ECE 541 Project 5

Elim Aschenberg

Colorado State University

October 2025

Introduction

This project explores the analysis of a thin wire scatterer. Initially, this report will discuss the theory and the equations used in this analysis, as well as the algorithm employed in the Method of Moments analysis. Then, the results from the applied algorithm will be reviewed. This will be done in two cases: 1) where the angle of incidence, θ, is varied, and 2) where the length of the wire and its radius are varied. Finally, these results will be validated against WIPL-D software.

1 Theory & Equations

For this report, the wire will be oriented along the z-axis. The analysis of a thin wire scatterer starts by dividing the wire into segments. An arbitrary value N is selected, which represents the number of segments. The distance between each segment is constant and can be calculated by

$$d = \frac{L}{N+1}$$

where L is the length of the wire. The position at the center of the wire is denoted by z_n and the perimeter by z_m, where $m = n = N$.

The first major step in this analysis is calculating the impedance matrix Z_{mn}

$$Z_{mn} = j\omega\mu_0 \frac{1}{4\pi}\left[\Psi_1(m,n) + \frac{1}{\beta^2 d}\Psi_2(m,n)\right].$$

$$\Psi_1(m,n) = \int_{Z_{n-1}}^{Z_{n+1}} [T(z-z_m)\frac{cos(\beta R_n)}{R_n}dz - \frac{cos(\beta a)}{R_m}]dz + cos(\beta a)\int_{Z_{n-1}}^{Z_{n+1}} \frac{1}{R_m}dz$$

$R_m \approx a$ when $m = n$, which creates a Quasi-singular function. This function is similar to a delta function, which is why it is called Quasi-singular, meaning near singular. Consequently, the above equation can be rewritten in two parts to accommodate this.

Case: $m \neq n$

$$\Psi_1(m,n) = \int_{-d}^{d}\left(1 - \frac{|x|}{d}\right)\frac{e^{-j\beta R}}{R}\,dx, \quad R = \sqrt{\left((n-m)d - x\right)^2 + a^2}$$

Case: $m = n$

$$\begin{aligned}\Psi_1(n,n) = 2\int_0^d &\left[\left(1-\frac{x}{d}\right)\frac{\cos(\beta R)}{R} - \frac{\cos(\beta a)}{R}\right]dx \\ &-2j\int_0^d\left(1-\frac{x}{d}\right)\frac{\sin(\beta R)}{R}\,dx \\ &+2\cos(\beta a)\,\ln\left(\frac{d+\sqrt{d^2+a^2}}{a}\right)\end{aligned}$$

$$R = \sqrt{x^2 + a^2}$$

For Ψ_2

$$R_0 = \sqrt{\left((n-m)d\right)^2 + a^2} \qquad R_1 = \sqrt{\left((n-m-1)d\right)^2 + a^2} \qquad R_2 = \sqrt{\left((n-m+1)d\right)^2 + a^2}$$

$$\Psi_2(m,n) = \frac{e^{-j\beta R_1}}{R_1} - 2\frac{e^{-j\beta R_0}}{R_0} + \frac{e^{-j\beta R_2}}{R_2}.$$

Next, the voltage matrix can be easily solved

$$v_m = E_z^{inc}(z') = -E_0 e^{j\beta z' cos\theta_{inc}} sin\theta_{inc}$$

Using the voltage matrix and the impedance matrix, the current can now be solved. Using the equation

$$[Z][I] = [V] \text{ which can be rearranged to be } [I] = [Z]^{-1}[V]$$

Using this new current matrix, the radar cross-section can now be calculated.

$$S_{rad}(\theta) = \frac{(\omega\mu_0 sin\theta|Q|)^2}{4\pi E_0^2}$$

where Q can be calculated by

$$Q = \sum_{n=1}^{N+1}[I_{n-1}\Psi_3(n) + \frac{I_n - I_{n-1}}{d}\Psi_4(n)]$$

$$\Psi_3(n) = \int_0^d e^{j\beta[(n-1)d+x]cos\theta}dx$$

$$\Psi_4(n) = \int_0^d xe^{j\beta[(n-1)d+x]cos\theta}dx$$

2 Using the characteristics in table 1, compute and plot the magnitude and phase of the current distribution

Antenna length, L	2 m
Antenna radius, a	12.5 mm
Frequency, f	300 MHz
Angle of incidence, θ	90, 30, 75 degrees
Incident plane wave,$E_\theta(0)$	1 V/m (rms magnitude)

