 -1 1 -0.5 0.5] * max(abs(F_AF_3D(:)))); % Set axis
   limits to make differences noticeable
% Plot the radiation pattern in 2D
subplot(1, 2, 2);
polarplot(theta, abs(F_AF)); % 2D pattern for a fixed phi slice
title(['2D Pattern for d/\lambda = ', num2str(d_lambda)]);
% Maximize the figure
set(fig, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
% Save the figure as a PNG file
saveas(fig, ['Pattern_d_lambda_', num2str(k), '.png']);
% Close the figure to avoid excessive open windows
close(fig);
end
```

Listing 2: Broadside array simulation with varying d/lambda

4 End-Fire Array Analysis

4.1 Varying Number of Elements (N) with Fixed $d/\lambda = 0.25$

4.1.1 Theory

For end-fire arrays with $\beta = -kd$, increasing N produces:

- Highly directional beam along the array axis
- Rapid beamwidth reduction
- Higher front-to-back ratio
- Increased directivity

4.1.2 Radiation Patterns

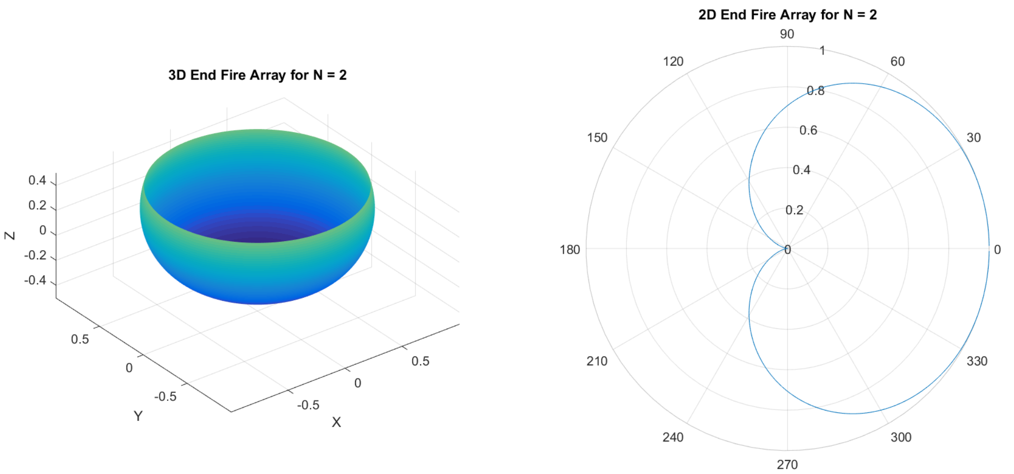


