Project 2

ECE 541 – Applied Electromagnetics

Elim Aschenberg

9/27/2025

I. MoM Analysis of an Air-Filled Microstrip Transmission Line with a Finite Ground Plane – Using Point Matching Testing Procedure

1. Write the theory and equations, and describe your MoM algorithm.

Theory:

The theory of Point matching for a finite ground plane is that a conductor centered above a finite ground plane is segmented into N1 segments. The bottom is segmented into N2 segments. Both together create N segments. Each individual segment needs to be compared to every other segment, regardless of whether it is on the same plane or not. Point matching takes the center point for every segment and then compares every other center point to that one.

Equations:

$$Z_{ij} = \frac{1}{2\pi\varepsilon_0} K_1\left(\frac{\Delta}{2},\, x_i - x_j\,,\, y_i - y_j\right)$$

$K_1 = Re\{(z - z')[\ln(z - z') - 1]\}$ evaluated at z' = -a to a

$$\Delta = \frac{w1}{N}$$

$$N2 = \frac{w2}{\Delta}$$

$$[\rho_{sj}] = [V_i][Z_{ij}]^{-1}$$

$$Q' = \sum_{j=1}^{N} \rho_{sj}\, \Delta l_j$$

$$C' = \frac{Q'}{v_1 - v_2}$$

Empirical Formulas:

$$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[\left(1 + 12\frac{h}{w}\right)^{-1/2} + p\right]$$

$$Z_0 = \frac{\eta_0}{2\pi\sqrt{\varepsilon_{reff}}} \ln\left(\frac{8h}{w} + \frac{w}{4h}\right) \quad \text{for} \quad \frac{w}{h} \le 1,$$

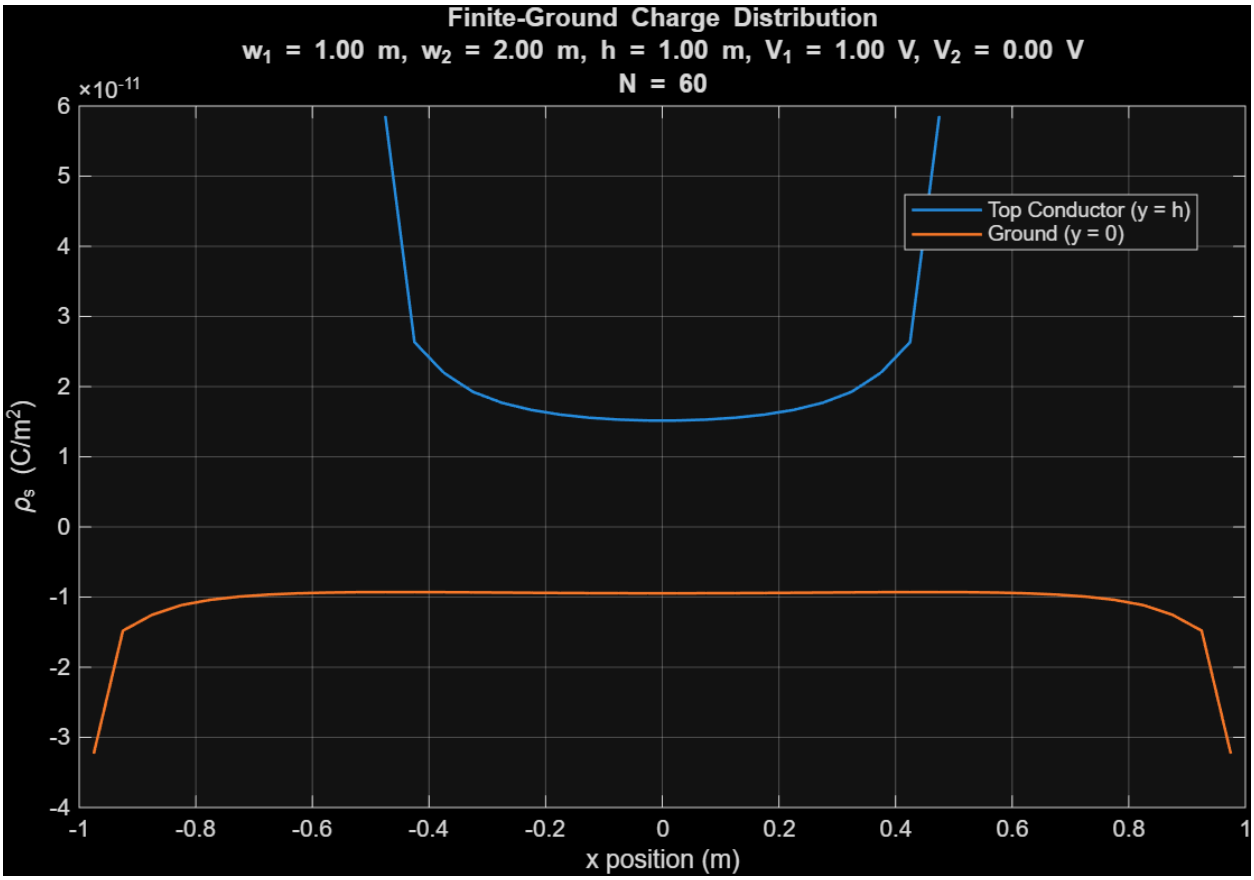
$$Z_0 = \frac{\eta_0}{\sqrt{\varepsilon_{reff}}}\left[\frac{w}{h} + 1.393 + 0.667 \ln\left(\frac{w}{h} + 1.444\right)\right]^{-1} \quad \text{for} \quad \frac{w}{h} > 1,$$

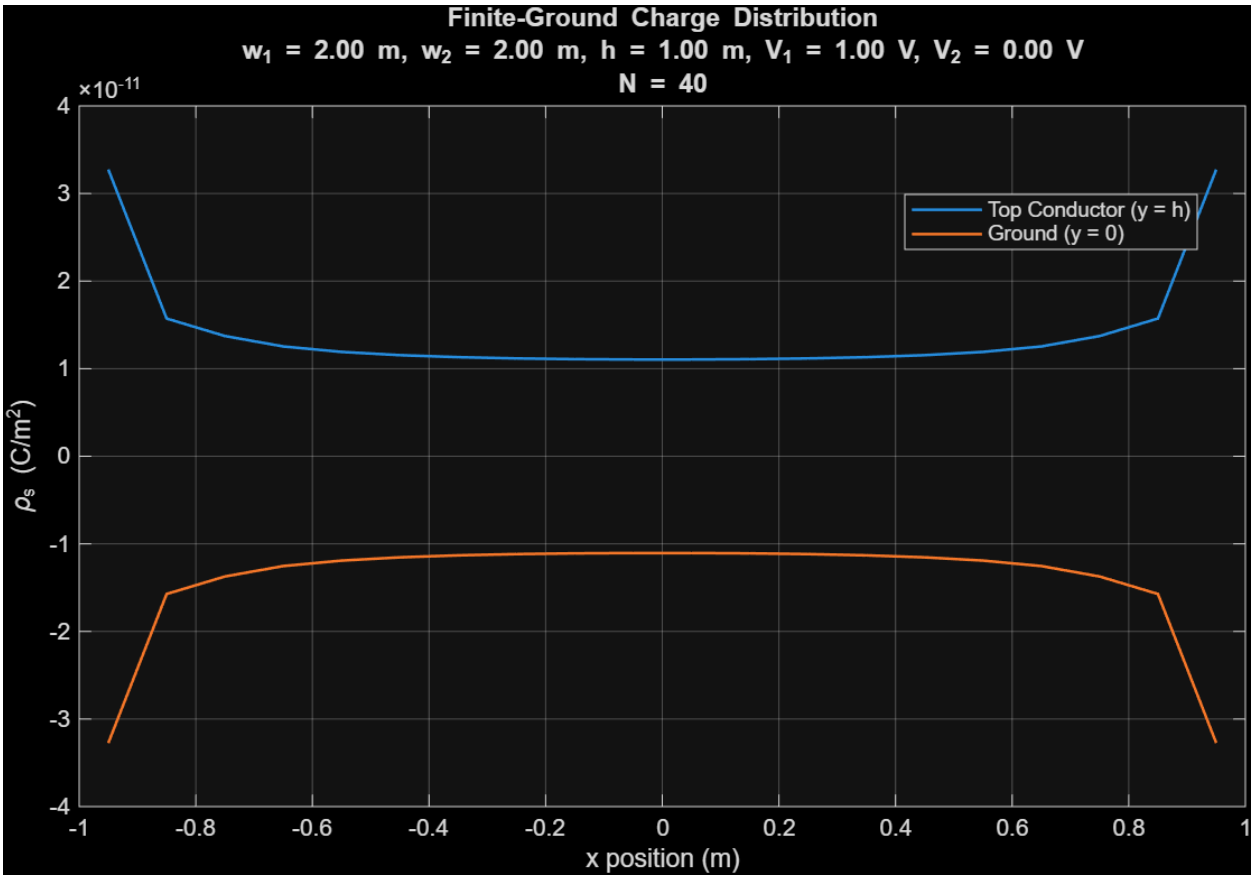
Find and plot the distribution of surface charges, ρs, along the upper strip and the ground strip, respectively, for several characteristic values of w1, w2, and h, and several values of N.

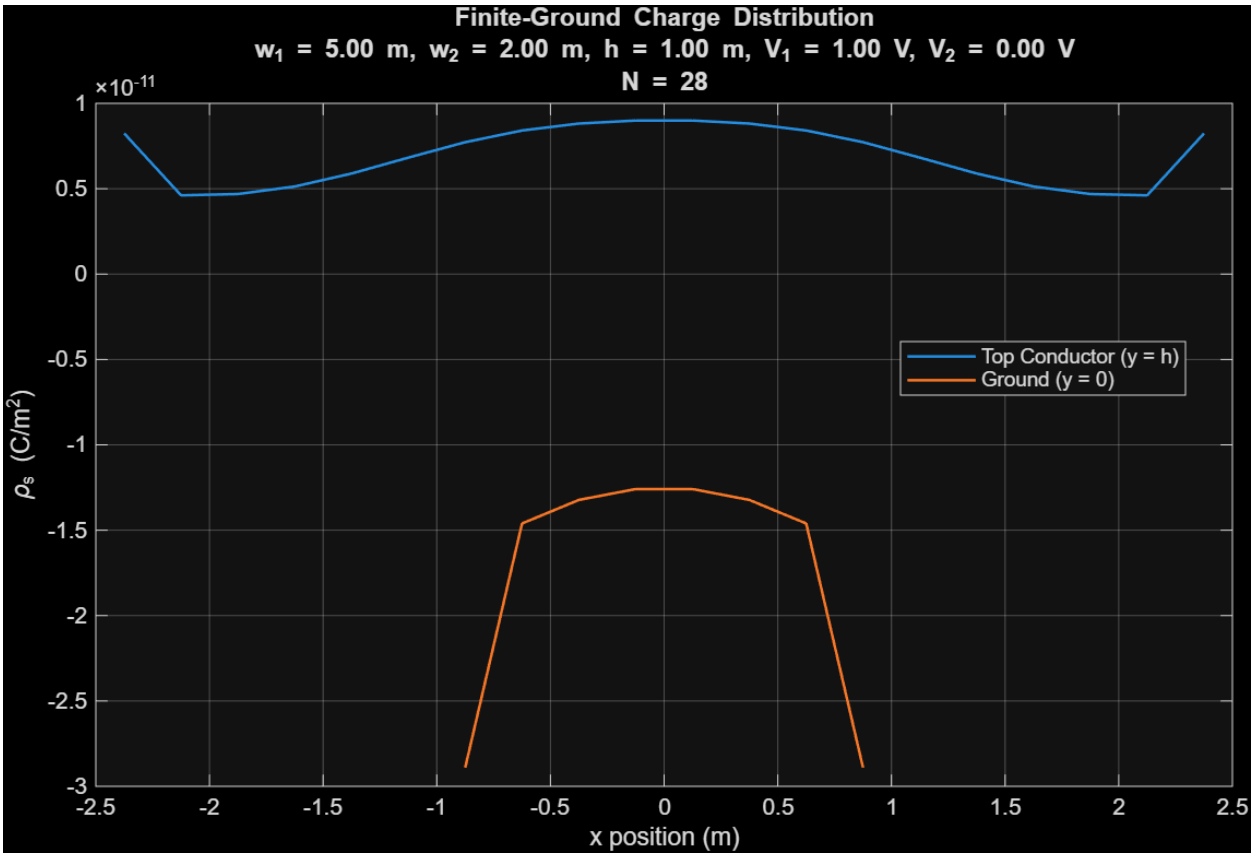
Point Matching (Infinite Ground):

In the point-matching method, the top conducting strip is divided into *N* equal segments, each having a width equal to the strip width divided by *N*. A voltage of one volt is applied to the strip, while the infinite ground plane is modeled using the image method and is held at zero volts at a distance of twice the height below the strip. The potential at the center of each segment (called the "matching points") is forced to equal the applied voltage by constructing an *N-by-N* impedance matrix, where each element represents the potential at one segment caused by a unit charge on another. The function named **K1_function** calculates this interaction and includes both the direct and image effects. Solving the linear equation for charge density gives the amount of charge on each segment. The total charge per unit length is then found by summing the charges on all segments, and dividing this by the applied voltage difference gives the capacitance per unit length. The process is repeated for several values of *N* to observe how the solution converges toward the capacitance predicted by the empirical formula.