Table 1: Characteristic Values

2.1 MATLAB

MATLAB Plots

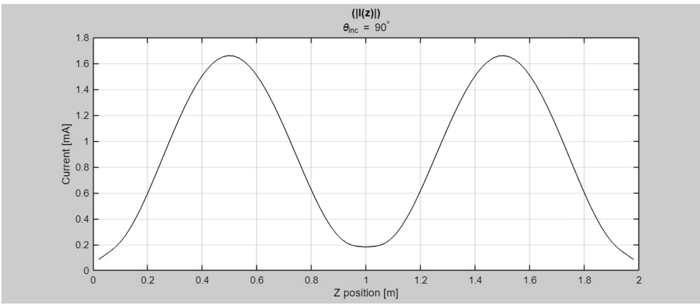
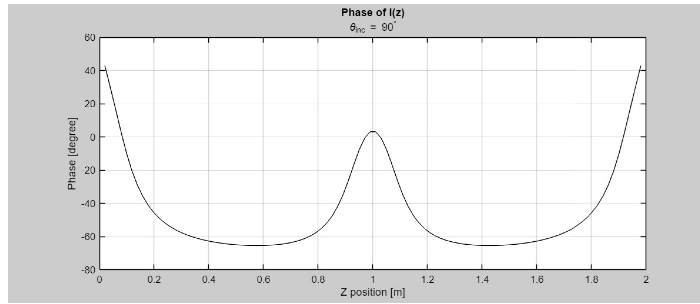
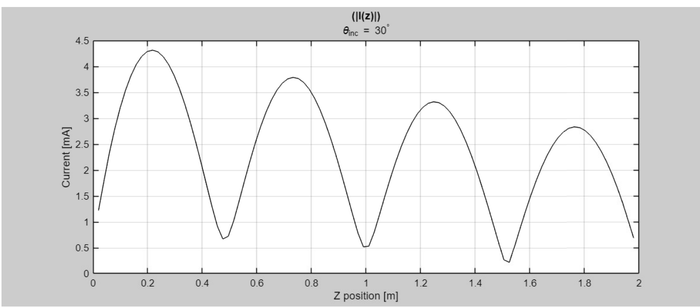
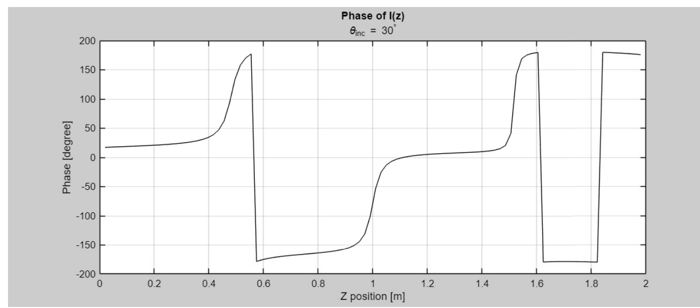
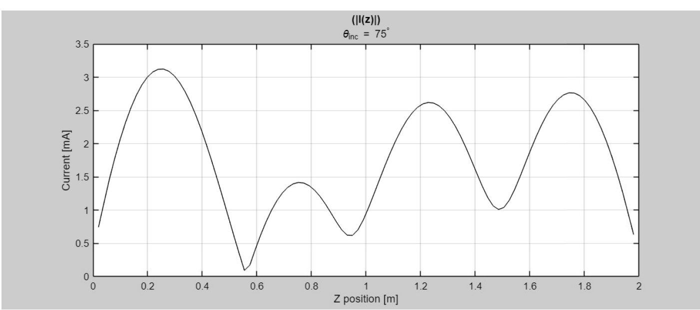
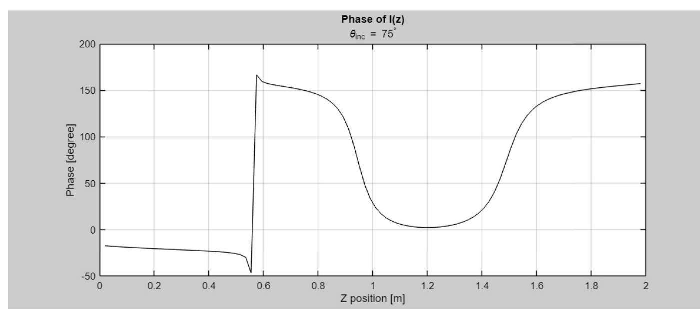