Figure 23: End-fire array radiation patterns for varying N=2

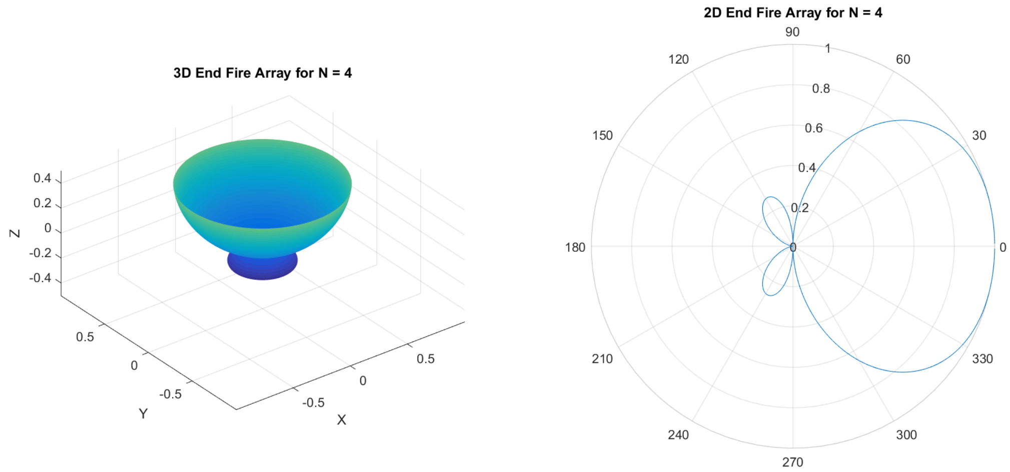


Figure 24: End-fire array radiation patterns for varying N=4

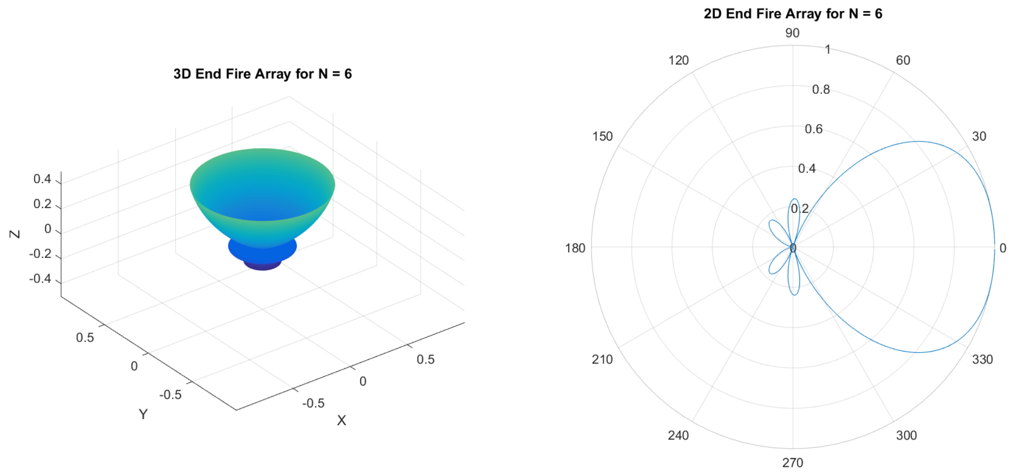


Figure 25: End-fire array radiation patterns for varying N=6

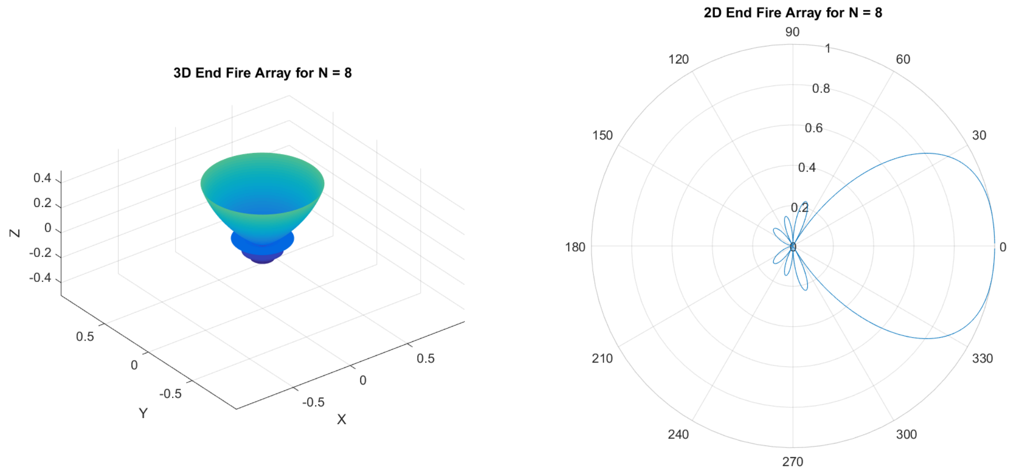


Figure 26: End-fire array radiation patterns for varying N=8

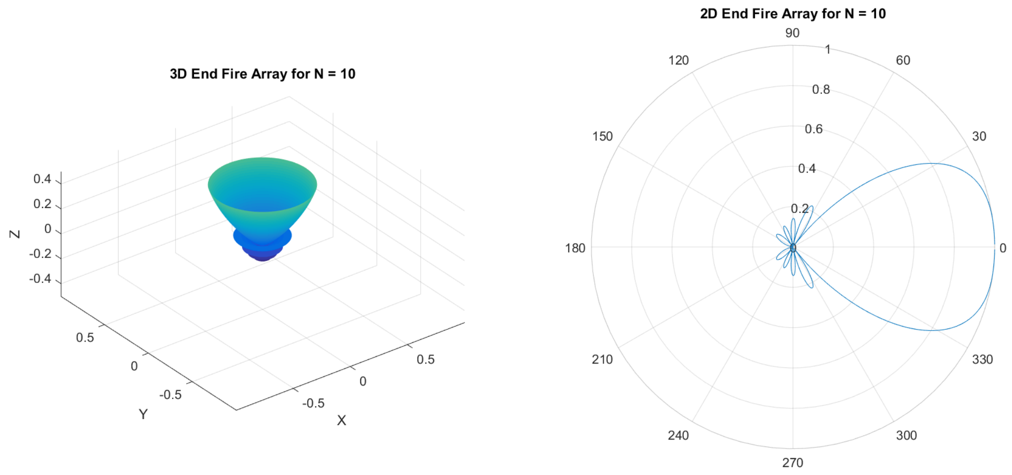


Figure 27: End-fire array radiation patterns for varying N=10

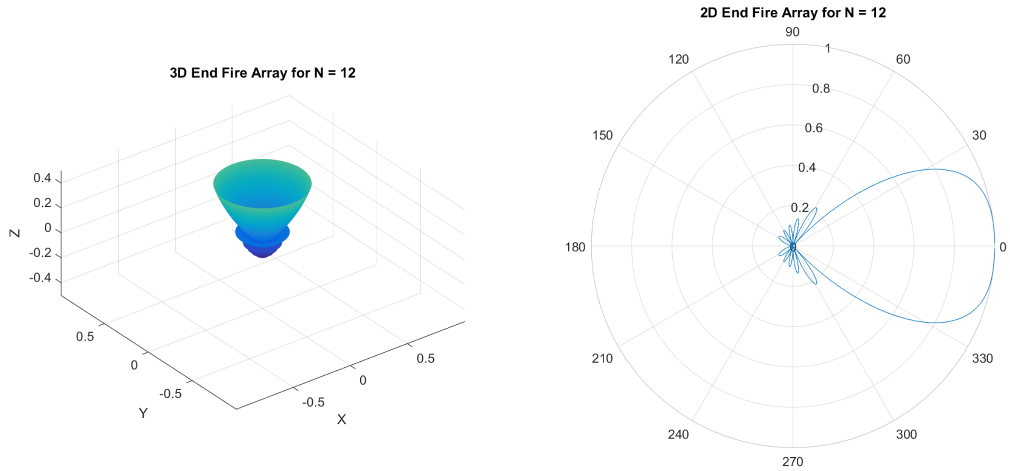


Figure 28: End-fire array radiation patterns for varying N=12

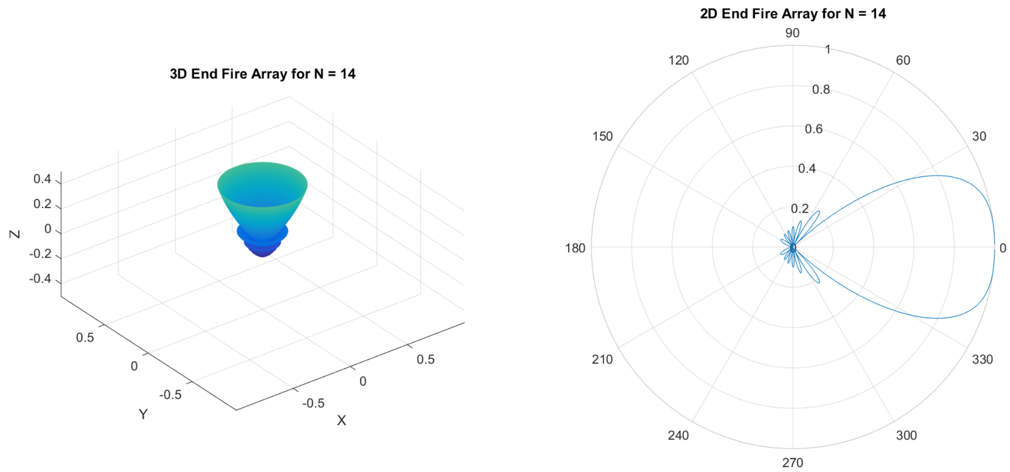


Figure 29: End-fire array radiation patterns for varying N=14

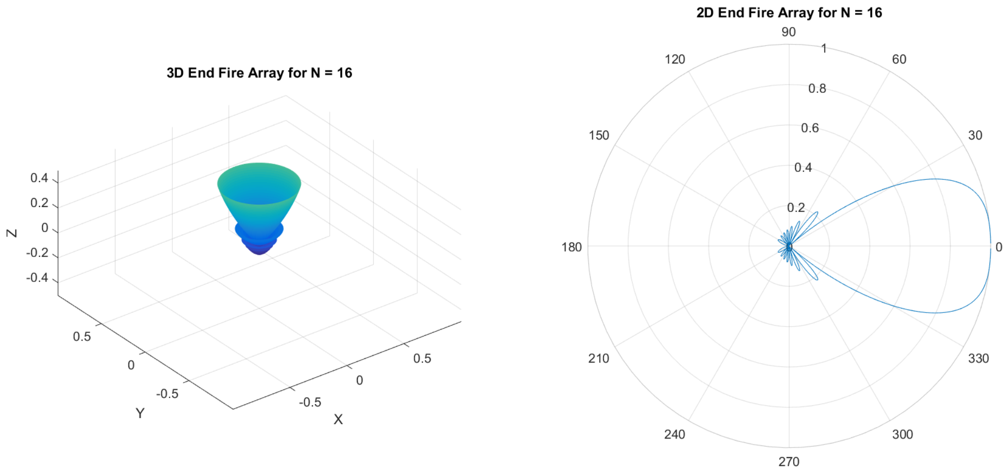


Figure 30: End-fire array radiation patterns for varying N=16

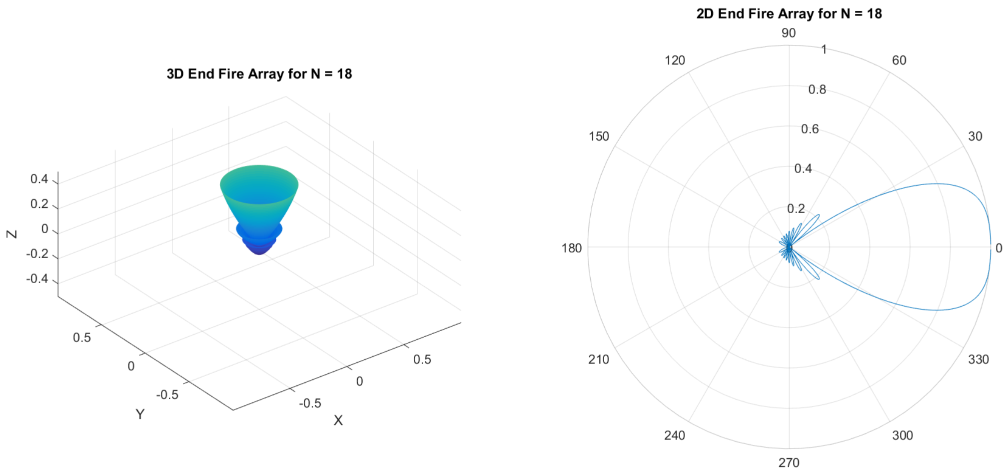


Figure 31: End-fire array radiation patterns for varying N=18

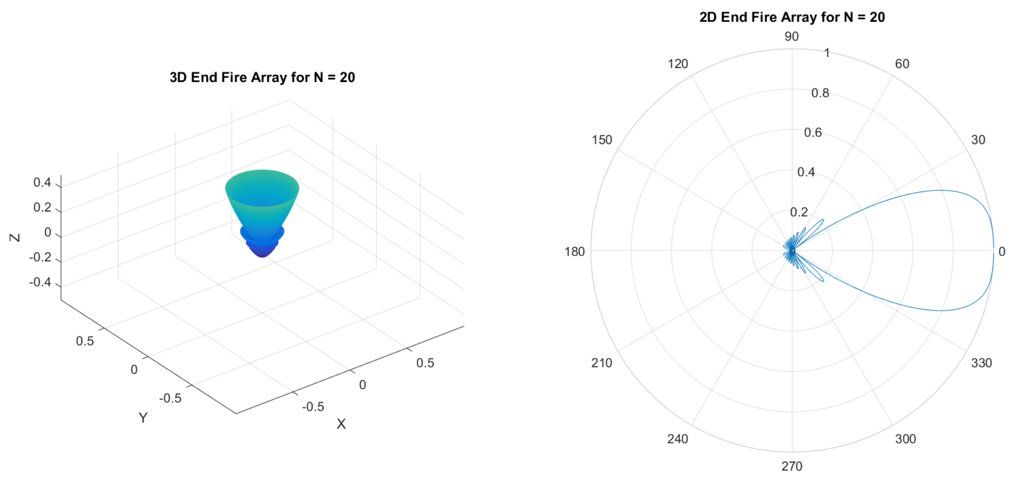