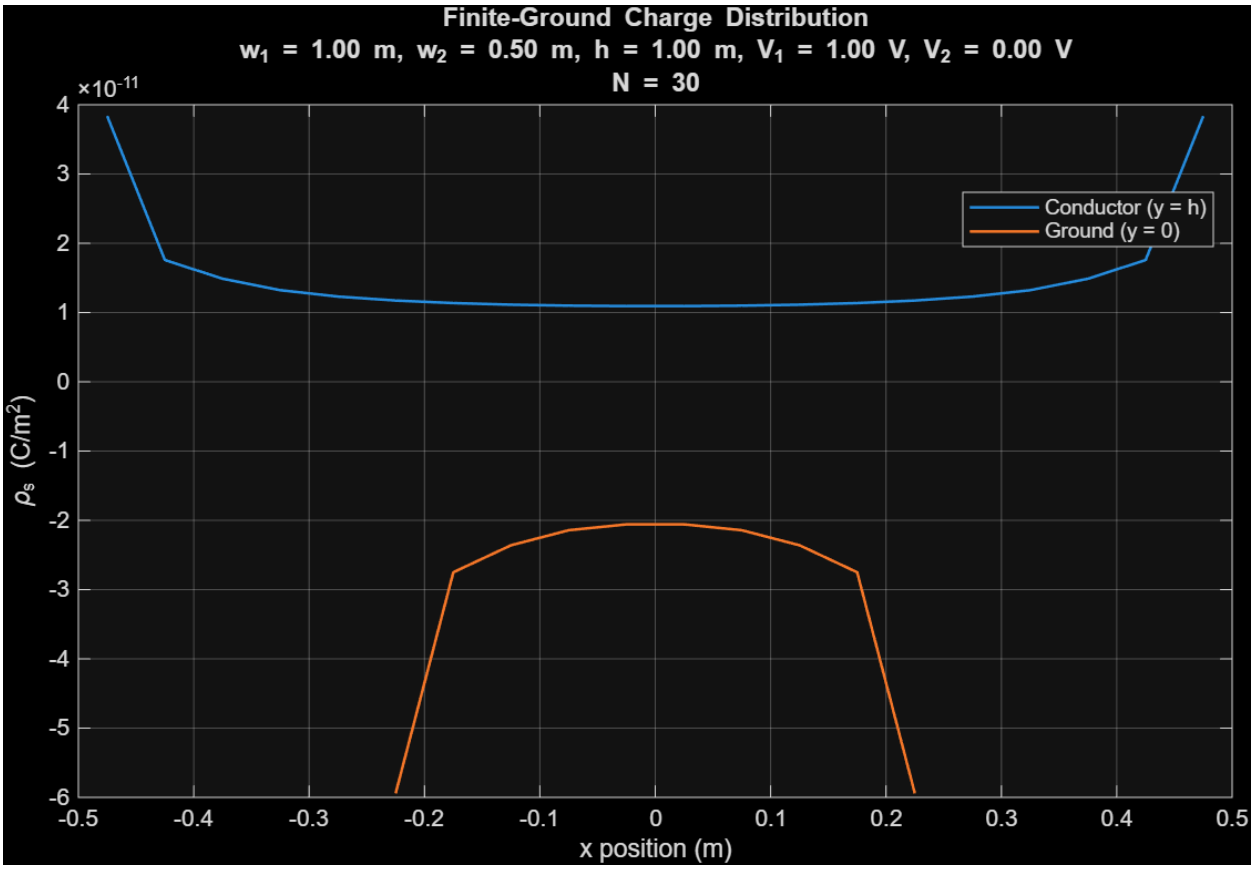
Variations in W1:

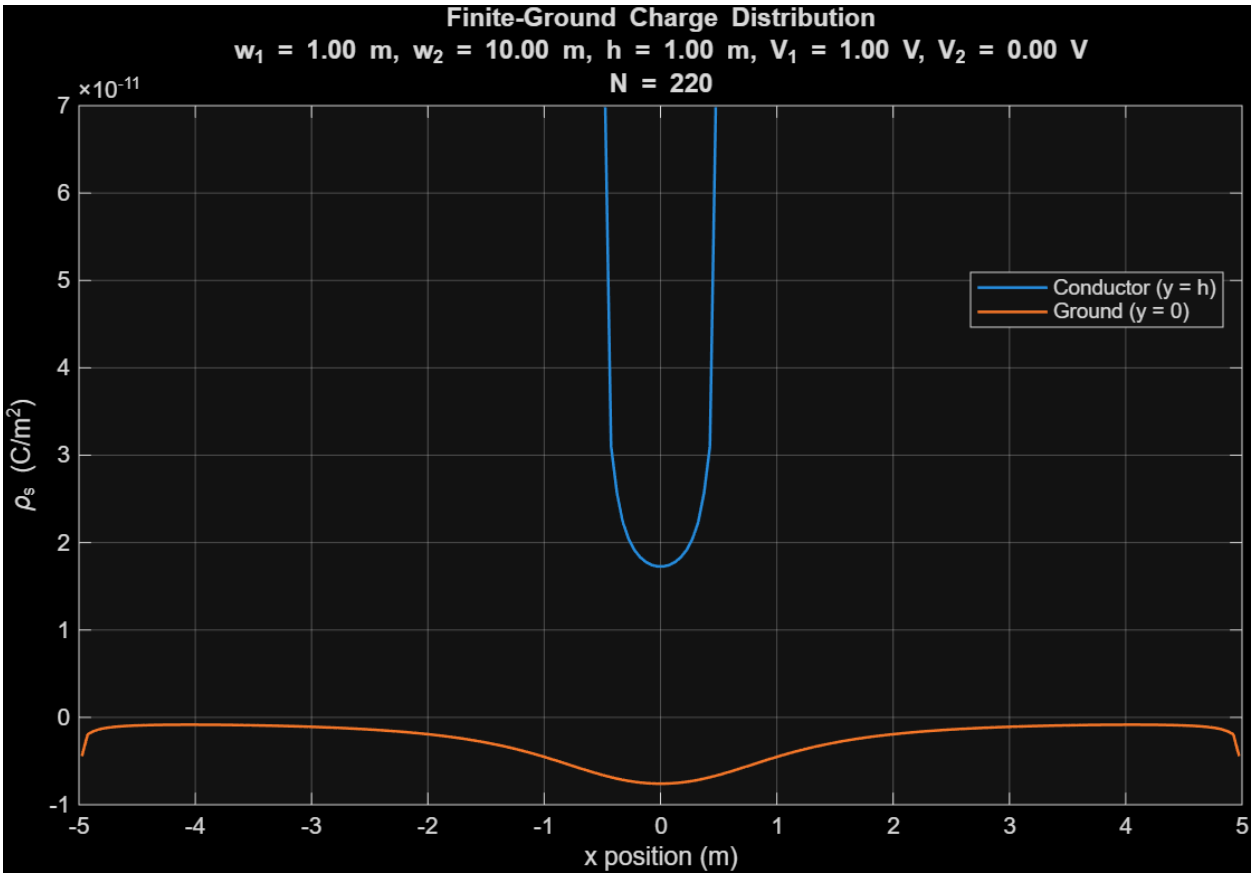


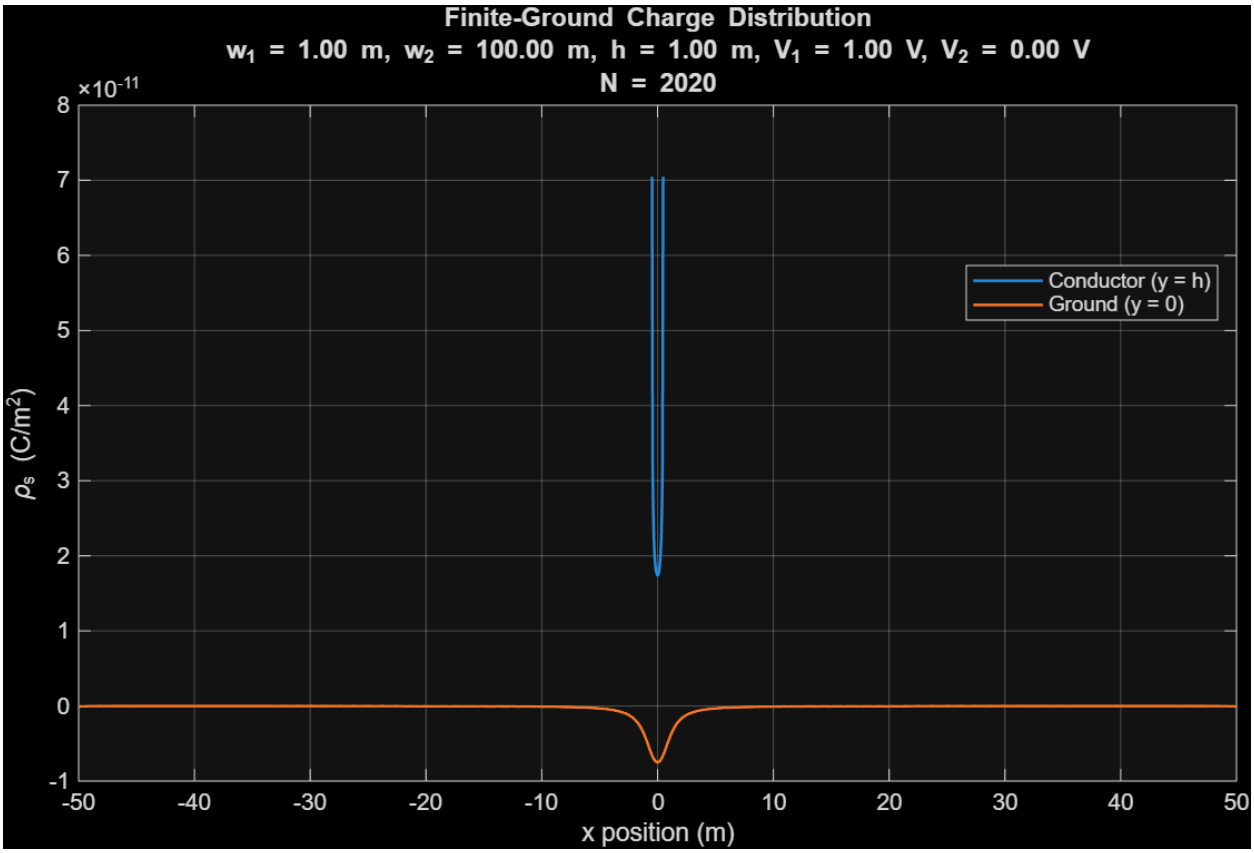

Finite-Ground Charge Distribution
w1 = 2.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 40
×10-11
4
3
2
1
0
-1
-2
-3
-4
ρs (C/m2)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x position (m)
Top Conductor (y = h)
Ground (y = 0)



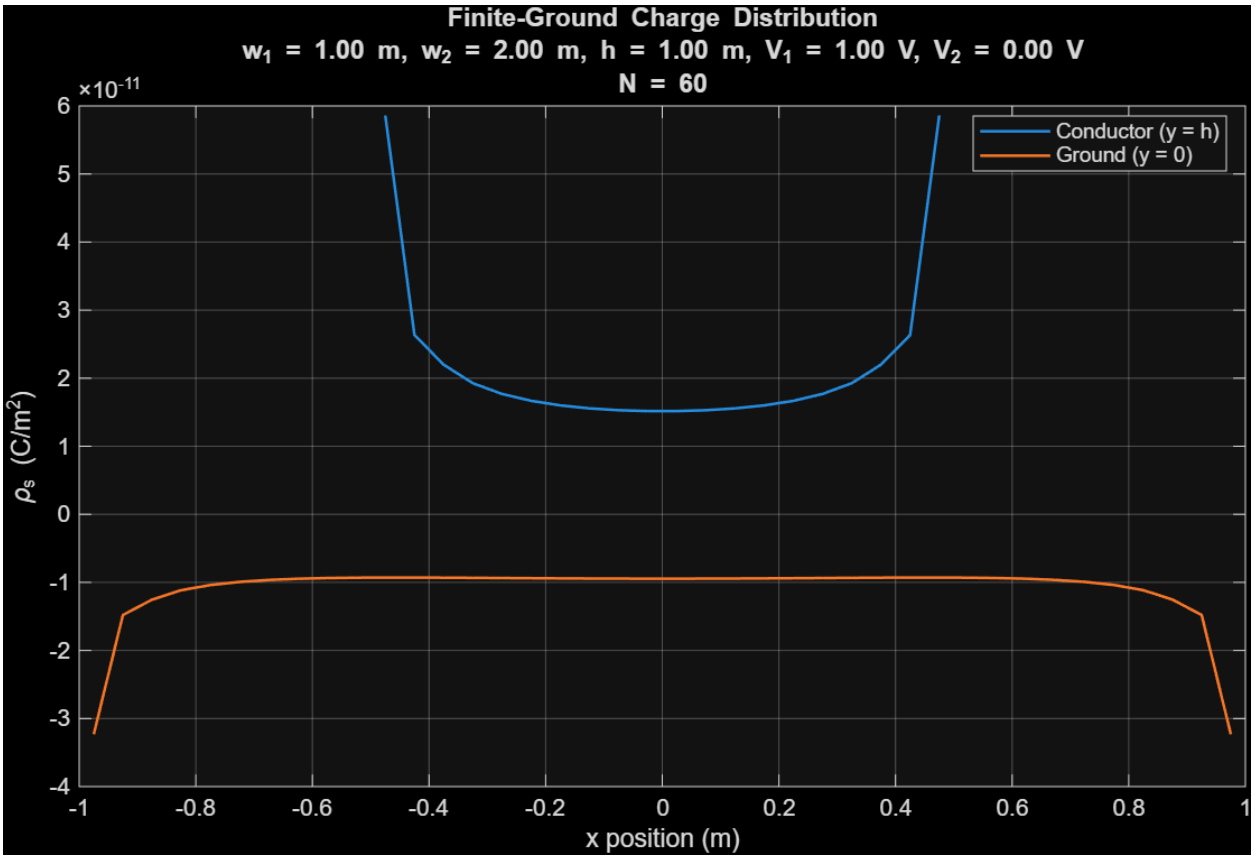
Variations in W2:

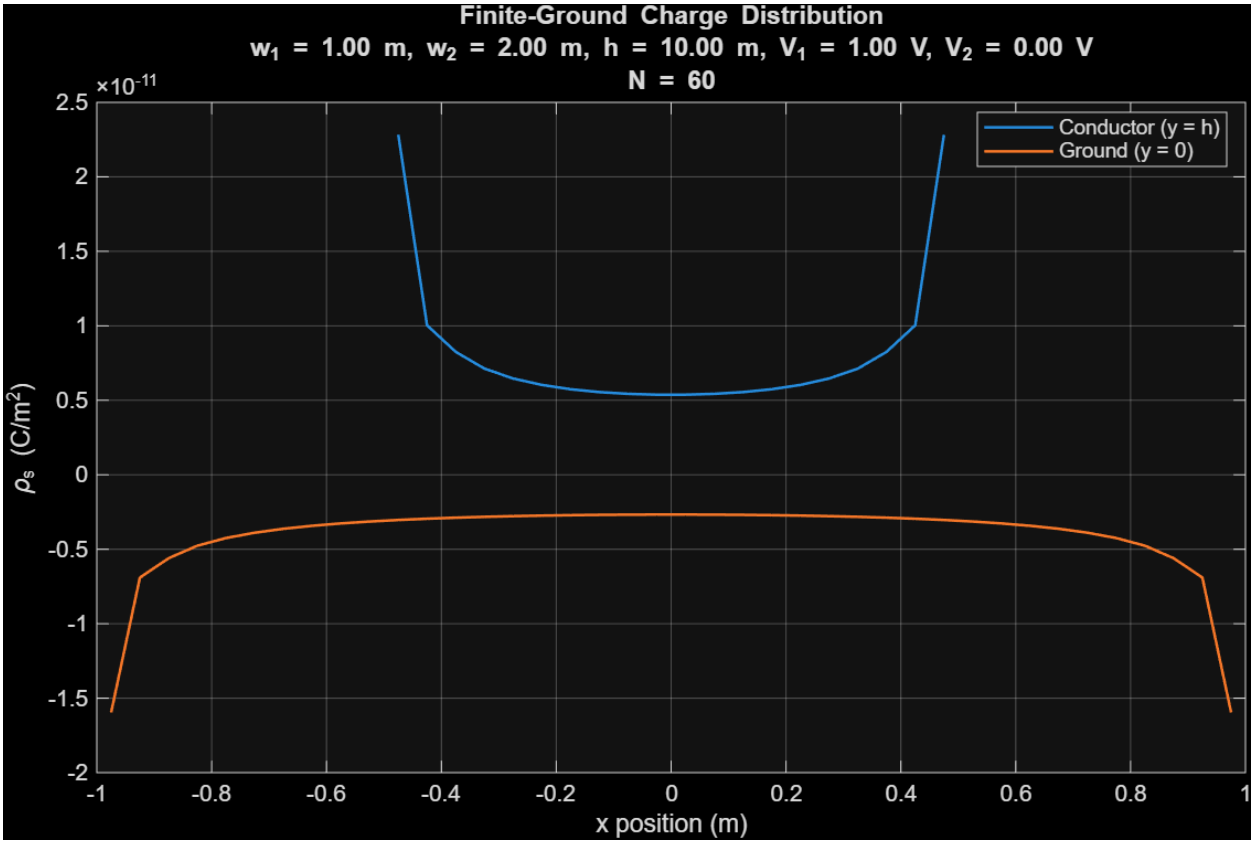

Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 0.50 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 30
×10-11
4
3
2
1
0
-1
-2
-3
-4
-5
-6
ρs (C/m2)
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
x position (m)
Conductor (y = h)
Ground (y = 0)

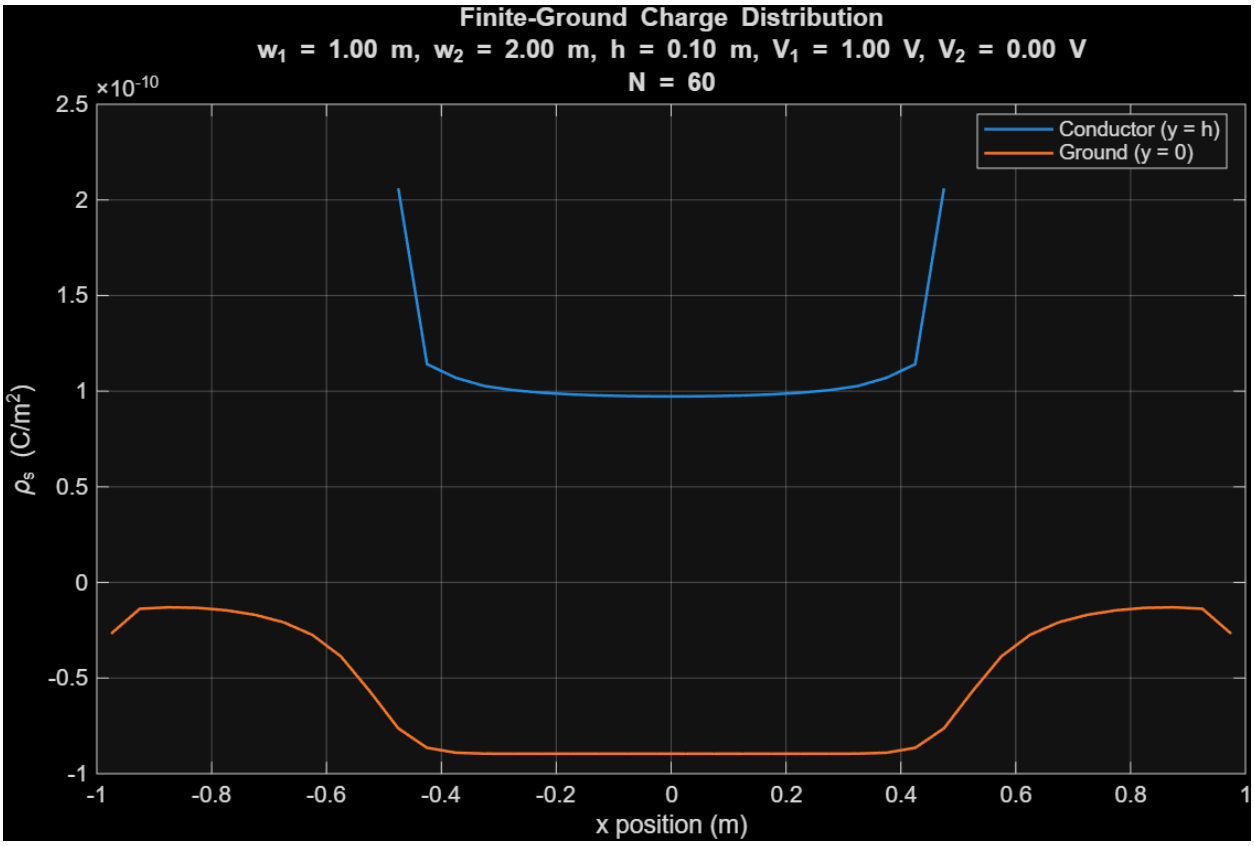

Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 10.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 220
×10-11
7
6
5
4
3
2
1
0
-1
ρs (C/m2)
-5
-4
-3
-2
-1
0
1
2
3
4
5
x position (m)
Conductor (y = h)
Ground (y = 0)



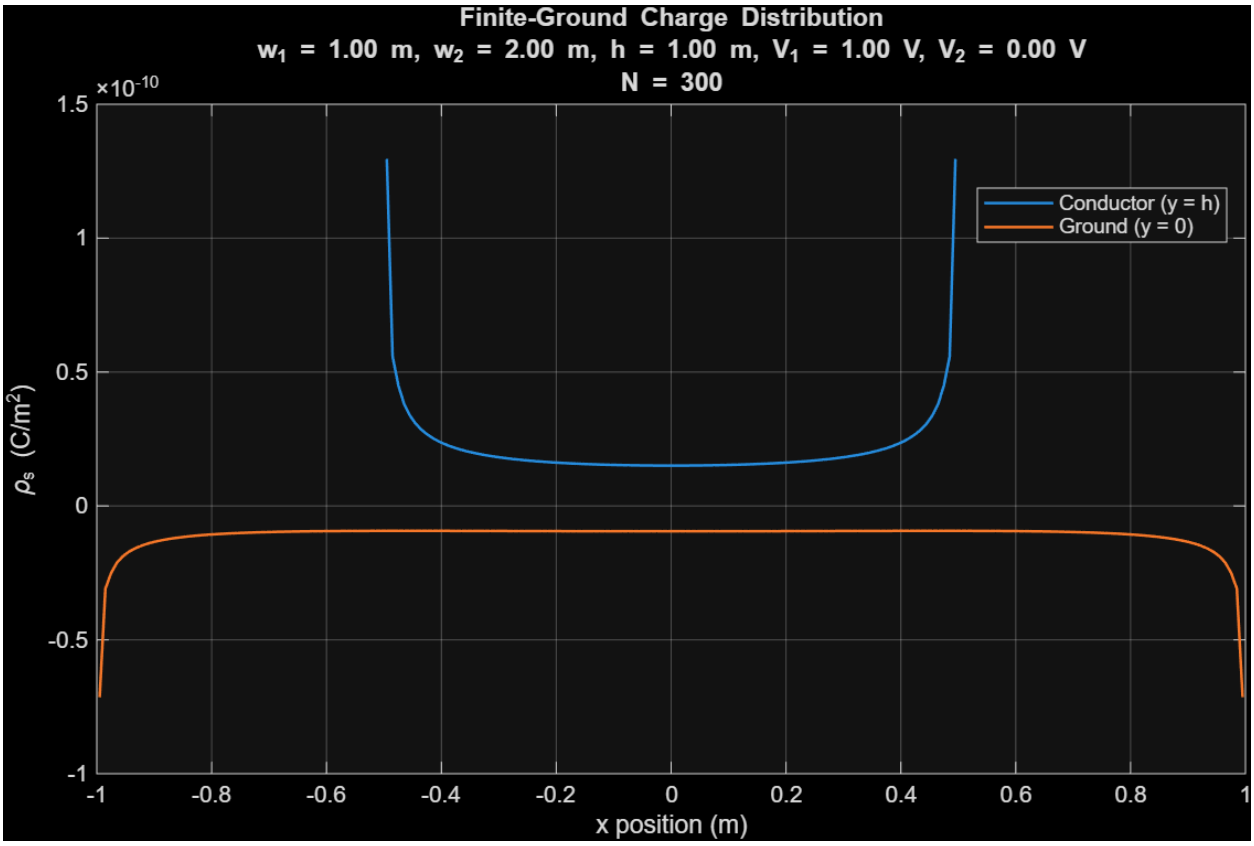
Different values of h:

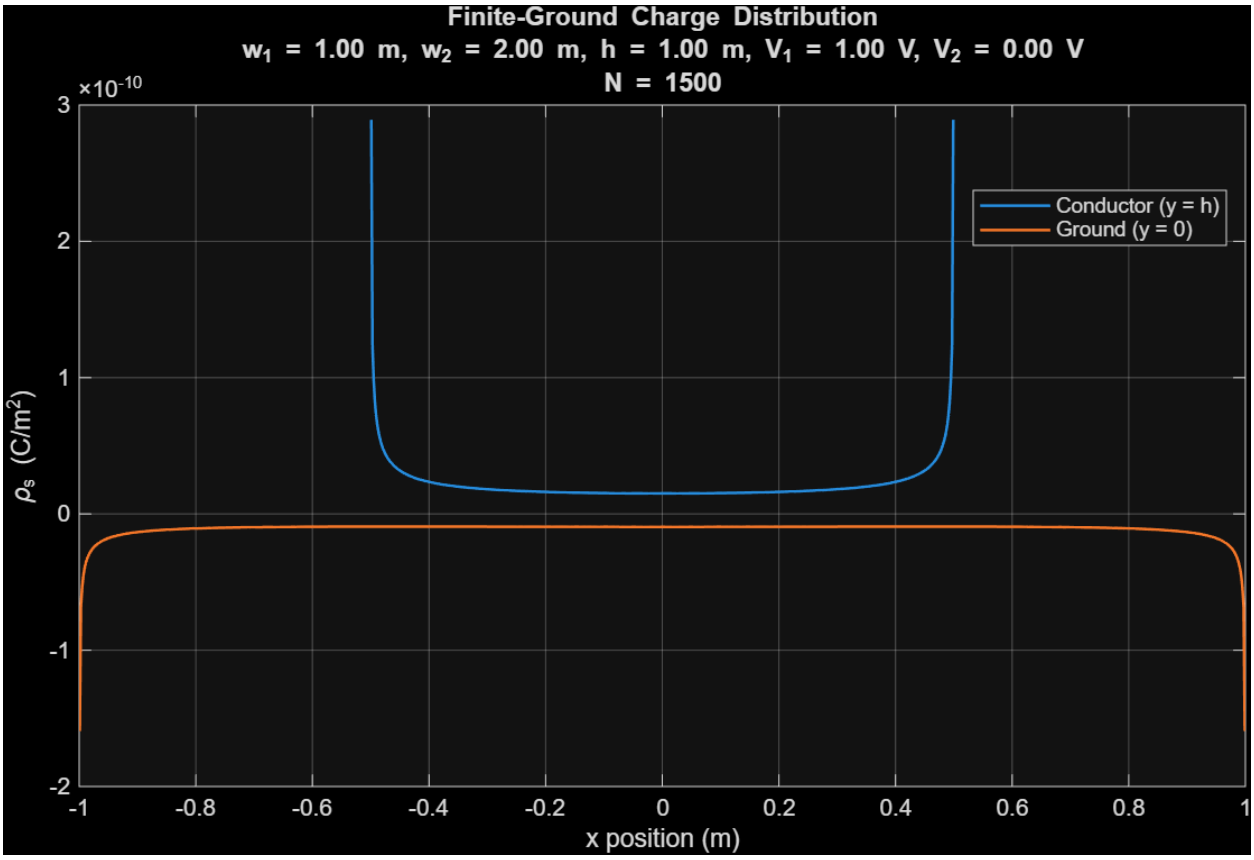

Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 60
×10-11
Conductor (y = h)
Ground (y = 0)
ρs (C/m2)
x position (m)