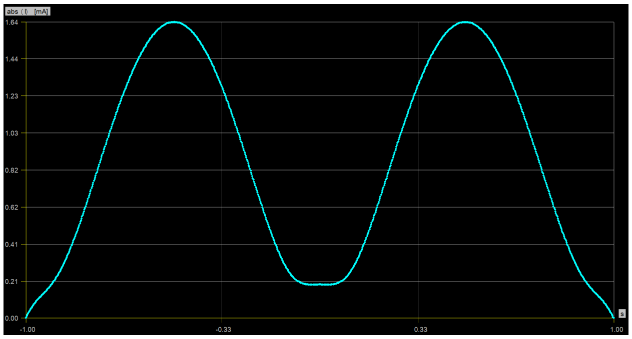
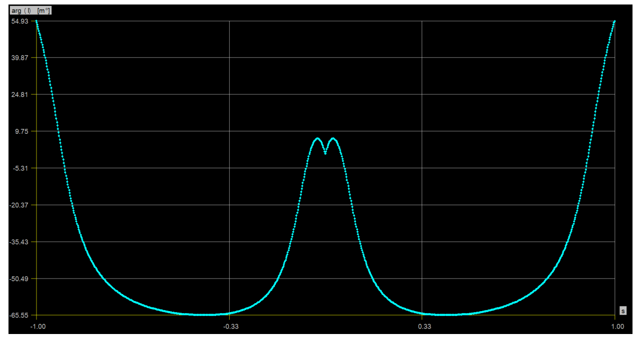
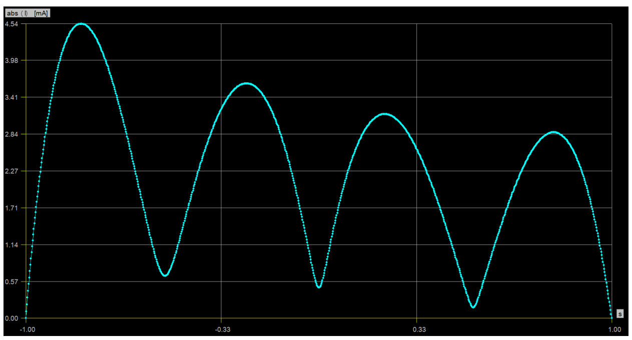
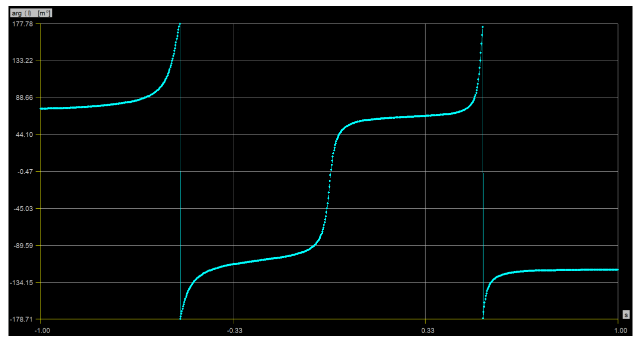
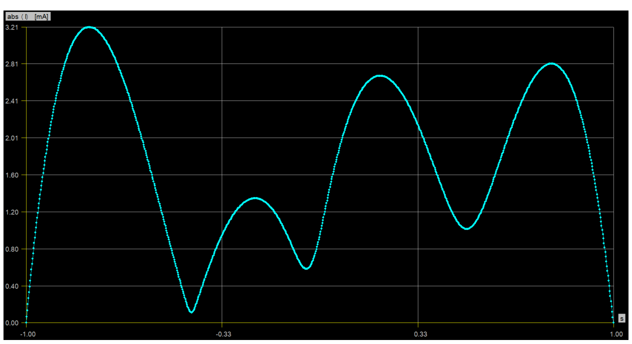
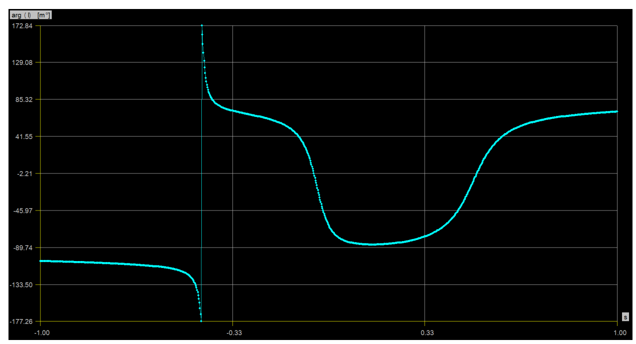
Current Magnitude	**Current Phase**
	