Figure 32: End-fire array radiation patterns for varying N=20

Analysis:

- $N = 1$: Omnidirectional pattern
- $N = 2 - 5$: Cardioid-like pattern develops
- $N = 8 - 12$: Sharp directional beam toward $\theta = 0$
- $N = 15 - 20$: Extremely narrow beam, very high directivity

4.1.3 MATLAB Code

```
% Take d = \lambda/2 > \beta * d = pi
% We vary N
theta = linspace(0, 2*pi, 500); % theta from 0 to 2*pi
phi = linspace(0, 2*pi, 500); % phi from 0 to 2*pi
[Theta, Phi] = meshgrid(theta, phi);
beta_d = pi./2;
N = 2:2:20;
for i = 1:length(N)
% Array factor calculation for broadside array
numerator_3D = sin(N(i) * 0.5 * beta_d * (cos(Theta)-1));
denominator_3D = N(i) * sin(0.5 * beta_d * (cos(Theta)-1));
numerator = sin(N(i) * 0.5 * beta_d * (cos(theta)-1));
denominator = N(i) * sin(0.5 * beta_d * (cos(theta)-1));
F_AF_3D = numerator_3D ./ denominator_3D;
F_AF = numerator ./ denominator;
% Convert to Cartesian coordinates for 3D plotting