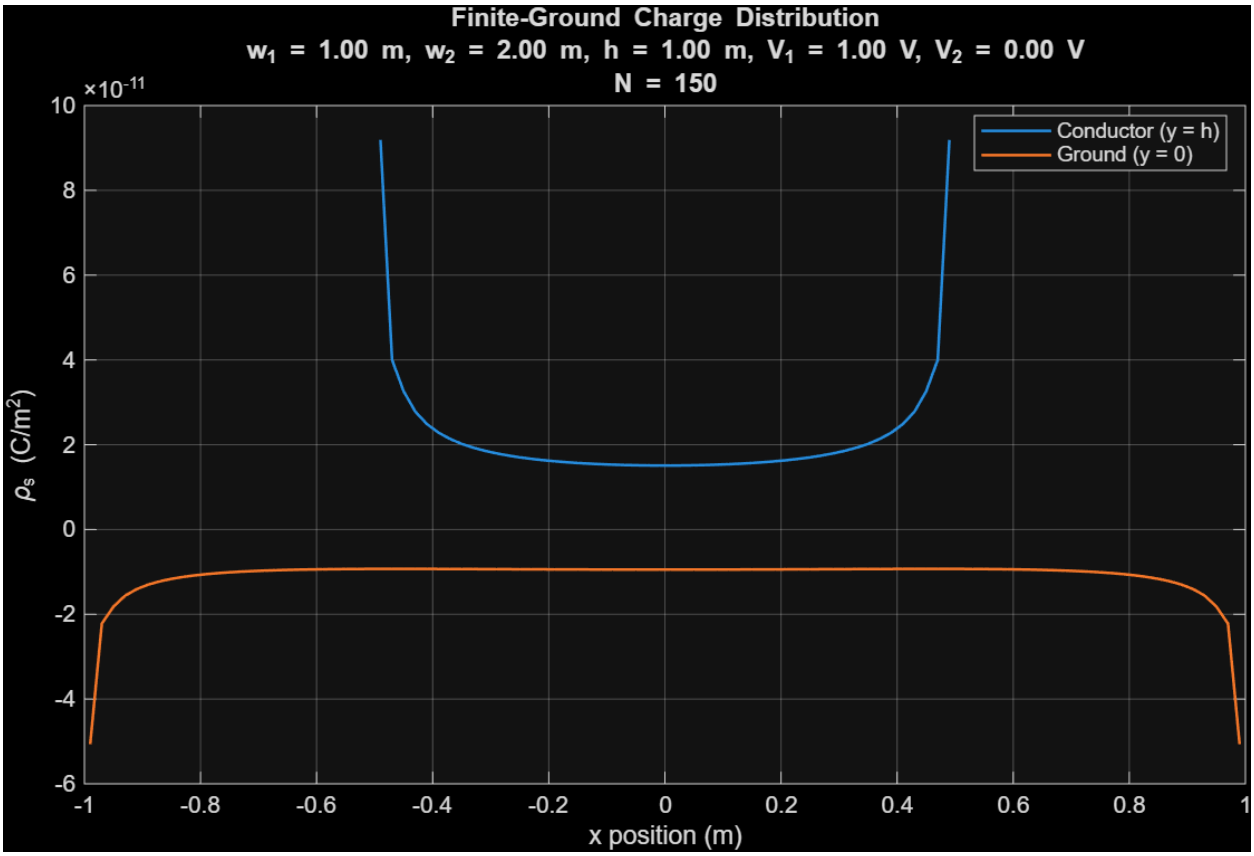

Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 10.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 60
×10-11
Conductor (y = h)
Ground (y = 0)
ρs (C/m2)
x position (m)

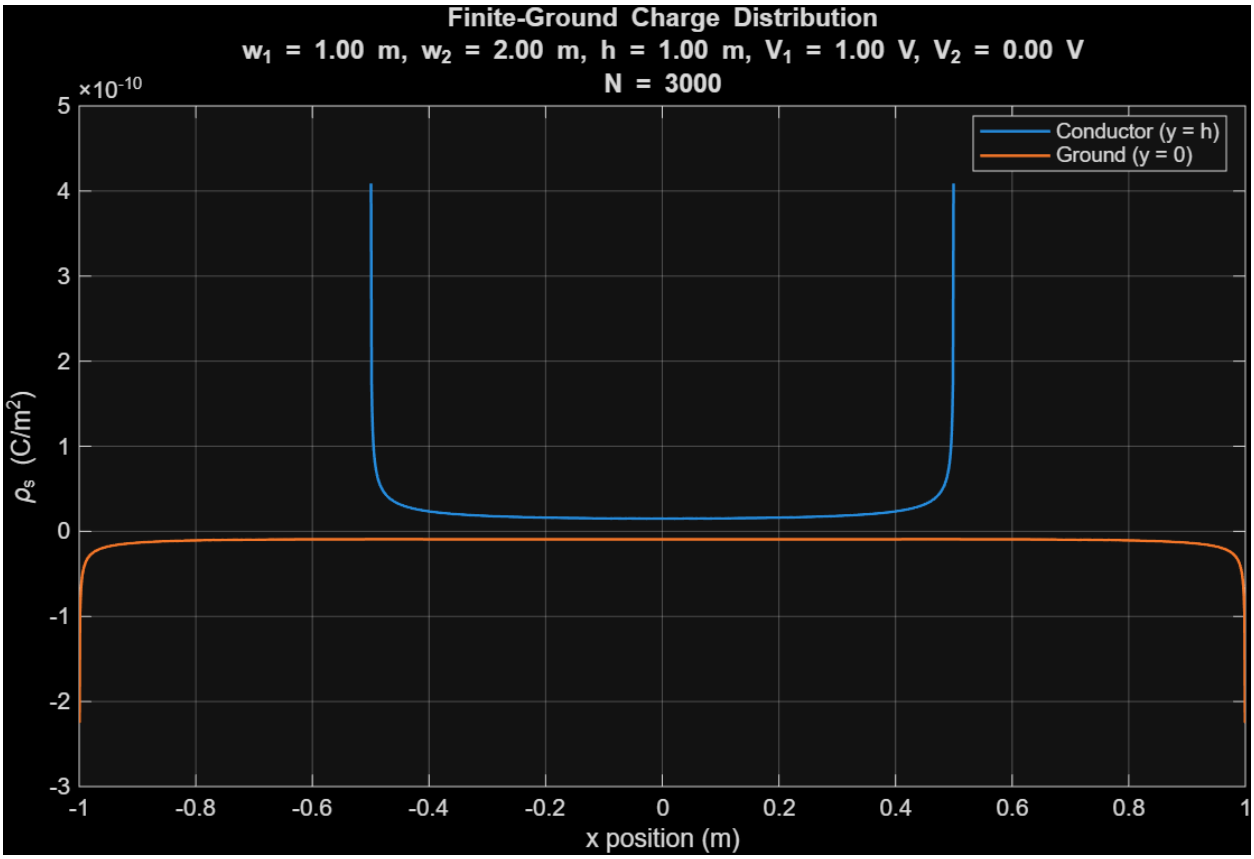


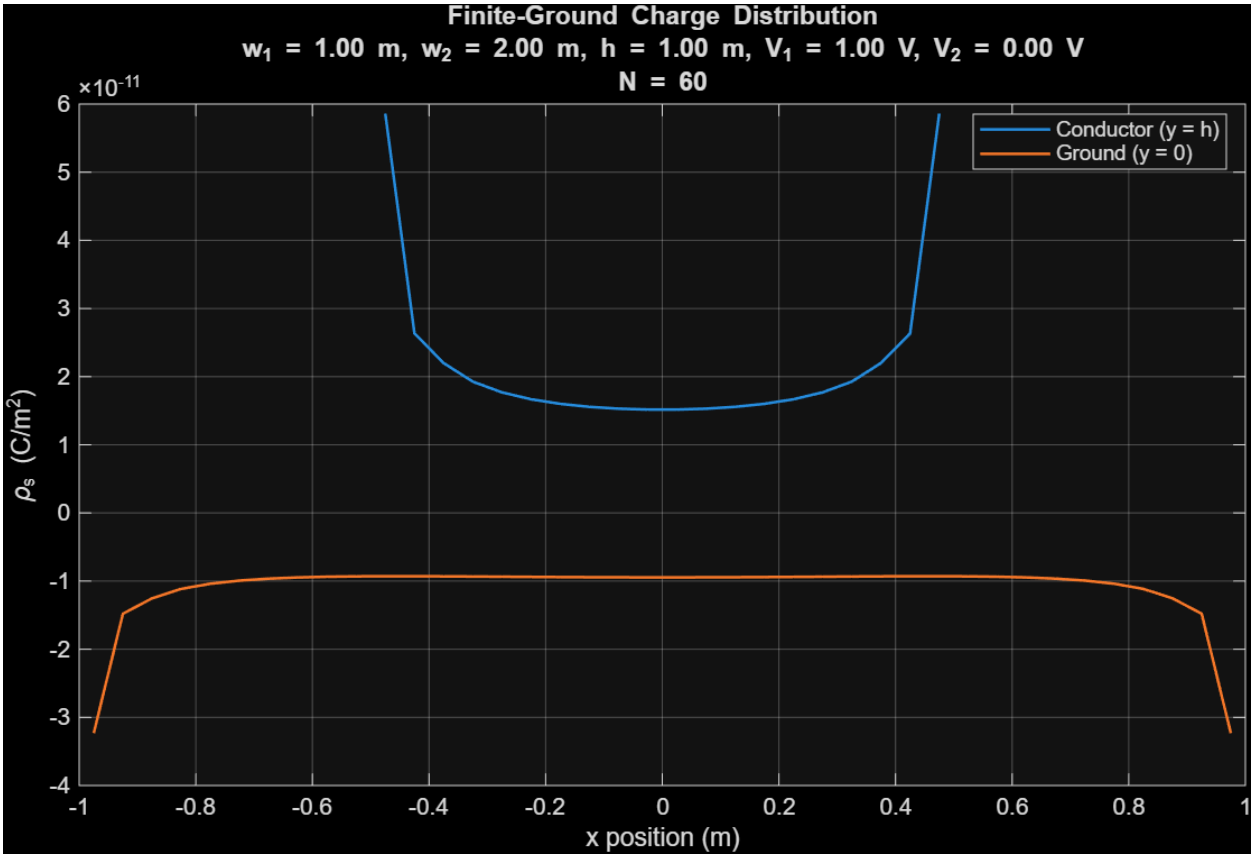
For several different values of N:


Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 300
×10-10
1.5
1
0.5
0
-0.5
-1
ρs (C/m2)
Conductor (y = h)
Ground (y = 0)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x position (m)

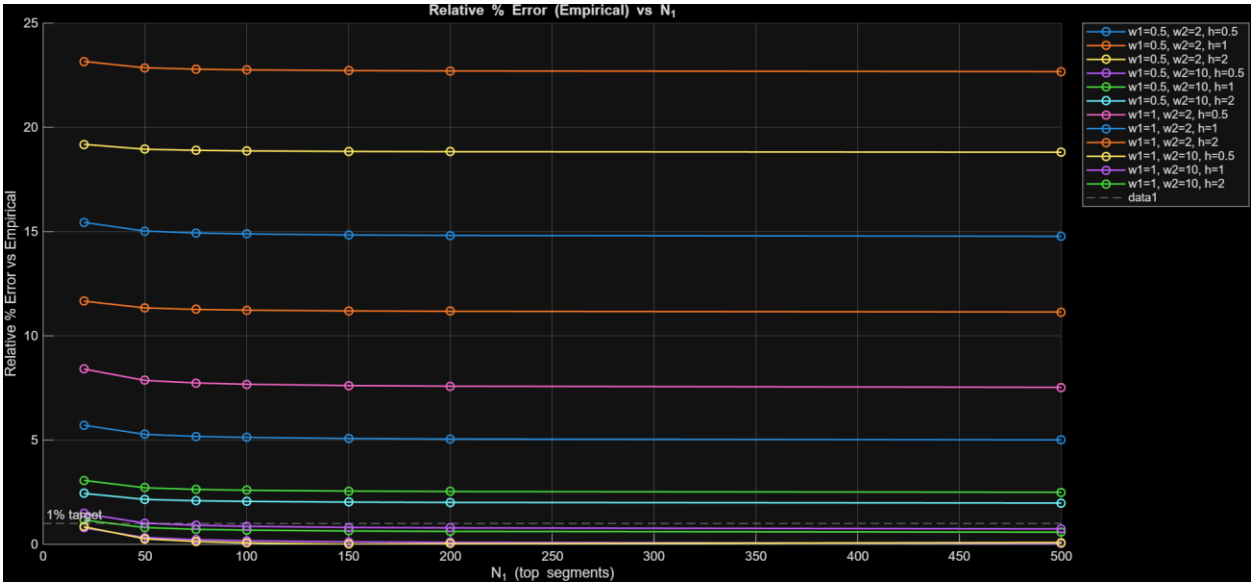

Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 1500
×10-10
3
2
1
0
-1
-2
ρs (C/m2)
Conductor (y = h)
Ground (y = 0)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x position (m)

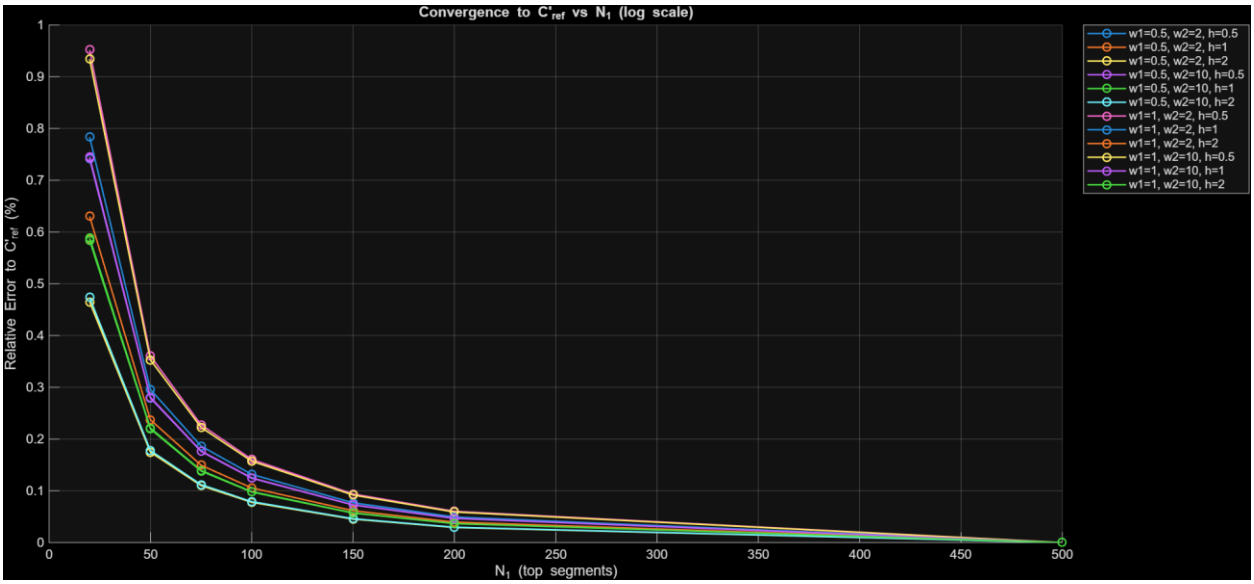

Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 150
×10-11
10
8
6
4
2
0
-2
-4
-6
ρs (C/m2)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x position (m)
Conductor (y = h)
Ground (y = 0)


Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 3000
×10-10
Conductor (y = h)
Ground (y = 0)
ρs (C/m2)
x position (m)


Finite-Ground Charge Distribution
w1 = 1.00 m, w2 = 2.00 m, h = 1.00 m, V1 = 1.00 V, V2 = 0.00 V
N = 60
×10-11
Conductor (y = h)
Ground (y = 0)
ρs (C/m2)
x position (m)

For several characteristic values of w1, w2, and h, find the capacitance per unit length of the transmission line, C', and tabulate (in a table) and plot (as a graph) the relative percentage error when compared to the result obtained by empirical formulas (p.569, Electromagnetics, Notaros— attached here) as a function of N. Discuss the convergence of MoM results with increasing N.