	
	

Table 2: Magnitude and Phase Plots (MATLAB)

2.2 WIPL-D

WIPL-D Plots

Table 3: Magnitude and Phase Plots (WIPL-D)

$\theta_{\textbf{incident}}$	**Current Magnitude**	**Current Phase**
90°		
30°		
75°		

2.3 Comment on results

The MATLAB results correlate nearly 1:1 with the WIPL-D simulation results. For $\theta_{\text{inc}} = 90°$, the peak value for the WIPL-D simulation is 1.64 mA and 1.671 mA for the MATLAB simulation. For $\theta_{\text{inc}} = 30°$, the peak value for the WIPL-D simulation is 4.54 mA and 4.054 mA for the MATLAB simulation. Finally, for $\theta_{\text{inc}} = 75°$, the peak value for the WIPL-D simulation is 3.21 mA and 2.985 mA for the MATLAB simulation.

While these values are not identical, they are remarkably close to one another. Furthermore, the general pattern that the graphs take is nearly identical.

The phase graphs are a little more difficult to compare; however, it is obvious that they are consistent with one another. For $\theta_{\text{inc}} = 90°$, the peak value for the WIPL-D simulation is 54.93° with the middle peak hovering around 9.75°. This is similar to MATLAB, in which the center peak is just above 8°. The peak value is not the same, but that can be assumed to be because the plot does not go explicitly to the 0 m and 2 m marks.

For $\theta_{\text{inc}} = 30°$, the peak value for the WIPL-D simulation is around 175° while in MATLAB it is just under 180°. The valleys are at the negative peak values for both, which makes sense because the phase wraps at ±180°. A more accurate way to compare the two would be to look at the position on the wire at which these peaks/valleys occur. However, due to the way WIPL-D was simulated, the values do not line up on the graph, so I will elect to avoid this comparison.

Finally, and concisely, the phases for when $\theta_{\text{inc}} = 75°$ are visually the same. Assuming the same trends as the previous two graphs, this also validates my results from MATLAB.
To touch on the topic of convergence, the simulation is run with varying values for N. There was no explicit equation given, and the data from WIPL-D isn't readily available, so I will be using the difference between $|I|_{max}$ to discuss how increasing N either increases or decreases the accuracy of the simulation.
For N = 50, the $|I|_{max} = 1.631$ mA, 4.705 mA, and 3.246 mA, for $\theta_{inc} = 90°, 30°, \text{and } 75°$ respectively. For N = 500, the $|I|_{max} = 1.677$ mA, 3.916 mA, and 2.898 mA, for $\theta_{inc} = 90°, 30°, \text{and } 75°$ respectively. I will also use my previous values, which were taken at N = 200 which are $|I|_{max} = 1.671$ mA, 4.054 mA, and 2.985 mA, for $\theta_{inc} = 90°, 30°, \text{and } 75°$ respectively.

Table 4: Peak current magnitudes $|I|_{\max}$ from MATLAB (varying N) and WIPL-D for different θ_{inc}.