X = abs(F_AF_3D) .* sin(Theta) .* cos(Phi);
Y = abs(F_AF_3D) .* sin(Theta) .* sin(Phi);
Z = abs(F_AF_3D) .* cos(Theta);
% Create a new figure for both 3D and 2D plots
```

```
fig = figure;
% Plot the radiation pattern in 3D
subplot(1, 2, 1);
surf(X, Y, Z);
shading interp;
title(['3D␣End␣Fire␣Array␣for␣N␣=␣', num2str(N(i))]);
xlabel('X'); ylabel('Y'); zlabel('Z');
axis equal;
axis([-1 1 -1 1 -0.5 0.5] * max(abs(F_AF_3D(:)))); % Set axis
    limits tomake differences noticeable
% Plot the radiation pattern in 2D
subplot(1, 2, 2);
polarplot(theta, abs(F_AF)); % 2D pattern for a fixed phi slice
title(['2D␣End␣Fire␣Array␣for␣N␣=␣', num2str(N(i))]);
% Maximize the figure
set(fig, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
% Save the figure as a PNG file
saveas(fig, ['End_Fire_Array_N_', num2str(N(i)), '.png']);
% Close the figure to avoid excessive open windows
close(fig);

end
```

Listing 3: End-fire array simulation with varying N

4.2 Varying Element Spacing (d/λ) with Fixed $N = 8$

4.2.1 Theory

Spacing variation in end-fire arrays affects:

- Beamwidth and directivity
- Back lobe suppression
- Occurrence of grating lobes
- Optimal spacing typically around $d/\lambda = 0.25$ to 0.5

4.2.2 Radiation Patterns

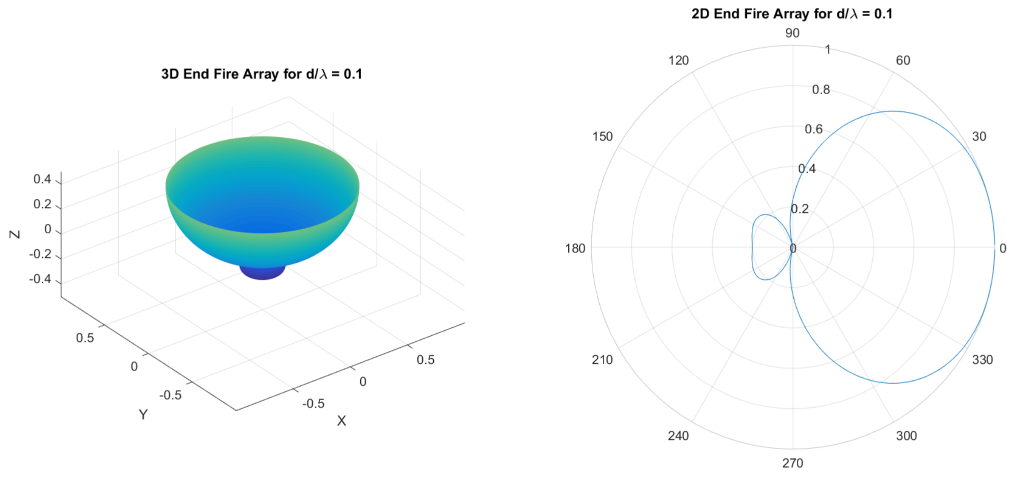


Figure 33: End-fire array radiation patterns for varying d/λ =0.1

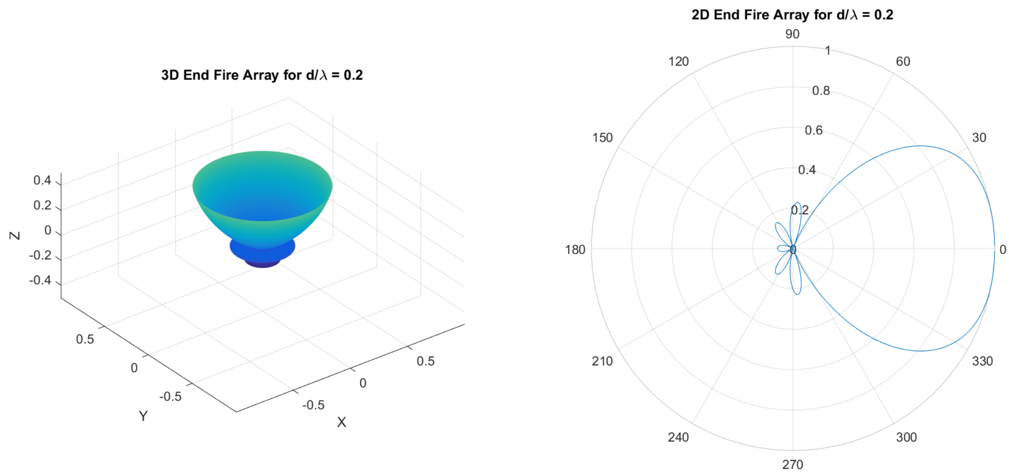


Figure 34: End-fire array radiation patterns for varying d/λ =0.2

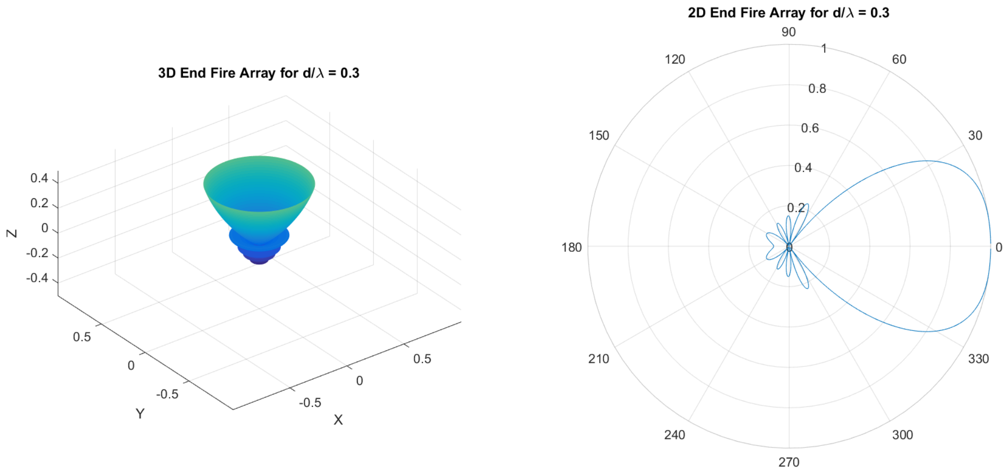


Figure 35: End-fire array radiation patterns for varying d/λ =0.3

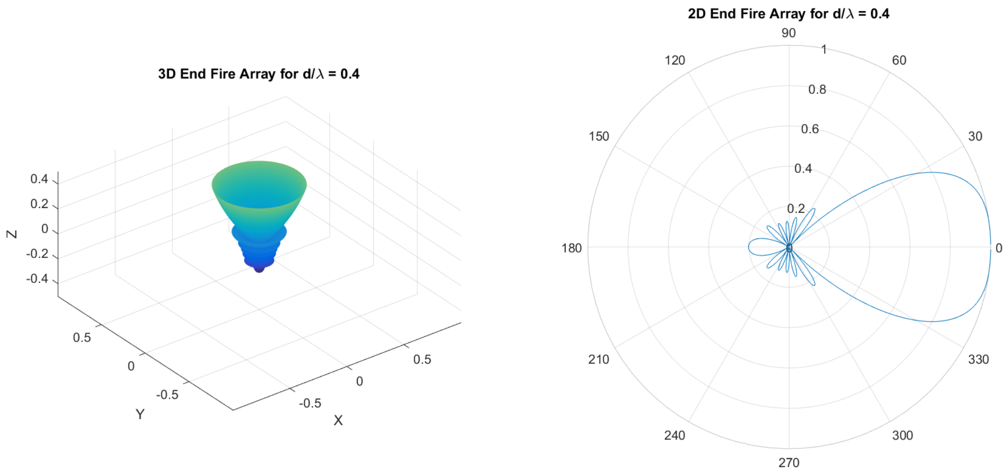


Figure 36: End-fire array radiation patterns for varying d/λ =0.4

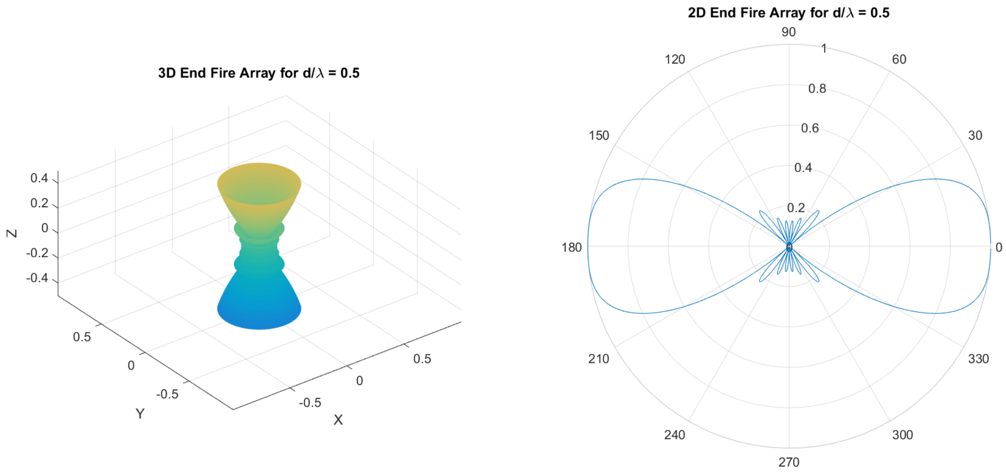


Figure 37: End-fire array radiation patterns for varying d/λ =0.5

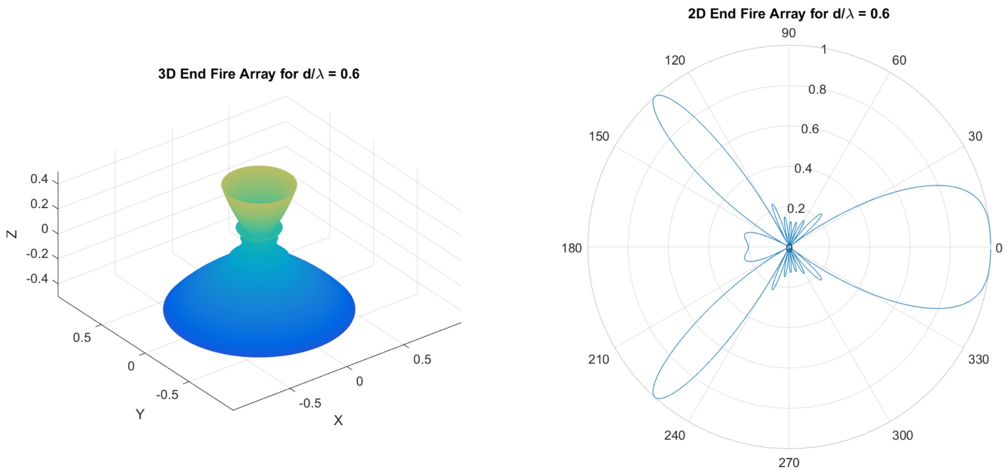


Figure 38: End-fire array radiation patterns for varying d/λ =0.6