w1_m	w2_used_m	h_m	N1	N2	N_total	Cprime_MoM_Fperm	Cprime_Emp_Fperm	Error_percent
0.5	2	0.5	20	80	100	2.4822e-11	2.6326e-11	5.7125
0.5	2	0.5	50	200	250	2.4938e-11	2.6326e-11	5.2735
0.5	2	0.5	75	300	375	2.4964e-11	2.6326e-11	5.1754
0.5	2	0.5	100	400	500	2.4977e-11	2.6326e-11	5.1263
0.5	2	0.5	150	600	750	2.499e-11	2.6326e-11	5.0771
0.5	2	0.5	200	800	1000	2.4996e-11	2.6326e-11	5.0525
0.5	2	0.5	500	2000	2500	2.5008e-11	2.6326e-11	5.0082
0.5	2	1	20	80	100	1.765e-11	1.9981e-11	11.665
0.5	2	1	50	200	250	1.7715e-11	1.9981e-11	11.34
0.5	2	1	75	300	375	1.773e-11	1.9981e-11	11.267
0.5	2	1	100	400	500	1.7737e-11	1.9981e-11	11.231
0.5	2	1	150	600	750	1.7744e-11	1.9981e-11	11.194
0.5	2	1	200	800	1000	1.7748e-11	1.9981e-11	11.176
0.5	2	1	500	2000	2500	1.7754e-11	1.9981e-11	11.144
0.5	2	2	20	80	100	1.2947e-11	1.6021e-11	19.185
0.5	2	2	50	200	250	1.2985e-11	1.6021e-11	18.949
0.5	2	2	75	300	375	1.2993e-11	1.6021e-11	18.897
0.5	2	2	100	400	500	1.2997e-11	1.6021e-11	18.871
0.5	2	2	150	600	750	1.3002e-11	1.6021e-11	18.844
0.5	2	2	200	800	1000	1.3004e-11	1.6021e-11	18.831
0.5	2	2	500	2000	2500	1.3007e-11	1.6021e-11	18.808
0.5	10	0.5	20	400	420	2.6118e-11	2.6326e-11	0.79157
0.5	10	0.5	50	1000	1050	2.624e-11	2.6326e-11	0.32707
0.5	10	0.5	75	1500	1575	2.6268e-11	2.6326e-11	0.22331
0.5	10	0.5	100	2000	2100	2.6281e-11	2.6326e-11	0.17136
0.5	10	0.5	150	3000	3150	2.6295e-11	2.6326e-11	0.11936
0.5	10	0.5	200	4000	4200	2.6302e-11	2.6326e-11	0.093344
0.5	10	0.5	500	10000	10500	2.6314e-11	2.6326e-11	0.046482
0.5	10	1	20	400	420	1.9748e-11	1.9981e-11	1.1672
0.5	10	1	50	1000	1050	1.9821e-11	1.9981e-11	0.80202
0.5	10	1	75	1500	1575	1.9837e-11	1.9981e-11	0.72062
0.5	10	1	100	2000	2100	1.9845e-11	1.9981e-11	0.67988
0.5	10	1	150	3000	3150	1.9853e-11	1.9981e-11	0.63913
0.5	10	1	200	4000	4200	1.9857e-11	1.9981e-11	0.61874
0.5	10	1	500	10000	10500	1.9865e-11	1.9981e-11	0.58203
0.5	10	2	20	400	420	1.5629e-11	1.6021e-11	2.4437
0.5	10	2	50	1000	1050	1.5675e-11	1.6021e-11	2.1539
0.5	10	2	75	1500	1575	1.5686e-11	1.6021e-11	2.0894
0.5	10	2	100	2000	2100	1.5691e-11	1.6021e-11	2.0571
0.5	10	2	150	3000	3150	1.5696e-11	1.6021e-11	2.0248
0.5	10	2	200	4000	4200	1.5699e-11	1.6021e-11	2.0086
0.5	10	2	500	10000	10500	1.5703e-11	1.6021e-11	1.9796
1	2	0.5	20	40	60	3.4157e-11	3.7293e-11	8.4089
1	2	0.5	50	100	150	3.4361e-11	3.7293e-11	7.8621
1	2	0.5	75	150	225	3.4407e-11	3.7293e-11	7.7391
1	2	0.5	100	200	300	3.443e-11	3.7293e-11	7.6774
1	2	0.5	150	300	450	3.4453e-11	3.7293e-11	7.6156
1	2	0.5	200	400	600	3.4464e-11	3.7293e-11	7.5846
1	2	0.5	500	1000	1500	3.4485e-11	3.7293e-11	7.5287
1	2	1	20	40	60	2.2261e-11	2.6326e-11	15.441
1	2	1	50	100	150	2.2371e-11	2.6326e-11	15.026
1	2	1	75	150	225	2.2395e-11	2.6326e-11	14.933
1	2	1	100	200	300	2.2407e-11	2.6326e-11	14.886
1	2	1	150	300	450	2.242e-11	2.6326e-11	14.839
1	2	1	200	400	600	2.2426e-11	2.6326e-11	14.816
1	2	1	500	1000	1500	2.2437e-11	2.6326e-11	14.774
1	2	2	20	40	60	1.5353e-11	1.9981e-11	23.161
1	2	2	50	100	150	1.5414e-11	1.9981e-11	22.857

1	2	2	75	150	225	1.5427e-11	1.9981e-11	22.789
1	2	2	100	200	300	1.5434e-11	1.9981e-11	22.755
1	2	2	150	300	450	1.5441e-11	1.9981e-11	22.721
1	2	2	200	400	600	1.5444e-11	1.9981e-11	22.704
1	2	2	500	1000	1500	1.545e-11	1.9981e-11	22.674
1	10	0.5	20	200	220	3.6978e-11	3.7293e-11	0.84543
1	10	0.5	50	500	550	3.7195e-11	3.7293e-11	0.2639
1	10	0.5	75	750	825	3.7243e-11	3.7293e-11	0.13345
1	10	0.5	100	1000	1100	3.7268e-11	3.7293e-11	0.068055
1	10	0.5	150	1500	1650	3.7292e-11	3.7293e-11	0.0025492
1	10	0.5	200	2000	2200	3.7304e-11	3.7293e-11	0.030246
1	10	0.5	500	5000	5500	3.7326e-11	3.7293e-11	0.089347
1	10	1	20	200	220	2.5938e-11	2.6326e-11	1.477
1	10	1	50	500	550	2.6059e-11	2.6326e-11	1.0174
1	10	1	75	750	825	2.6086e-11	2.6326e-11	0.91472
1	10	1	100	1000	1100	2.6099e-11	2.6326e-11	0.86331
1	10	1	150	1500	1650	2.6113e-11	2.6326e-11	0.81186
1	10	1	200	2000	2200	2.6119e-11	2.6326e-11	0.78612
1	10	1	500	5000	5500	2.6132e-11	2.6326e-11	0.73975
1	10	2	20	200	220	1.9368e-11	1.9981e-11	3.0667
1	10	2	50	500	550	1.9439e-11	1.9981e-11	2.7115
1	10	2	75	750	825	1.9455e-11	1.9981e-11	2.6323
1	10	2	100	1000	1100	1.9463e-11	1.9981e-11	2.5927
1	10	2	150	1500	1650	1.9471e-11	1.9981e-11	2.5531
1	10	2	200	2000	2200	1.9475e-11	1.9981e-11	2.5333
1	10	2	500	5000	5500	1.9482e-11	1.9981e-11	2.4975

As the number of segments N increases, the MoM results get more accurate and start to settle toward a consistent value. When the ground plane is wide enough, the MoM values line up really closely with the empirical formula—usually within a fraction of a percent. For smaller or more extreme geometries, the error levels off even if N keeps increasing, which means the difference comes from the model itself, not the resolution. Overall, increasing N makes the results smoother and more reliable until they stop changing much, showing that the solution has converged.

```
% ECE 541 – Applied Electromagnetics – Project 2 (Point Matching)
% Elim Aschenberg
% 10/9/2025
% Point Matching Method
clc; clear;
eps0 = 8.8541878188e-12;
er = 1;
eps = eps0 * er;
coef = 1/(2*pi*eps);
eta0 = 377;
c0 = 3e8;
v1 = 1;
v2 = 0;
w1 = 1;
w2 = 20;
```

```
N1 = 20;
h = 1;
delta = w1 / N1;
a = delta/2;
N2 = max(1, round(w2 / delta));
w2 = N2 * delta;
N = N1 + N2;
x1 = linspace(-w1/2 + a, +w1/2 - a, N1).'; y1 = h*ones(N1,1);
x2 = linspace(-w2/2 + a, +w2/2 - a, N2).'; y2 = zeros(N2,1);
Z1_self = zeros(N1,N1);
for i = 1:N1
for j = 1:N1
Z1_self(i,j) = coef * K1_function(a, x1(i)-x1(j), y1(i)-y1(j));
end
end
Z2_self = zeros(N2,N2);
for i = 1:N2
for j = 1:N2
Z2_self(i,j) = coef * K1_function(a, x2(i)-x2(j), y2(i)-y2(j));
end
end
Z1_to_Z2 = zeros(N1,N2);
for i = 1:N1
for j = 1:N2
Z1_to_Z2(i,j) = coef * K1_function(a, x1(i)-x2(j), y1(i)-y2(j));
end
end
Z2_to_Z1 = zeros(N2,N1);
for i = 1:N2
for j = 1:N1
Z2_to_Z1(i,j) = coef * K1_function(a, x2(i)-x1(j), y2(i)-y1(j));
end
end
% Make Z matrix
Z = [Z1_self, Z1_to_Z2; Z2_to_Z1, Z2_self];
Z = Z - Z(1,:);
Z(1,:) = delta;
% Make V matrix
V = [zeros(N1,1); (v2 - v1)*ones(N2,1)];
rho = Z \ V;
rho1 = rho(1:N1);
rho2 = rho(N1+1:end);
Q_prime = sum(rho1)*delta;
C_prime = Q_prime/(v1 - v2);
% Plot charge distribution
fig_dist = figure; hold on; grid on;
plot(x1, rho1, 'LineWidth', 1.2);
```

```
plot(x2, rho2, 'LineWidth', 1.2);
xlabel('x position (m)'); ylabel('\rho_s (C/m^2)');
legend('Conductor (y=h)','Ground (y=0)','Location','best');
title(sprintf('Finite Ground Charge Distribution w_1=%.1f, w_2=%.1f, h=%.1f', w1, w2,
h));
% Create multiple sets of w1,w2,h, and N
w1_list = [0.5, 1.0];
w2_list = [2.0, 10.0];
h_list = [0.5, 1.0, 2.0];
N1_list = [20, 50, 75, 100, 150, 200, 500];
% Figure 1: % Error vs N1
fig_err = figure; hold on; grid on;
xlabel('N_1 (Segments on conductor)');
ylabel('Relative %Error vs Empirical');
title('Relative %Error (Empirical) vs N_1');
% Figure 2: Convergence vs N1
fig_conv = figure; hold on; grid on;
xlabel('N_1 (Segments on conductor)');
ylabel('Relative Error to C''_{ref} (%)');
title('Convergence to C''_{ref} vs N_1 (log scale)');
% Results table to help with reporting
max_rows = length(w1_list)*length(w2_list)*length(h_list)*length(N1_list);
results = zeros(max_rows, 9);
row = 1;
for ia = 1:length(w1_list)
for ib = 1:length(w2_list)
for ic = 1:length(h_list)
w1 = w1_list(ia);
w2_target = w2_list(ib);
h = h_list(ic);
% Empirical C' calculations
ratio = w1 / h;
if ratio < 1, p = 0.04*(1 - ratio)^2; else, p = 0; end
eps_eff = (er + 1)/2 + (er - 1)/2 * ((1 + 12/ratio)^(-0.5) + p);
if ratio <= 1
Z0_emp = (eta0/(2*pi*sqrt(eps_eff))) * log(8/ratio + ratio/4);
else
Z0_emp = (eta0/sqrt(eps_eff)) / (ratio + 1.393 + 0.667*log(ratio + 1.444));
end
C_emp = 1/(Z0_emp * c0);
% Arrays for this geometry
Percent_error_empirical = zeros(size(N1_list));
Cap_point_matching_list = zeros(size(N1_list));
% Sweep N1
for k = 1:length(N1_list)
N1 = N1_list(k);
delta = w1 / N1;
```

```
a = delta/2;
N2 = max(1, round(w2_target / delta));
w2_current = N2 * delta;
% Centers with new geometry
x1 = linspace(-w1/2 + a, +w1/2 - a, N1).'; yt = h*ones(N1,1);
x2 = linspace(-w2_current/2 + a, +w2_current/2 - a, N2).'; yg = zeros(N2,1);
Z1_self = zeros(N1,N1);
for i = 1:N1
for j = 1:N1
Z1_self(i,j) = coef * K1_function(a, x1(i)-x1(j), yt(i)-yt(j));
end
end
Z2_self = zeros(N2,N2);
for i = 1:N2
for j = 1:N2
Z2_self(i,j) = coef * K1_function(a, x2(i)-x2(j), yg(i)-yg(j));
end
end
Z1_to_Z2 = zeros(N1,N2);
for i = 1:N1
for j = 1:N2
Z1_to_Z2(i,j) = coef * K1_function(a, x1(i)-x2(j), yt(i)-yg(j));
end
end
Z2_to_Z1 = zeros(N2,N1);
for i = 1:N2
for j = 1:N1
Z2_to_Z1(i,j) = coef * K1_function(a, x2(i)-x1(j), yg(i)-yt(j));
end
end
% Make new Z matrix
Z = [Z1_self, Z1_to_Z2; Z2_to_Z1, Z2_self];
Z = Z - Z(1,:);
Z(1,:) = delta;
% Re-impliment V matrix
V = [zeros(N1,1); (v2 - v1)*ones(N2,1)];
rho = Z \ V;
% C' and empirical % error
Q_prime = sum(rho(1:N1)) * delta;
C_prime = Q_prime / (v1 - v2);
Cap_point_matching_list(k) = C_prime;
Percent_error_empirical(k) = 100 * abs(C_prime - C_emp) / C_emp;
% Save a row for reporting
results(row,:) = [w1, w2_current, h, N1, N2, N1+N2, C_prime, C_emp,
Percent_error_empirical(k)];
row = row + 1;
end
```

```
% Graph 1: %error vs N1
figure(fig_err.Number);
plot(N1_list, Percent_error_empirical, '-o', 'LineWidth', 1.2, ...
'DisplayName', sprintf('w1=%.2g, w2=%.2g, h=%.2g', w1, w2_target, h));
% Graph 2: Convergence vs N1
C_ref = Cap_point_matching_list(end);
ErrToRefPct = 100 * abs(Cap_point_matching_list - C_ref) / max(eps, abs(C_ref));
figure(fig_conv.Number);
semilogy(N1_list, ErrToRefPct, '-o', 'LineWidth', 1.2, 'DisplayName',
sprintf('w1=%.2g, w2=%.2g, h=%.2g', w1, w2_target, h));
end
end
end
% Finish figures
figure(fig_err.Number);
yline(1,'--','1% target','LabelHorizontalAlignment','left');
legend('show','Location','northeastoutside');
figure(fig_conv.Number);
legend('show','Location','northeastoutside');
% Table
results = results(1:row-1, :);
T = table(results(:,1), results(:,2), results(:,3), results(:,4), results(:,5),
results(:,6), results(:,7), results(:,8), results(:,9),'VariableNames',
{'w1_m','w2_used_m','h_m','N1','N2','N_total',
'Cprime_MoM_Fperm','Cprime_Emp_Fperm','Error_percent'});
disp(' ');
disp('All runs (each row = one geometry at one N1):');
disp(T);
```

II. MoM Analysis of an Air-Filled Microstrip Transmission Line with a Finite Ground Plane – Using Galerkin Testing Method

1.Write the theory and equations, and describe your MoM algorithm.