Method	N	$\|I\|_{\max}$ at $\theta_{\text{inc}} = 90°$ [mA]	$\|I\|_{\max}$ at $\theta_{\text{inc}} = 30°$ [mA]	$\|I\|_{\max}$ at $\theta_{\text{inc}} = 75°$ [mA]
MATLAB	50	1.631	4.705	3.246
MATLAB	60	1.639	4.840	3.259
MATLAB	75	1.649	4.605	3.247
MATLAB	100	1.661	4.316	3.124
MATLAB	150	1.669	4.120	3.023
MATLAB	200	1.671	4.054	2.985
MATLAB	500	1.677	3.916	2.898
WIPL-D	–	1.64	4.54	3.21

The convergence of the MoM solution with respect to N can be seen by tracking how $|I|_{\max}$ behaves relative to the WIPL-D reference values. For $\theta_{\text{inc}} = 90°$, the WIPL-D peak is 1.64 mA, and the MATLAB result moves from 1.631 mA at $N = 50$ to 1.639 mA at $N = 60$ (the closest match), then slowly increases up to 1.677 mA at $N = 500$, slightly overshooting the reference as N grows. For $\theta_{\text{inc}} = 30°$, MATLAB initially overestimates the peak (up to 4.840 mA at $N = 60$ versus 4.54 mA), and then $|I|_{\max}$ monotonically decreases with N, crossing past the WIPL-D value and approaching 3.916 mA at $N = 500$. For $\theta_{\text{inc}} = 75°$, the MATLAB values start slightly above 3.21 mA (about 3.246–3.259 mA for $N = 50$–60) and then decrease below the WIPL-D result as N increases, reaching 2.898 mA at $N = 500$. Overall, the data show that the numerical solution stabilizes as N increases, with the closest agreement to WIPL-D often occurring at moderate values of N rather than at the largest discretization.