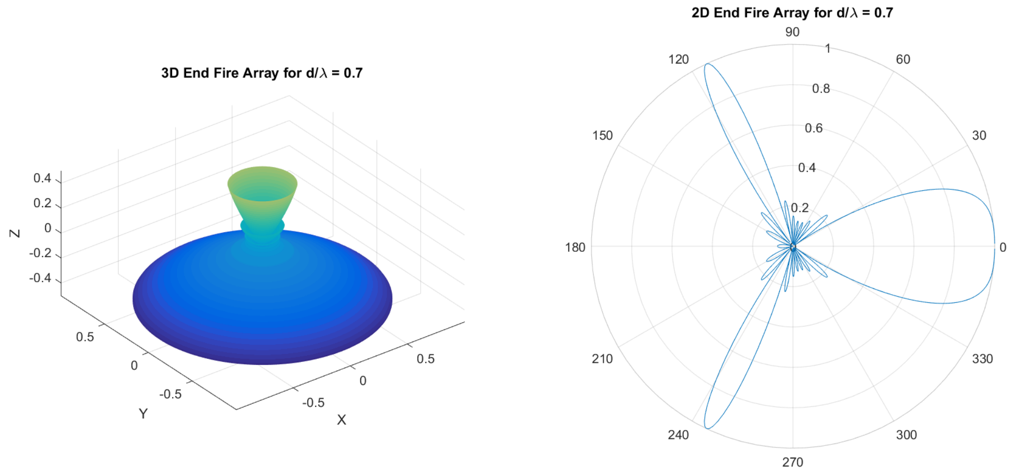


Figure 39: End-fire array radiation patterns for varying d/λ =0.7

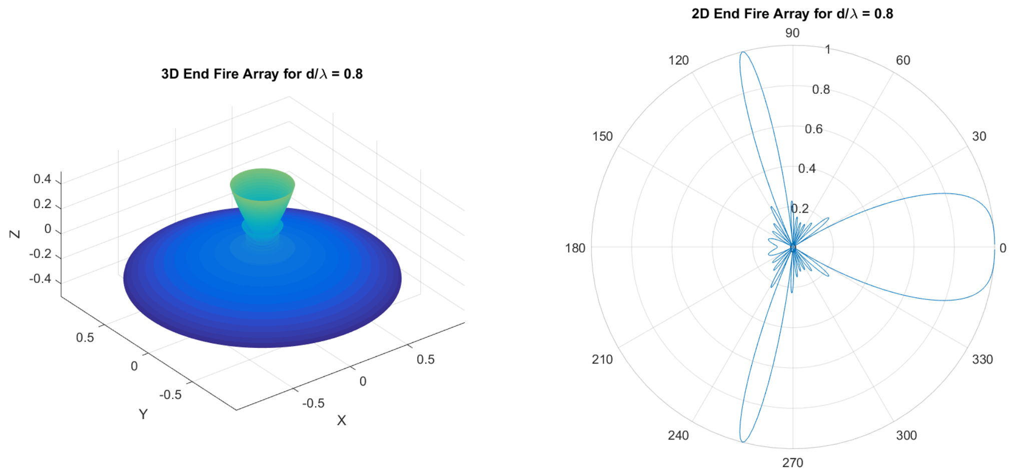


Figure 40: End-fire array radiation patterns for varying d/λ =0.8

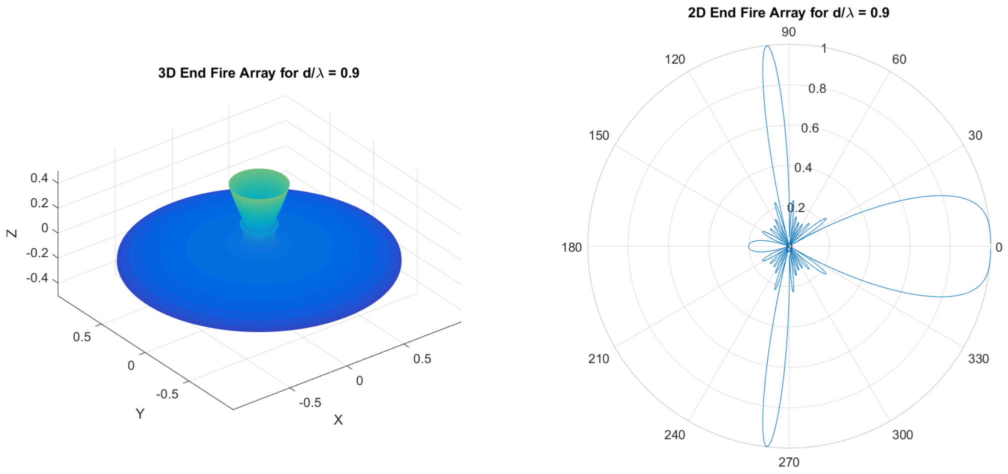


Figure 41: End-fire array radiation patterns for varying d/λ =0.9

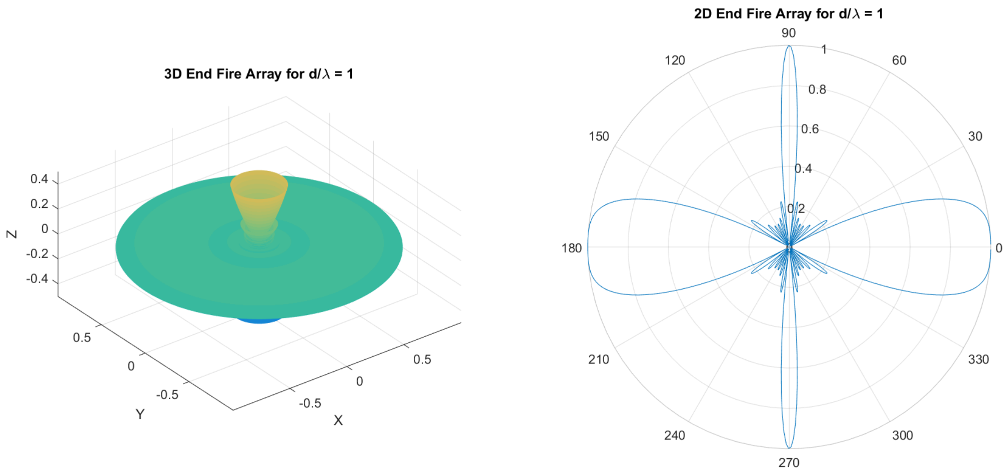
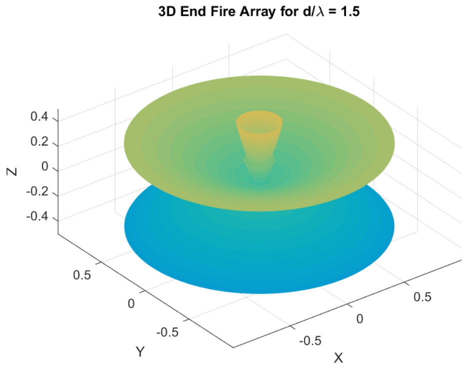


Figure 42: End-fire array radiation patterns for varying d/λ =1



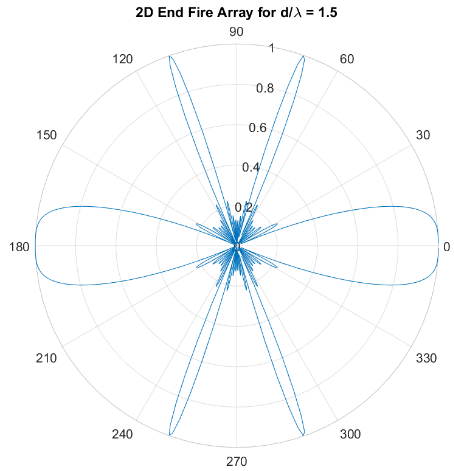
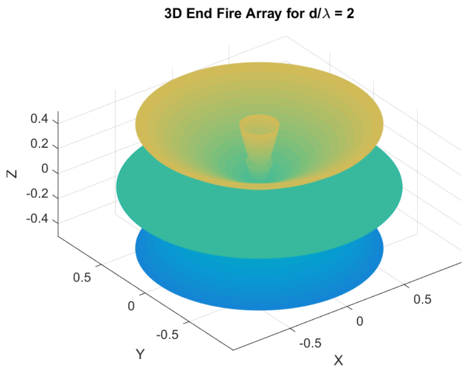


Figure 43: End-fire array radiation patterns for varying d/λ =1.5



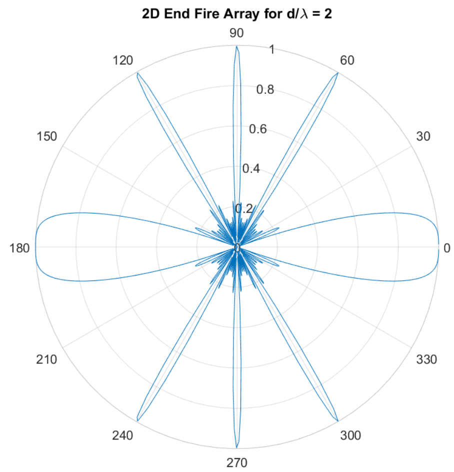


Figure 44: End-fire array radiation patterns for varying d/λ =2

Analysis:

- $d/\lambda = 0.1 - 0.3$: Broad beam, low directivity
- $d/\lambda = 0.4 - 0.6$: Well-defined directional pattern
- $d/\lambda = 0.7 - 0.9$: Narrower beam, potential back lobes