Theory:

The Galerkin method is like point matching except that it integrates instead of using summations. Similarly, the transmission line is broken into segments. However, this time, instead of choosing a center point and analyzing it from that singular point within the segment, the entire segment is integrated over. This provides a more accurate representation of the transmission line's properties.

Equations:

$$E_2(a,\ x1\ ,\ y1,\ x2\ ,\ y2) = \Re e\left\{ l * \frac{(z-z')^2}{2}\left[\ln(z-z') - \frac{3}{2}\right]\right\}$$

evaluated for z' = -a to a and z = z_1 to z_2

$$l = \frac{((x_2 - x_1) + j(y_2 - y_1))}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

$$d = 2h$$

$$\Delta = \frac{w}{N}$$

$$a = \frac{\Delta}{2}$$

$$[\rho_{sj}] = [v_i][Z_{ji}]^{-1}$$

$$Q' = \sum_{j=1}^{N} \rho_{sj}\Delta l_j$$

$$C' = \frac{Q'}{v_1 - v_2}$$

$$Z_{ij} = \frac{1}{2\pi\varepsilon_0} K_2\left(\frac{\Delta}{2},\ x_i - x_j\ -\ \frac{\Delta}{2},\ y_i\ -\ y_j\ ,\ x_i\ -\ x_j\ +\frac{\Delta}{2},\ y_i\ -\ y_j\right)$$

Empirical Formulas:

$$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[\left(1 + 12\frac{h}{w}\right)^{-1/2} + p\right]$$

$$Z_0 = \frac{\eta_0}{2\pi\sqrt{\varepsilon_{reff}}} \ln\left(\frac{8h}{w} + \frac{w}{4h}\right) \quad \text{for} \quad \frac{w}{h} \le 1,$$

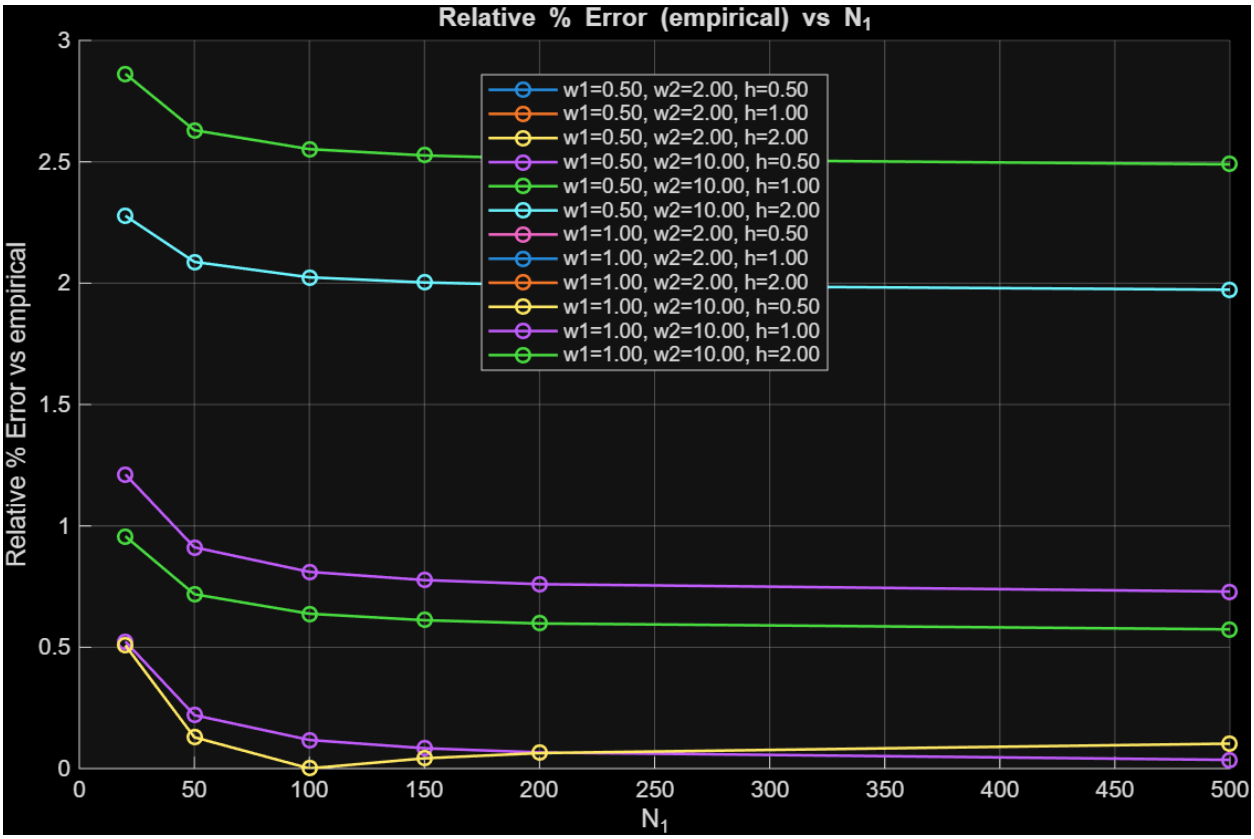
$$Z_0 = \frac{\eta_0}{\sqrt{\varepsilon_{reff}}}\left[\frac{w}{h} + 1.393 + 0.667 \ln\left(\frac{w}{h} + 1.444\right)\right]^{-1} \quad \text{for} \quad \frac{w}{h} > 1,$$

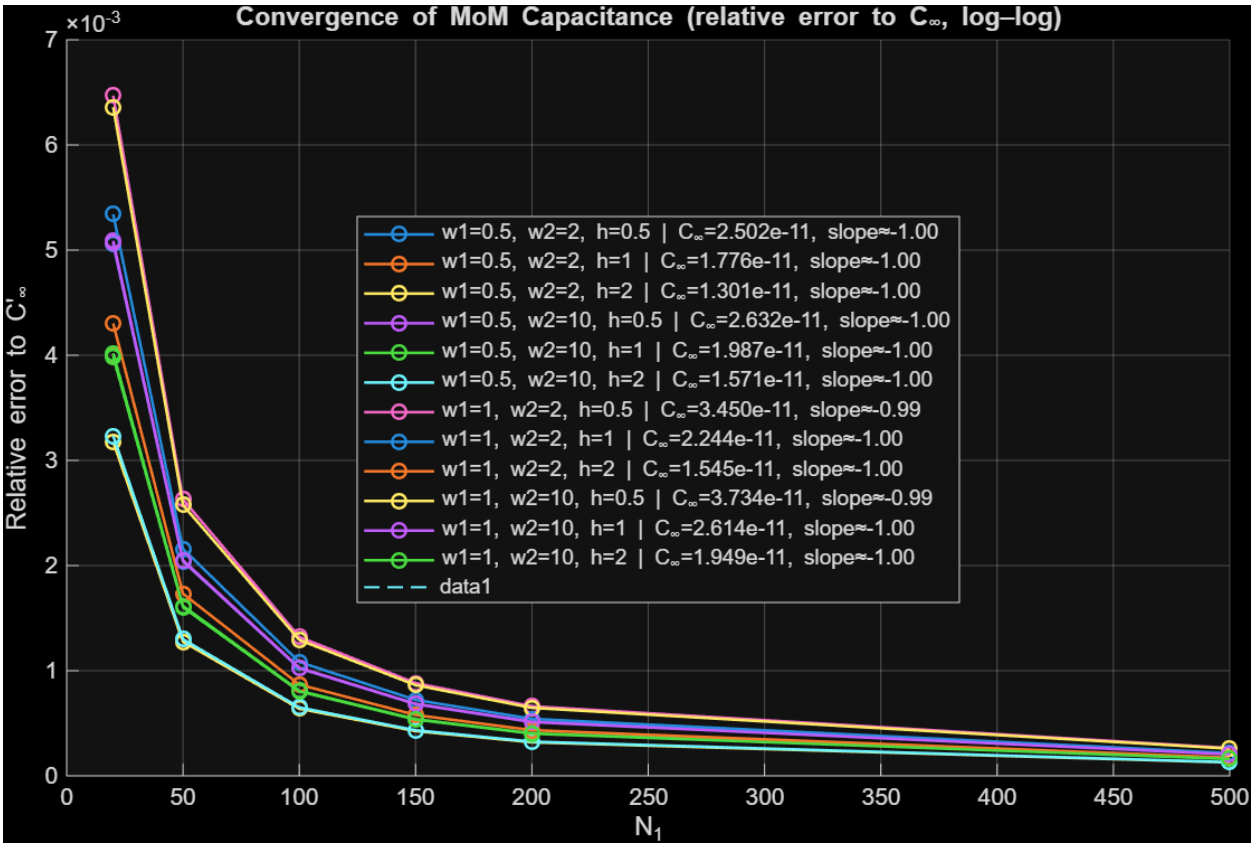
Galerkin (Finite Ground):

In the Galerkin method, both the top conductor (with width w1) and the finite ground plane (with width w2) are divided into small, equal-width pulse segments. Each segment carries an unknown uniform charge, and the testing is done using the same pulse functions, which is why this approach is called “pulse–pulse” Galerkin. The method builds a symmetric block matrix that includes self and mutual impedance terms, calculated using the K2_function, which integrates the potential interactions between all segments on the two conductors. The top conductor is set to one volt, and the ground plane is set to zero volts. Solving the complete matrix equation provides the charge density on both conductors. The total charge on the top conductor divided by the voltage difference gives the capacitance per unit length. This calculation is then repeated for larger numbers of segments on the top conductor to see how the capacitance value converges toward a stable result that represents the theoretical limit of the method.

2.

For several characteristic values of w1, w2, and h, find the capacitance per unit length of the transmission line, C’, and tabulate (in a table) and plot (as a graph) the relative percentage error when compared to the result obtained by empirical formulas (p.569, Electromagnetics, Notaros – attached here) as a function of N. Discuss the convergence of MoM results with increasing N.


Relative % Error (empirical) vs N1
Relative % Error vs empirical
N1
w1=0.50, w2=2.00, h=0.50
w1=0.50, w2=2.00, h=1.00
w1=0.50, w2=2.00, h=2.00
w1=0.50, w2=10.00, h=0.50
w1=0.50, w2=10.00, h=1.00
w1=0.50, w2=10.00, h=2.00
w1=1.00, w2=2.00, h=0.50
w1=1.00, w2=2.00, h=1.00
w1=1.00, w2=2.00, h=2.00
w1=1.00, w2=10.00, h=0.50
w1=1.00, w2=10.00, h=1.00
w1=1.00, w2=10.00, h=2.00


Convergence of MoM Capacitance (relative error to C∞, log–log)
×10-3
Relative error to C'∞
N1
w1=0.5, w2=2, h=0.5 | C∞=2.502e-11, slope≈-1.00
w1=0.5, w2=2, h=1 | C∞=1.776e-11, slope≈-1.00
w1=0.5, w2=2, h=2 | C∞=1.301e-11, slope≈-1.00
w1=0.5, w2=10, h=0.5 | C∞=2.632e-11, slope≈-1.00
w1=0.5, w2=10, h=1 | C∞=1.987e-11, slope≈-1.00
w1=0.5, w2=10, h=2 | C∞=1.571e-11, slope≈-1.00
w1=1, w2=2, h=0.5 | C∞=3.450e-11, slope≈-0.99
w1=1, w2=2, h=1 | C∞=2.244e-11, slope≈-1.00
w1=1, w2=2, h=2 | C∞=1.545e-11, slope≈-1.00
w1=1, w2=10, h=0.5 | C∞=3.734e-11, slope≈-0.99
w1=1, w2=10, h=1 | C∞=2.614e-11, slope≈-1.00
w1=1, w2=10, h=2 | C∞=1.949e-11, slope≈-1.00
data1

As we increase the number of segments N_1, the MoM results clearly get more accurate. You can see that the relative error drops fast at first and then levels off once N_1 gets large, meaning the solution has basically converged. All the cases show about the same trend — roughly a straight line with slope −1 on the log–log plot, which lines up with first-order convergence. After around 300 segments, the changes in capacitance are super small, so using more points doesn't really make a difference. Overall, this shows that the MoM approach gives reliable and stable results once the mesh is fine enough.