3 Monostatic radar cross-section of the scatterer for L ranging from 0.25 m to 2 m

3.1 MATLAB

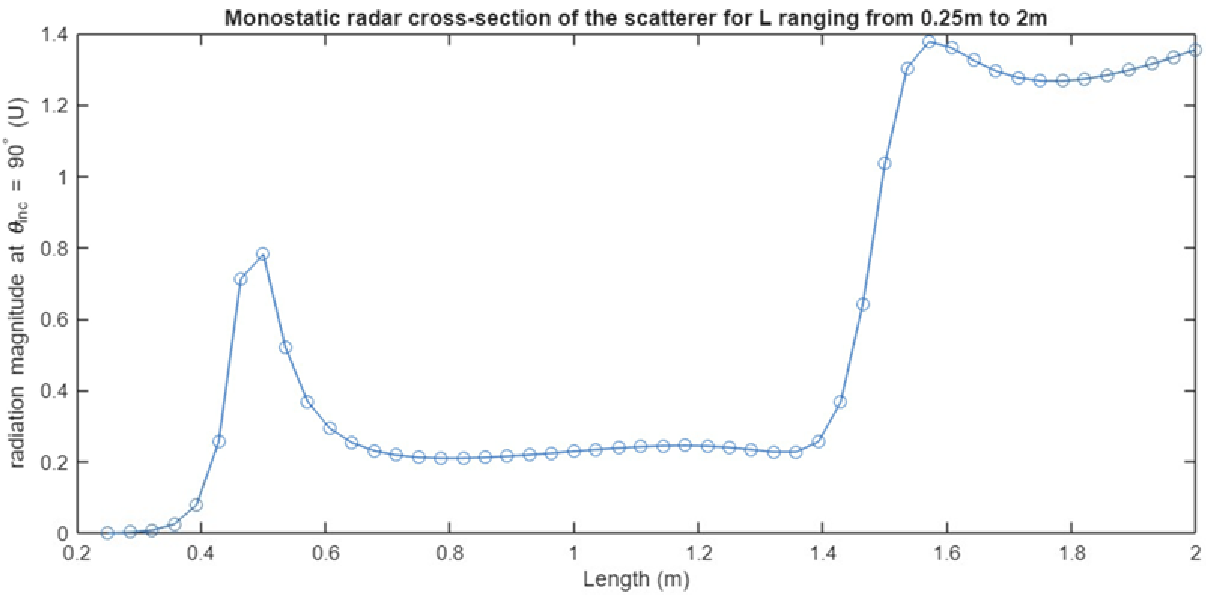


Figure 1: MATLAB Monostatic radar cross-section with 0.25 m ≤ L ≤ 2 m

3.2 WIPL-D

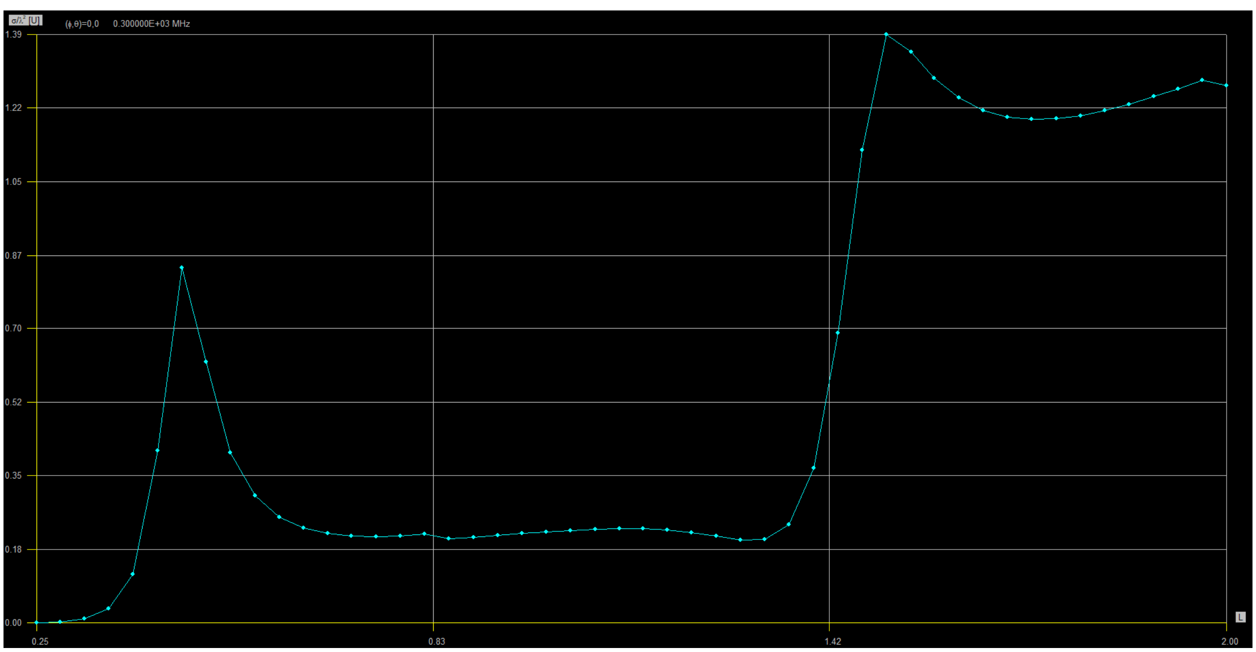


Figure 2: WIPL-D Monostatic radar cross-section with 0.25 m ≤ L ≤ 2 m

Table 5: MATLAB Radiation magnitude S_{rad} versus length L.

L [m]	S_{rad} [U]	L [m]	S_{rad} [U]
0.2500	0.0012	1.1429	0.2459
0.2857	0.0034	1.1786	0.2467
0.3214	0.0094	1.2143	0.2453
0.3571	0.0262	1.2500	0.2414
0.3929	0.0785	1.2857	0.2352
0.4286	0.2571	1.3214	0.2286
0.4643	0.7140	1.3571	0.2289
0.5000	0.7834	1.3929	0.2574
0.5357	0.5209	1.4286	0.3680
0.5714	0.3702	1.4643	0.6416
0.6071	0.2948	1.5000	1.0359
0.6429	0.2547	1.5357	1.3053
0.6786	0.2323	1.5714	1.3782
0.7143	0.2199	1.6071	1.3609
0.7500	0.2135	1.6429	1.3258
0.7857	0.2110	1.6786	1.2966
0.8214	0.2112	1.7143	1.2780
0.8571	0.2132	1.7500	1.2693
0.8929	0.2165	1.7857	1.2686
0.9286	0.2208	1.8214	1.2743
0.9643	0.2255	1.8571	1.2848
1.0000	0.2305	1.8929	1.2991
1.0357	0.2354	1.9286	1.3161
1.0714	0.2398	1.9643	1.3350
1.1071	0.2435	2.0000	1.3552

3.3 Comment on results

Both plots are plotted with 50 different values of the length, ranging from 0.25 m to 2 m. The plots undeniably resemble each other. For the WIPL-D plot, the peak value is at 1.39 U, and for the MATLAB plot, it is at 1.38 U. There are a few discrepancies between the two plots. Most notably, the mono-static radar cross-section value for the MATLAB plot appears to increase between 1.8 and 2 m. In the WIPL-D plot, however, it increases slightly but has a negative slope between the last two length values. Additionally, there is a distinct point at the first peak, but in the MATLAB plot, it isn't as distinct. This is likely due to slightly different changes in the length as it is being swept across.

