Wave propagation

Draw short dipole with constant current voltage field radiation pattern of normalized $\mathbf{E}_\theta$ (θ plane) and normalized $\mathbf{H}_\phi$ (ϕ plane) using MATLAB

Assignment 3

Supervisors:

Dr.**Nur ad-Din M. Salem**

Eng.**Ehab**

Thesis written by:

Ahmed khaled Fathi

16/10/2024

0.1 Antenna Radiation Pattern Analysis

0.1.1 Code Implementation

The implementation consists of four main parts that visualize both E-plane and H-plane radiation patterns in polar and Cartesian coordinates.

Variable Initialization

```
theta = linspace(0, pi, 360);
phi = linspace(0, 2*pi, 360);
E_theta = sin(theta);
H_phi = ones(size(phi));
```

Listing 1: variable Initialization

Here, we initialize the angular vectors and field patterns:

- **theta**: Angular vector from 0 to π with 360 points
- **phi**: Angular vector from 0 to 2π with 360 points
- **E_theta**: Electric field pattern following sine function
- **H_phi**: Magnetic field pattern (uniform/omnidirectional)

E-plane Polar Plot

```
  figure (1);
  polarplot([-theta, theta], [E_theta, E_theta], 'r', 'LineWidth', 1.5);
  title("Plot E_theta (E-plane) in polar coordinates")
  ax = gca;
  ax.ThetaDir = 'clockwise';
  ax.ThetaZeroLocation = 'top';
```

Listing 2: Code to plot $\mathbf{E}_\theta$ in polar coordinates

The E-plane radiation pattern is plotted in polar coordinates showing:

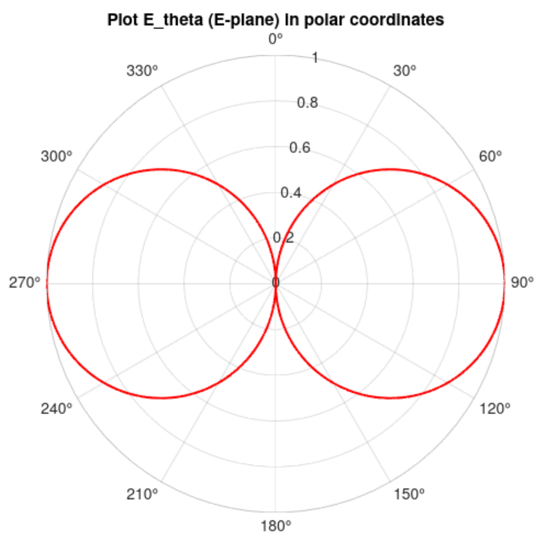

Figure 1: Plot for $\mathbf{E}_\theta$ in polar coordinate

- Symmetric pattern about the vertical axis
- Shaped pattern characteristic of short dipole antennas
- Clockwise orientation with zero at top position

H-plane Polar Plot

```
figure (2);
polarplot(phi, H_phi, 'r', 'LineWidth', 1.5);
title('H-plane Radiation Pattern (Polar Plot)');
ax = gca;
ax.ThetaDir = 'clockwise';
ax.ThetaZeroLocation = 'top';
rlim([0 1]);
```

Listing 3: Code to plot $\mathbf{H}_{\phi}$ in polar coordinates

The E-plane radiation pattern is plotted in polar coordinates showing:

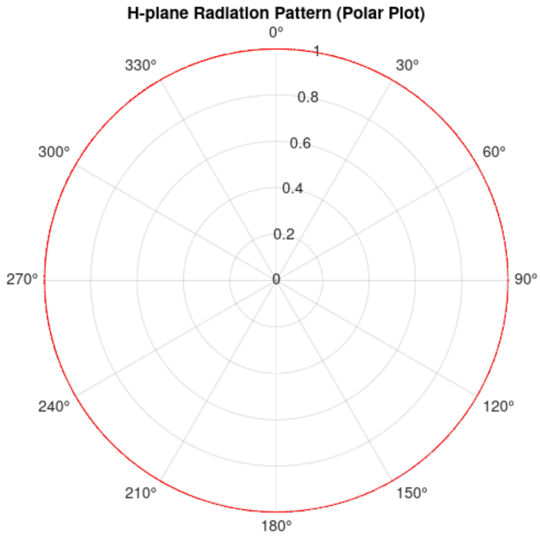

Figure 2: Plot for $\mathbf{H}_{\phi}$ in polar coordinate

The H-plane radiation pattern demonstrates:

- Omnidirectional radiation pattern
- Uniform field strength across all angles
- Limited to unit radius for normalized representation