w1	w2_used	h	N1	N2	Ntot	C'_MoM(F/m)	C'_emp(F/m)	Err(%)
0.50	2.00	0.50	20	80	100	2.488909e-11	2.632637e-11	5.459
0.50	2.00	0.50	50	200	250	2.496465e-11	2.632637e-11	5.172
0.50	2.00	0.50	100	400	500	2.499010e-11	2.632637e-11	5.076
0.50	2.00	0.50	150	600	750	2.499861e-11	2.632637e-11	5.043
0.50	2.00	0.50	200	800	1000	2.500288e-11	2.632637e-11	5.027
0.50	2.00	0.50	500	2000	2500	2.501056e-11	2.632637e-11	4.998
0.50	2.00	1.00	20	80	100	1.768754e-11	1.998088e-11	11.478
0.50	2.00	1.00	50	200	250	1.773004e-11	1.998088e-11	11.265
0.50	2.00	1.00	100	400	500	1.774432e-11	1.998088e-11	11.193
0.50	2.00	1.00	150	600	750	1.774910e-11	1.998088e-11	11.170
0.50	2.00	1.00	200	800	1000	1.775148e-11	1.998088e-11	11.158
0.50	2.00	1.00	500	2000	2500	1.775579e-11	1.998088e-11	11.136
0.50	2.00	2.00	20	80	100	1.296868e-11	1.602055e-11	19.050
0.50	2.00	2.00	50	200	250	1.299339e-11	1.602055e-11	18.895
0.50	2.00	2.00	100	400	500	1.300168e-11	1.602055e-11	18.844
0.50	2.00	2.00	150	600	750	1.300445e-11	1.602055e-11	18.826
0.50	2.00	2.00	200	800	1000	1.300584e-11	1.602055e-11	18.818
0.50	2.00	2.00	500	2000	2500	1.300833e-11	1.602055e-11	18.802
0.50	10.00	0.50	20	400	420	2.618841e-11	2.632637e-11	0.524
0.50	10.00	0.50	50	1000	1050	2.626839e-11	2.632637e-11	0.220
0.50	10.00	0.50	100	2000	2100	2.629531e-11	2.632637e-11	0.118
0.50	10.00	0.50	150	3000	3150	2.630431e-11	2.632637e-11	0.084
0.50	10.00	0.50	200	4000	4200	2.630882e-11	2.632637e-11	0.067
0.50	10.00	0.50	500	10000	10500	2.631694e-11	2.632637e-11	0.036
0.50	10.00	1.00	20	400	420	1.978960e-11	1.998088e-11	0.957
0.50	10.00	1.00	50	1000	1050	1.983735e-11	1.998088e-11	0.718
0.50	10.00	1.00	100	2000	2100	1.985339e-11	1.998088e-11	0.638
0.50	10.00	1.00	150	3000	3150	1.985874e-11	1.998088e-11	0.611
0.50	10.00	1.00	200	4000	4200	1.986142e-11	1.998088e-11	0.598
0.50	10.00	1.00	500	10000	10500	1.986625e-11	1.998088e-11	0.574

0.50	10.00	2.00	20	400	420	1.565572e-11	1.602055e-11	2.277
0.50	10.00	2.00	50	1000	1050	1.568611e-11	1.602055e-11	2.088
0.50	10.00	2.00	100	2000	2100	1.569630e-11	1.602055e-11	2.024
0.50	10.00	2.00	150	3000	3150	1.569970e-11	1.602055e-11	2.003
0.50	10.00	2.00	200	4000	4200	1.570140e-11	1.602055e-11	1.992
0.50	10.00	2.00	500	10000	10500	1.570447e-11	1.602055e-11	1.973
1.00	2.00	0.50	20	40	60	3.427597e-11	3.729297e-11	8.090
1.00	2.00	0.50	50	100	150	3.440847e-11	3.729297e-11	7.735
1.00	2.00	0.50	100	200	300	3.445359e-11	3.729297e-11	7.614
1.00	2.00	0.50	150	300	450	3.446874e-11	3.729297e-11	7.573
1.00	2.00	0.50	200	400	600	3.447633e-11	3.729297e-11	7.553
1.00	2.00	0.50	500	1000	1500	3.449003e-11	3.729297e-11	7.516
1.00	2.00	1.00	20	40	60	2.232452e-11	2.632637e-11	15.201
1.00	2.00	1.00	50	100	150	2.239594e-11	2.632637e-11	14.930
1.00	2.00	1.00	100	200	300	2.242008e-11	2.632637e-11	14.838
1.00	2.00	1.00	150	300	450	2.242817e-11	2.632637e-11	14.807
1.00	2.00	1.00	200	400	600	2.243222e-11	2.632637e-11	14.792
1.00	2.00	1.00	500	1000	1500	2.243952e-11	2.632637e-11	14.764
1.00	2.00	2.00	20	40	60	1.538806e-11	1.998088e-11	22.986
1.00	2.00	2.00	50	100	150	1.542778e-11	1.998088e-11	22.787
1.00	2.00	2.00	100	200	300	1.544114e-11	1.998088e-11	22.720
1.00	2.00	2.00	150	300	450	1.544561e-11	1.998088e-11	22.698
1.00	2.00	2.00	200	400	600	1.544785e-11	1.998088e-11	22.687
1.00	2.00	2.00	500	1000	1500	1.545188e-11	1.998088e-11	22.667
1.00	10.00	0.50	20	200	220	3.710352e-11	3.729297e-11	0.508
1.00	10.00	0.50	50	500	550	3.724482e-11	3.729297e-11	0.129
1.00	10.00	0.50	100	1000	1100	3.729271e-11	3.729297e-11	0.001
1.00	10.00	0.50	150	1500	1650	3.730877e-11	3.729297e-11	0.042
1.00	10.00	0.50	200	2000	2200	3.731681e-11	3.729297e-11	0.064
1.00	10.00	0.50	500	5000	5500	3.733131e-11	3.729297e-11	0.103

1.00	10.00	1.00	20	200	220	2.600723e-11	2.632637e-11	1.212
1.00	10.00	1.00	50	500	550	2.608636e-11	2.632637e-11	0.912
1.00	10.00	1.00	100	1000	1100	2.611299e-11	2.632637e-11	0.810
1.00	10.00	1.00	150	1500	1650	2.612190e-11	2.632637e-11	0.777
1.00	10.00	1.00	200	2000	2200	2.612636e-11	2.632637e-11	0.760
1.00	10.00	1.00	500	5000	5500	2.613440e-11	2.632637e-11	0.729
1.00	10.00	2.00	20	200	220	1.940891e-11	1.998088e-11	2.863
1.00	10.00	2.00	50	500	550	1.945536e-11	1.998088e-11	2.630
1.00	10.00	2.00	100	1000	1100	1.947095e-11	1.998088e-11	2.552
1.00	10.00	2.00	150	1500	1650	1.947617e-11	1.998088e-11	2.526
1.00	10.00	2.00	200	2000	2200	1.947877e-11	1.998088e-11	2.513
1.00	10.00	2.00	500	5000	5500	1.948347e-11	1.998088e-11	2.489

```
clc; clear;
epsilon0 = 8.854187817e-12;
epsilon_r = 1;
epsilon = epsilon0*epsilon_r;
```

```
w1 = 1;
w2 = 10;
h = 1;
v1 = 1;
v2 = 0;
N1 = 100;
eta0 = 377;
c0 = 3e8;
delta = w1 / N1;
a = delta/2;
N2 = max(1, round(w2 / delta));
w2 = N2 * delta;
N = N1 + N2;
coef = 1/(2*pi*epsilon);
% set conductor centered above ground
x1 = linspace(-w1/2 + a, +w1/2 - a, N1).'; y1 = h*ones(N1,1);
x2 = linspace(-w2/2 + a, +w2/2 - a, N2).'; y2 = zeros(N2,1);
Z1_self = zeros(N1,N1);
for i = 1:N1
for j = 1:N1
dx = x1(i) - x1(j);
x_right = dx + a;
x_left = dx - a;
Z1_self(i,j) = coef * K2_function(a, x_right, 0, x_left, 0);
end
end
Z2_self = zeros(N2,N2);
for i = 1:N2
for j = 1:N2
dx = x2(i) - x2(j);
x_right = dx + a;
x_left = dx - a;
Z2_self(i,j) = coef * K2_function(a, x_right, 0, x_left, 0);
end
end
Z1_to_Z2 = zeros(N1,N2);
for i = 1:N1
for j = 1:N2
dx = x1(i) - x2(j);
x_right = dx + a;
x_left = dx - a;
Z1_to_Z2(i,j) = coef * K2_function(a, x_right, h, x_left, h);
end
end
% symmetry
Z2_to_Z1 = Z1_to_Z2.';
% Make Z matrix
```

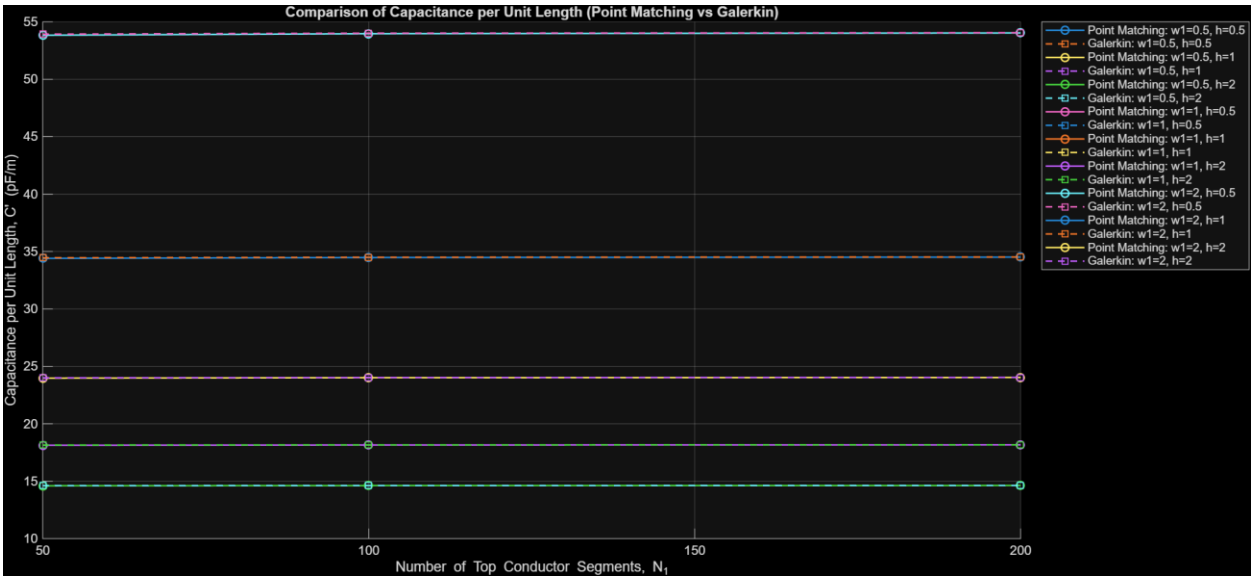
```
Z = [Z1_self, Z1_to_Z2; Z2_to_Z1, Z2_self];
Z = Z - Z(1,:);
Z(1,:) = delta;
% Make V matrix
V = delta * [zeros(N1,1) ; (v1-v2)*ones(N2,1) ];
% Surface charge density
rho = Z \ V;
rho1 = rho(1:N1);
rho2 = rho(N1+1:end);
Q_prime = sum(rho1) * delta;
C_prime = Q_prime / (v1 - v2);
% --- Diagnostics ---
% sym_err = norm(Z - Z.', 'fro') / max(1,norm(Z,'fro'));
% fprintf("Symmetry rel. error: %.3e\n", sym_err);
% fprintf("N1 = %d, N2 = %d (N = %d)\n", N1, N2, N);
% fprintf("C' = %.6e F/m\n", C_prime);
% plotting rho
figure;
plot(x1, rho1, 'LineWidth', 1.2); hold on;
plot(x2, rho2, 'LineWidth', 1.2); grid on;
xlabel('x position (m)'); ylabel('\rho_s (C/m^2)');
legend('Conductor (y = h)', 'Ground (y = 0)', 'Location','best');
title(sprintf('Galerkin (Pulse–Pulse) — w_1=%.3g, w_2=%.3g, h=%.3g, N_1=%d,
N_2=%d', ...
w1, w2, h, N1, N2));
% several characteristic values of w1, w2, and h
w1_list = [0.5, 1.0];
w2_list = [2.0, 10.0];
h_list = [0.5, 1.0, 2.0];
% Needs to be a function of N
N_list = [20, 50, 100, 150, 200, 500];
% results: [w1, w2_used, h, N1, N2, Ntot, Cprime_MoM, Cprime_emp, Err%]
max_rows = length(w1_list)*length(w2_list)*length(h_list)*length(N_list);
results = zeros(max_rows,9);
row = 1;
for i_a = 1:length(w1_list)
for i_b = 1:length(w2_list)
for i_c = 1:length(h_list)
w1_cur = w1_list(i_a);
w2_cur = w2_list(i_b);
h_cur = h_list(i_c);
ratio = w1_cur / h_cur;
if ratio < 1
p = 0.04*(1 - ratio)^2;
else
p = 0;
end
```

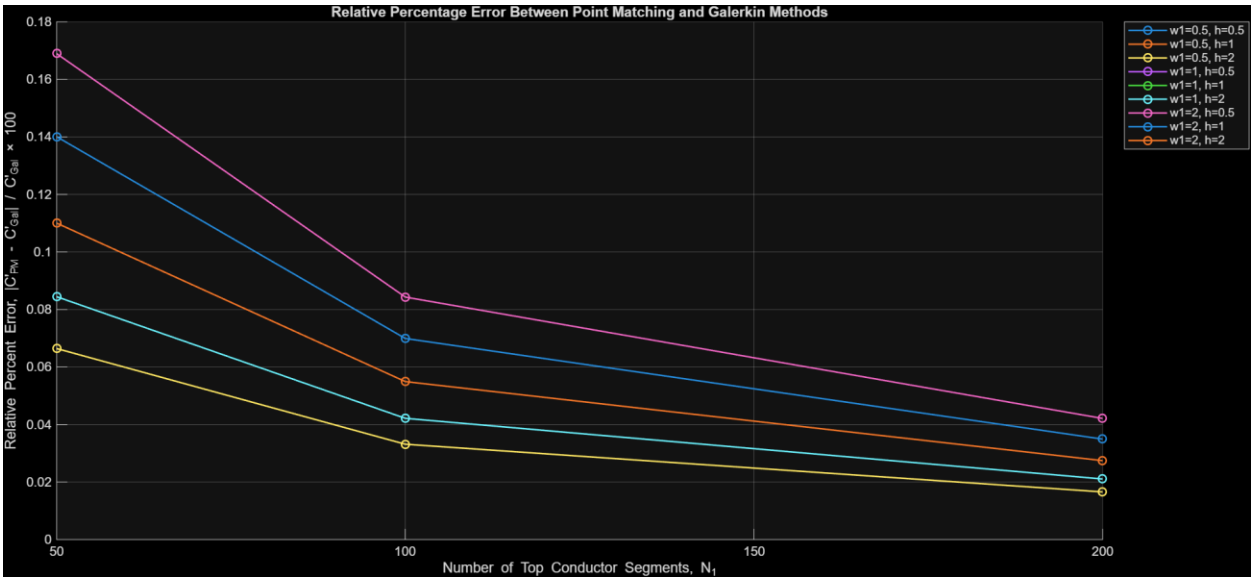
```
eps_eff = (epsilon_r + 1)/2 + ((epsilon_r - 1)/2)*((1 + 12/ratio)^(-1/2) + p);
if ratio <= 1
Z0_emp = eta0/(2*pi*sqrt(eps_eff)) * log(8/ratio + ratio/4);
else
Z0_emp = eta0/sqrt(eps_eff) * (ratio + 1.393 + 0.667 * log(ratio + 1.444))^-1;
end
Cprime_emp = 1/(Z0_emp*c0);
for i_n = 1:length(N_list)
N1_cur = N_list(i_n);
delta = w1_cur / N1_cur;
a = delta/2;
N2_cur = max(1, round(w2_cur / delta));
w2_used = N2_cur * delta;
% Re-establish X1 and and X2 coordinates with new variables
x1 = linspace(-w1_cur/2 + a, +w1_cur/2 - a, N1_cur).';
x2 = linspace(-w2_used/2 + a, +w2_used/2 - a, N2_cur).';
Z1_self = zeros(N1_cur,N1_cur);
for i = 1:N1_cur
for j = 1:N1_cur
dx = x1(i) - x1(j);
x_right = dx + a;
x_left = dx - a;
Z1_self(i,j) = coef * K2_function(a, x_right, 0, x_left, 0);
end
end
Z2_self = zeros(N2_cur,N2_cur);
for i = 1:N2_cur
for j = 1:N2_cur
dx = x2(i) - x2(j);
x_right = dx + a;
x_left = dx - a;
Z2_self(i,j) = coef * K2_function(a, x_right, 0, x_left, 0);
end
end
Z1_to_Z2 = zeros(N1_cur,N2_cur);
for i = 1:N1_cur
for j = 1:N2_cur
dx = x1(i) - x2(j);
x_right = dx + a;
x_left = dx - a;
Z1_to_Z2(i,j) = coef * K2_function(a, x_right, h_cur, x_left, h_cur);
end
end
% symmetry
Z2_to_Z1 = Z1_to_Z2.';
% Make z matrix
Z = [Z1_self, Z1_to_Z2; Z2_to_Z1, Z2_self];
```