4 Conclusion

In this project, the Method of Moments was successfully applied to a thin-wire scatterer to compute the current distribution and monostatic radar cross-section, and the numerical results were validated against WIPL-D simulations. The impedance and voltage matrices were formed using the thin-wire integral equations, and the resulting current distributions for multiple angles of incidence showed close agreement with WIPL-D in both magnitude and phase, with only small discrepancies near the peaks and phase-wrapping regions. A convergence study with respect to the number of segments N demonstrated that the MoM solution stabilizes as N increases, with moderate values of N often giving the closest match to the WIPL-D peak currents. Finally, the monostatic radar cross-section as a function of wire length L exhibited the same qualitative behavior in MATLAB and WIPL-D, including the primary peaks and overall trend, confirming that the implemented MoM formulation provides an accurate and reliable model for the scattering behavior of the thin wire.

MATLAB Code

Listing 1: Main MATLAB script for current distribution and radiation

```
clc
clear

% Given Parameters
global L a E0 N M beta w d epsilon0 mu0 j

L = 2;
a = 12.5e-3;
f = 300e6;
N = 500;
M = N;

deg90 = pi/2;
deg30 = pi/6;
deg75 = 5*pi/12;

theta_incident = [deg90 deg30 deg75];
theta_plot = linspace(0, 2*pi, N);
E0 = 1;

% Constants
mu0 = 4 * pi * 10^-7;
epsilon0 = 8.854 * 10^-12;
j = 1i;

% variables
w = 2 * pi * f;
beta = w * sqrt(mu0 * epsilon0);
d = L/(N + 1);

% initiate arrays
z_n = zeros(N + 2, 1);
z_m = zeros(M + 2, 1);
v_m = zeros(N, 1);

% fill z and z' matrices with locations
for n = 1:N

    z_n(n + 1) = z_n(n) + d;
    z_m(n + 1) = z_m(n) + d;

end

    for m = 1:M

        for n = 1:N

            Z(m,n) = j * w * mu0 / (4 * pi) * (psi_funct(1, n, 0, m) + 1 / (beta^2 * d) * psi_funct
    (2, n, 0, m));

        end

    end

for n_theta = 1:length(theta_incident)
```

```

    theta = theta_incident(n_theta);

    % V matrix
    for n = 1:N

        v_m(n) = E0 * exp(j * beta * z_m(n) * cos(theta)) * sin(theta);

        % E_inc,z = -E_theta(0) * sin(theta) * exp(j * beta * z' *
        % cos(theta))
        % v_m = E_inc,v (z'_m)

    end

    % I matrix [Z_mn][I_n] = [V_m]
    I_N = Z \ v_m;

    % Radiation
    for n = 1:N

        SRad(n) = s_rad(I_N, theta_plot(n));

    end

% Current Magnitude
figure;
plot(z_n(2:length(z_n) - 1), abs(I_N)*10^3);
xlabel("Z position [m]");
ylabel("Current [mA]");
title("(|I(z)|)")
subtitle("\theta_{inc} = " + theta * 180/pi + "^{\circ}");
grid on;

% Current Phase
figure;
plot(z_n(2:length(z_n) - 1), angle(I_N) * 180 / pi);
xlabel("Z position [m]");
ylabel("Phase [degree]");
title("Phase of I(z)");
subtitle("\theta_{inc} = " + theta * 180/pi + "^{\circ}")
grid on;

fprintf('Maximum |I|_{max} = %.3f mA for \theta_{inc} = %.3f^{\circ}', max(abs(I_N)) * 1e3, theta * 180/pi);

end
```

Listing 2: MATLAB function for radiation density

```
function s_rad = s_rad_function(I, theta)

    global w mu0 E0

    s_rad = (w * mu0 * sin(theta) * abs(Q(I, theta)))^2 / (4 * pi * E0^2);

end
```