- $d/\lambda = 1.0 - 2.0$: Multiple lobes appear, pattern degradation

4.2.3 MATLAB Code

```
% Take N = 8, vary d / lambda
theta = linspace(0, 2*pi, 500); % theta from 0 to 2*pi
phi = linspace(0, 2*pi, 500); % phi from 0 to 2*pi
[Theta, Phi] = meshgrid(theta, phi);
N = 8;
d_lambda_values = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,
    1, 1.5, 2]; % Vary d/lambda from 0.1 to 1
for k = 1:length(d_lambda_values)
% Calculate beta_d from d/lambda
d_lambda = d_lambda_values(k);
beta_d = 2 * pi * d_lambda; % Beta * d = 2 * pi * (d / lambda)
% Array factor calculation for broadside array
numerator_3D = sin(N * 0.5 * beta_d * (cos(Theta)-1));
denominator_3D = N * sin(0.5 * beta_d * (cos(Theta)-1));
numerator = sin(N * 0.5 * beta_d * (cos(theta)-1));
denominator = N * sin(0.5 * beta_d * (cos(theta)-1));
F_AF_3D = numerator_3D ./ denominator_3D;
F_AF = numerator ./ denominator;
% Convert to Cartesian coordinates for 3D plotting
X = abs(F_AF_3D) .* sin(Theta) .* cos(Phi);
Y = abs(F_AF_3D) .* sin(Theta) .* sin(Phi);
Z = abs(F_AF_3D) .* cos(Theta);
% Create a new figure for both 3D and 2D plots
fig = figure;
% Plot the radiation pattern in 3D
subplot(1, 2, 1);
surf(X, Y, Z);
shading interp;
title(['3D End Fire Array for d/\lambda = ', num2str(d_lambda)]);
xlabel('X'); ylabel('Y'); zlabel('Z');
axis equal;
axis([-1 1 -1 1 -0.5 0.5] * max(abs(F_AF_3D(:)))); % Set axis
    limits to make differences noticeable
% Plot the radiation pattern in 2D
subplot(1, 2, 2);
polarplot(theta, abs(F_AF)); % 2D pattern for a fixed phi slice
title(['2D End Fire Array for d/\lambda = ', num2str(d_lambda)]);
% Maximize the figure
set(fig, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
% Save the figure as a PNG file
saveas(fig, ['End Fire Array_d_lambda_', num2str(k), '.png']);
% Close the figure to avoid excessive open windows
close(fig);
end
```

Listing 4: End-fire array simulation with varying d/lambda