E-plane Cartesian Plot

For the E-plane Cartesian representation:

```
figure (3);
x_E = E_theta .* cos(theta);
y_E = E_theta .* sin(theta);
x_EMirror = -E_theta .* cos(theta);
y_EMirror = -E_theta .* sin(theta);
plot(x_E, y_E,  'b' ,x_EMirror,y_EMirror, 'b' , 'LineWidth', 1.5);
axis equal;
%-------labels--------------
xlabel('\theta');
xticks([-1.5 -1 -0.5 0 0.5 1 1.5]);
xticklabels({'-540', '-360', '-180', '0', '180', '360', '540'});
ylabel('E_\theta');
title('E-plane Radiation Pattern (Cartesian Plot)');
%-------limits&corections--------------
xlim([-1.5, 1.5]);
ylim([-1.5, 1.5]);
view(90, -90);
grid on;
```

Listing 4: Code to plot $\mathbf{E}_{\theta}$ in Cartesian coordinates

The conversion to Cartesian coordinates involves:

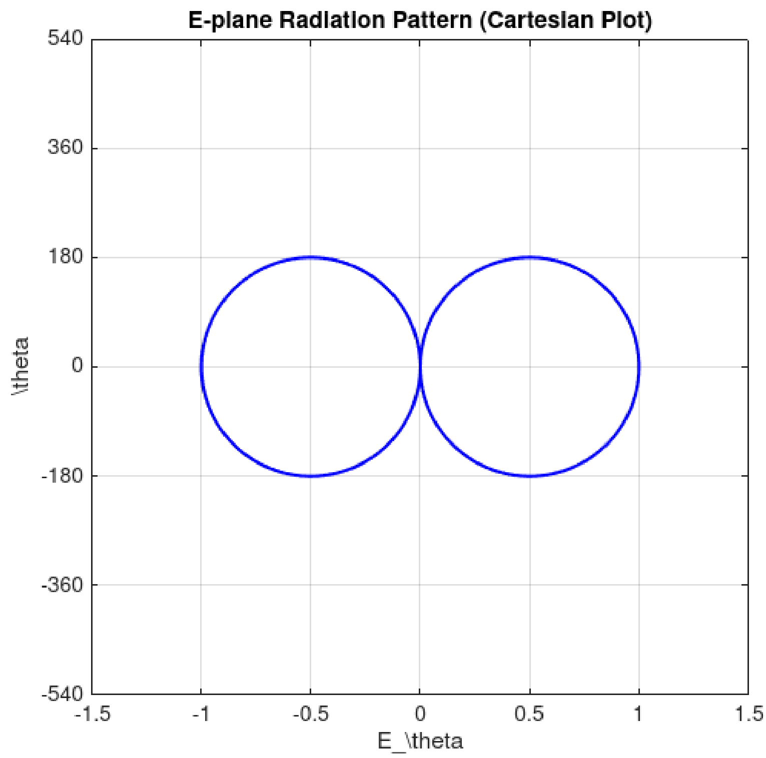

Figure 3: Plot for $\mathbf{E}_\theta$ in Cartesian coordinate

- Polar to Cartesian transformation using trigonometric functions
- Mirrored pattern creation for complete visualization
- Custom axis labeling in degrees

Plot Customization

Key visualization parameters include:

- Axis limits:$[-1.5, 1.5]$ for both x and y

```
xlim([-1.5, 1.5]);
ylim([-1.5, 1.5]);
```

- Grid overlay for better readability
- Degree markings from -540° to 540°

```
xticks([-1.5 -1 -0.5 0 0.5 1 1.5]);
xticklabels({'-540', '-360', '-180', '0', '180', '360', '540'});
```

- Equal axis scaling for proper pattern representation

```
axis equal;
```

H-plane Cartesian Plot

For the H-plane Cartesian representation:

```
figure (4);
x_H = H_phi .* cos(phi);
y_H = H_phi .* sin(phi);
plot(x_H , y_H, 'r', 'LineWidth', 1.5);
axis equal;
%-------labels--------------
xlabel('\phi (degrees)');
ylabel('H_\phi');
title('H-plane Radiation Pattern (Cartesian Plot)');
%-------limits&coorections--------------
xticks([-1.5 -1 -0.5 0 0.5 1 1.5]);
xticklabels({'-540', '-360', '-180', '0', '180', '360', '540'});
xlim([-1.5, 1.5]);
ylim([-1.5, 1.5]);
grid on;
```

Listing 5: Code to plot $\mathbf{H}_\phi$ in Cartesian coordinates

The conversion to Cartesian coordinates involves:

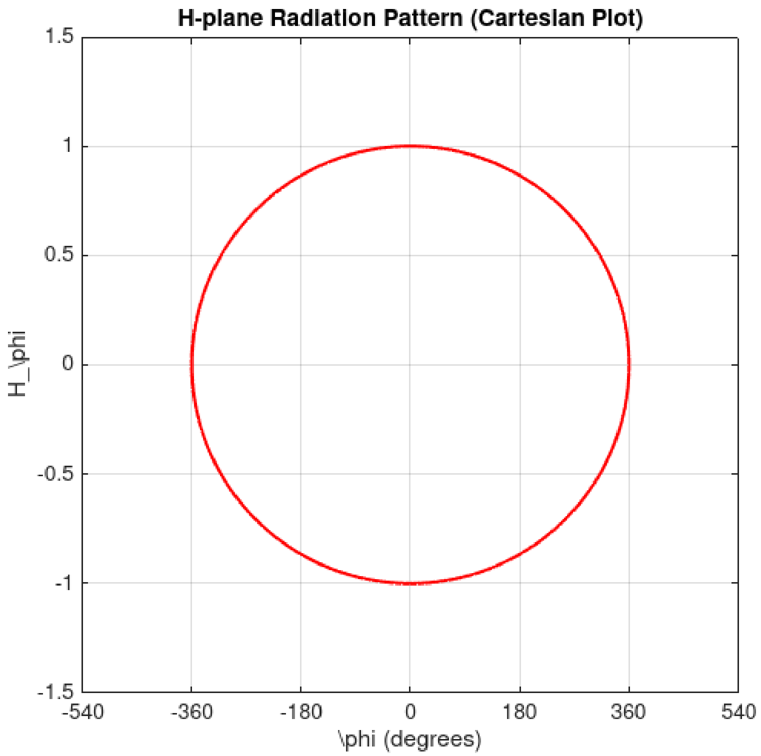

Figure 4: Plot for $\mathbf{H}_\phi$ in Cartesian coordinate

- Polar to Cartesian transformation using trigonometric functions
- Mirrored pattern creation for complete visualization
- Custom axis labeling in degrees

Plot Customization

Key visualization parameters include:

- All $\mathbf{E}_\theta$ plot customizations
- Degree markings from -540° to 540° on the y Axis

0.2 Extra

we can plot $\mathbf{E}_\theta$ and $\mathbf{H}_\phi$ on a 3d plane using

```
[theta, phi] = meshgrid(linspace(0, pi, 100), linspace(0, 2*pi, 100));
E_theta = sin(theta);
x = E_theta .* sin(theta) .* cos(phi);
y = E_theta .* sin(theta) .* sin(phi);
z = E_theta .* cos(theta);
figure('Position', [100, 100, 800, 600]);
plot(1);
mask = y >= 0;
x_masked = x;
y_masked = y;
z_masked = z;
x_masked(~mask) = NaN;
y_masked(~mask) = NaN;
z_masked(~mask) = NaN;
surf(x_masked, y_masked, z_masked);
hold on;
patch([0 1.5 1.5 0], [0 0 0 0], [-1.5 -1.5 1.5 1.5], 'FaceColor', [0.8 0.8 0.8], '
  FaceAlpha', 0.3, 'EdgeColor', 'k');
theta_line = linspace(0, pi, 100);
x_line = sin(theta_line) .* sin(theta_line);
z_line = sin(theta_line) .* cos(theta_line);
plot3(x_line, zeros(size(x_line)), z_line, 'r', 'LineWidth', 2);
shading interp
colormap('winter')
lighting gouraud
material([0.6 0.8 0.1])
camlight('headlight')
xlabel('x')
ylabel('y')
zlabel('z')
axis equal
axis([-1.5 1.5 0 1.5 -1.5 1.5])
grid on
view(45, 30)
title('3D Radiation Pattern with Cutaway')
```

Listing 6: Code to plot $\mathbf{E}_\theta$ and $\mathbf{H}_\phi$ in 3d Cartesian coordinates

That will give us :

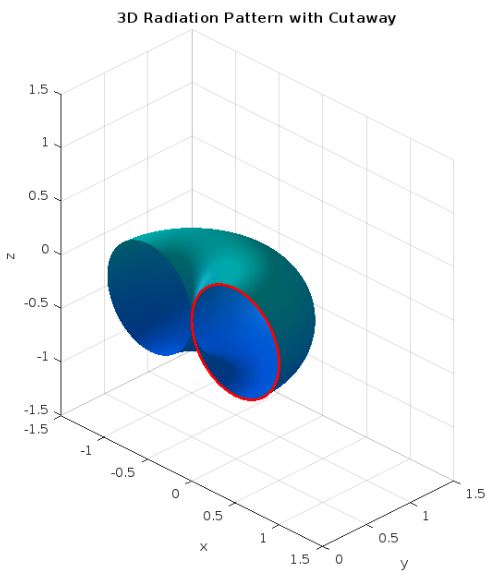

Figure 5: Plot for $\mathbf{E}_\theta$ and $\mathbf{H}_\phi$ in Cartesian coordinate