Listing 3: MATLAB function for Q

```
function Q = Q_function(I, theta)

    global N d

    Q = 0;

    for n = 2:N

        Q = Q + I(n - 1) * psi_funct(3, n, theta, 0) + (I(n) - I(n - 1)) / d * psi_funct(4, n,
    theta, 0);

    end

end
```

Listing 4: MATLAB function psi_funct

```
function val = psi_funct(type, n, theta, m)

    global epsilon0 mu0 w beta d j a

    switch type

        case 1

            if m ~= n         % ~= means logic NOT

                R = @(x) sqrt(( (n - m) * d - x ).^2 + a.^2);
                psi_1_function = @(x) (1 - abs(x) / d) .* (exp (-j * beta * R(x)) ./ R(x));

                val = integral(psi_1_function, -d, d);

            else

                R = @(x) sqrt(x.^2 + a^2);
                psi_1_function1 = @(x) (1 - x ./ d) .* exp(-j * beta * R(x)) ./ R(x) - cos(beta * a
    ) ./ R(x);

                val = 2 * integral(psi_1_function1, 0, d) + 2 * cos(beta * a) * log ((d + sqrt(d^2
    + a^2)) / a);

            end

        case 2

            R0 = sqrt(((n - m) * d)^2 + a^2);
            R1 = sqrt(((n - m - 1) * d)^2 + a^2);
            R2 = sqrt(((n + 1 - m) * d)^2 + a^2);


            val = exp(-j * beta * R1) / R1 - 2 * exp(-j * beta * R0) / R0 + exp(-j * beta * R2) /
    R2;

        case 3

            psi_3_function = @(x) exp(j * beta * ((n - 1) * d + x) * cos(theta));

            val = integral(psi_3_function, 0, d);

```

```
        case 4

            psi_4_function = @(x) x .* exp(j * beta * ((n - 1) * d + x) * cos(theta));

            val = integral(psi_4_function, 0, d);

    end

end
```

Listing 5: MATLAB script for monostatic RCS vs. length

```
clc
clear

% Given Parameters
global L a E0 N M beta w d epsilon0 mu0 j

f = 300e6;
N = 50;
M = N;
L = linspace(0.25, 2, N);

theta = pi/2;
theta_plot = linspace(0, 2*pi, N);
E0 = 1;

% Constants
mu0 = 4 * pi * 10^-7;
epsilon0 = 8.854 * 10^-12;
j = 1i;

% variables
w = 2 * pi * f;
beta = w * sqrt(mu0 * epsilon0);

% initiate arrays
z_n = zeros(N + 2, 1);
z_m = zeros(M + 2, 1);
v_m = zeros(N, 1);

for i = 1:N

    d = L(i)/(N + 1);
    a = L(i)/160;

    % fill z and z' matrices with locations
    for n = 1:N

        z_n(n + 1) = z_n(n) + d;
        z_m(n + 1) = z_m(n) + d;

    end

    for m = 1:M

        for n = 1:N

            Z(m,n) = j * w * mu0 / (4 * pi) * (psi_funct(1, n, 0, m) + 1 / (beta^2 * d) * psi_funct
```

```
    (2, n, 0, m));

            end

    end


    % V matrix
    for n = 1:N

        v_m(n) = -E0 * exp(j * beta * z_m(n) * cos(theta)) * sin(theta);

        % E_inc,z = -E_theta(0) * sin(theta) * exp(j * beta * z' *
        % cos(theta))
        % v_m = E_inc,v (z'_m)

    end

    % I matrix [Z_mn][I_n] = [V_m]
    I_N = Z \ v_m;

    SRad(i) = s_rad(I_N, pi/2);

    % Plot Radiation pattern cross section
    %{
    figure;
    polarplot(abs(SRad));
    hold on;
    polarplot([theta; theta], [0; max(abs(SRad))]);
    legend("S_{rad}", "\Theta_{inc}");
    title("Length = " + L(i) + ", radius =" + a + " & \theta_{inc} = " + theta * 180/pi + " degrees ");
    hold off;
    %}
end

figure;
plot(L, SRad, '-o');
xlabel('Length (m)');
ylabel('radiation magnitude at \theta_{inc} = 90^{\circ} (U)')
title("Monostatic radar cross-section of the scatterer for L ranging from 0.25m to 2m");
```