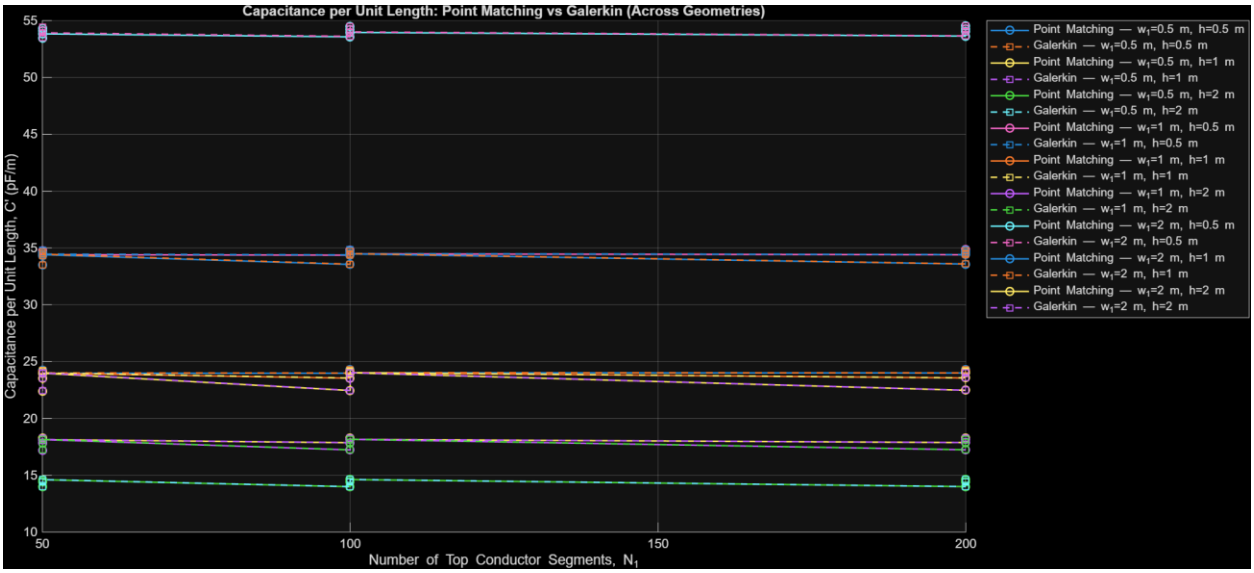
```
Z = Z - Z(1,:);
Z(1,:) = delta;
% make V matrix
V = delta * [zeros(N1_cur,1); (v1-v2) * ones(N2_cur,1)];
% solve, top charge -> C'
rho = Z \ V;
rho1 = rho(1:N1_cur);
Q_prime = sum(rho1) * delta;
C_prime_Galerkin = Q_prime/(v1 - v2);
% percent error
percent_error = abs(C_prime_Galerkin - Cprime_emp)/Cprime_emp*100;
% store row
results(row,1) = w1_cur;
results(row,2) = w2_used;
results(row,3) = h_cur;
results(row,4) = N1_cur;
results(row,5) = N2_cur;
results(row,6) = N1_cur + N2_cur;
results(row,7) = C_prime_Galerkin;
results(row,8) = Cprime_emp;
results(row,9) = percent_error;
row = row + 1;
end
end
end
end
results = results(1:row-1,:);
fprintf('\nCapacitance and Relative Error (MoM vs Empirical)\n');
fprintf(' w1 w2_used h N1 N2 Ntot C''_MoM(F/m) C''_emp(F/m) Err(%%)\n');
for r = 1:size(results,1)
fprintf('%5.2f %7.2f %5.2f %5d %5d %5d %14.6e %14.6e %7.3f\n', ...
results(r,1), results(r,2), results(r,3), ...
results(r,4), results(r,5), results(r,6), ...
results(r,7), results(r,8), results(r,9));
end
% Plot: empirical % error vs N1
figure; hold on; grid on;
labels = {};
idx = 1;
for i_a = 1:length(w1_list)
for i_b = 1:length(w2_list)
for i_c = 1:length(h_list)
Ns = zeros(length(N_list),1);
Es = zeros(length(N_list),1);
for k = 1:length(N_list)
Nwant = N_list(k);
for r = 1:size(results,1)
```

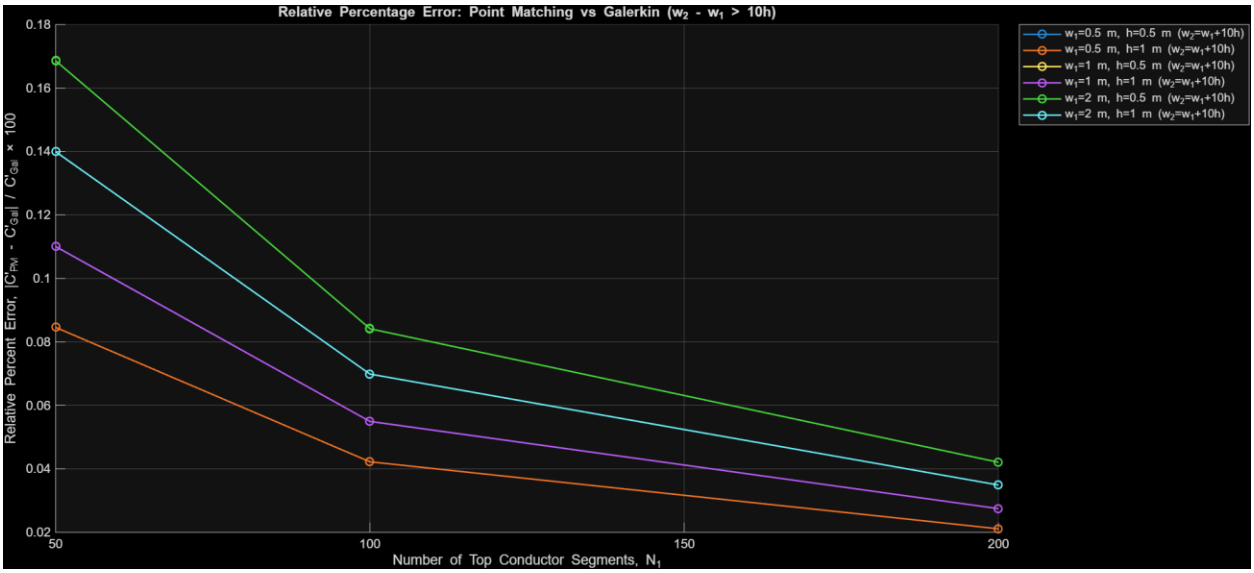
```
if results(r,1)==w1_list(i_a) && results(r,3)==h_list(i_c) && results(r,4)==Nwant
Ns(k) = results(r,4);
Es(k) = results(r,9);
w2_used_for_label = results(r,2);
end
end
end
%Outside of K for loop
plot(Ns, Es, '-o', 'LineWidth', 1.2);
labels{idx} = sprintf('w1=%.2f, w2=%.2f, h=%.2f', w1_list(i_a), w2_list(i_b),
h_list(i_c));
idx = idx + 1;
end
end
end
xlabel('N_1'); ylabel('Relative % Error vs empirical');
title('Relative % Error (empirical) vs N_1');
legend(labels, 'Location', 'best'); hold off;
% Convergence graph: relative error to C_inf
% For each (w1, w2_current, h) triplet, estimate C_inf with 1/N extrapolation
% using the two largest N1 runs: C(N) = C_inf + A/N => C_inf ≈ (N2*C2 - N1*C1)/(N2-
N1).
triplets = unique(results(:,[1,2,3]), 'rows', 'stable'); % [w1, w2_used, h]
figure; hold on; grid on;
leg = strings(size(triplets,1),1);
for t = 1:size(triplets,1)
w1c = triplets(t,1); w2c = triplets(t,2); hc = triplets(t,3);
% rows for this triplet
mask = (results(:,1)==w1c) & (results(:,2)==w2c) & (results(:,3)==hc);
sub = results(mask, :);
% sort by N1
[Nvec, idx] = sort(sub(:,4));
Cvec = sub(idx,7); % C'_MoM
if numel(Nvec) < 2
continue; % need at least two points for extrapolation
end
% Two largest N1 → C_inf
N1a = Nvec(end-1); N1b = Nvec(end);
C1a = Cvec(end-1); C1b = Cvec(end);
Cinf = (N1b*C1b - N1a*C1a) / (N1b - N1a);
% Relative error to C_inf
relErr = abs(Cvec - Cinf) ./ abs(Cinf);
% slope on log-log: relErr ~ k * N^slope
p = polyfit(log(Nvec), log(relErr), 1);
slope_est = p(1);
loglog(Nvec, relErr, '-o', 'LineWidth', 1.2);
leg(t) = sprintf('w1=%.2g, w2=%.2g, h=%.2g | C_\\infty=%.3e, slope≈%.2f', ...
```

```
w1c, w2c, hc, Cinf, slope_est);
end
xlabel('N_1');
ylabel('Relative error to C''_\infty');
title('Convergence of MoM Capacitance (relative error to C_\infty, log–log)');
legend(leg(leg~=""), 'Location','southwest'); % drop empties if any
hold off;
% 1/N reference
ax = gca;
xGuide = get(ax, 'XLim');
xg = logspace(log10(xGuide(1)), log10(xGuide(2)), 50);
yGuide = get(ax, 'YLim');
k = 0.5 * yGuide(2);
hold on;
loglog(xg, k./xg, '--', 'LineWidth', 1.0);
text(exp(mean(log(xGuide))), k/exp(mean(log(xGuide))), ' ~1/N',
'VerticalAlignment','bottom');
hold off;
```




Relative Percentage Error Between Point Matching and Galerkin Methods
Relative Percent Error, |C'PM - C'Gal| / C'Gal × 100
Number of Top Conductor Segments, N1
w1=0.5, h=0.5
w1=0.5, h=1
w1=0.5, h=2
w1=1, h=0.5
w1=1, h=1
w1=1, h=2
w1=2, h=0.5
w1=2, h=1
w1=2, h=2

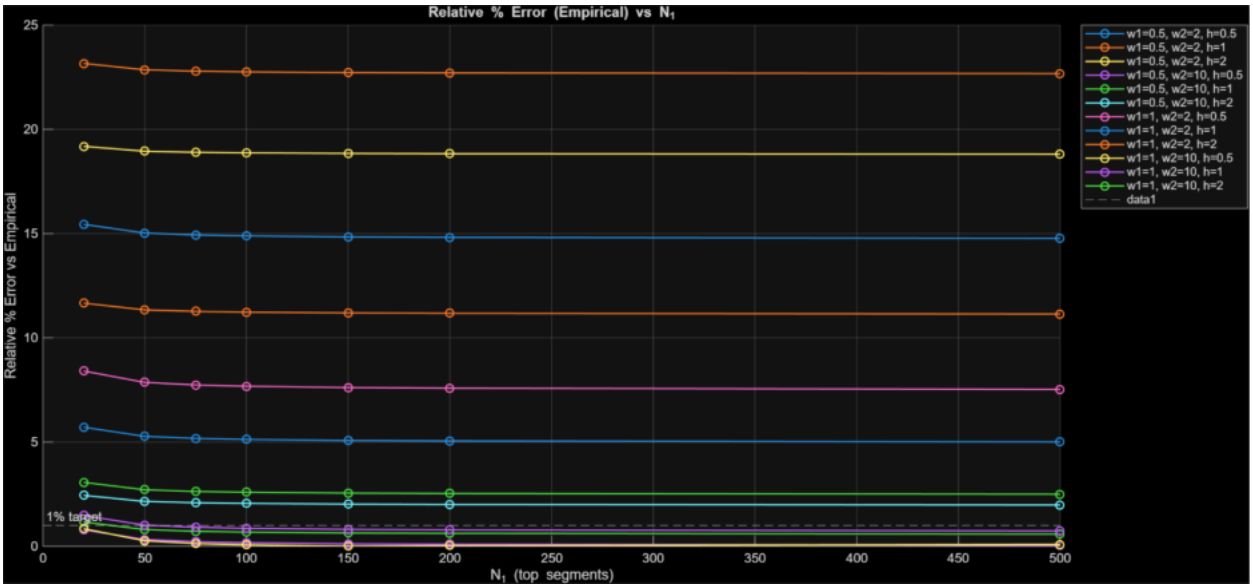

Capacitance per Unit Length: Point Matching vs Galerkin (Across Geometries)
Capacitance per Unit Length, C' (pF/m)
Number of Top Conductor Segments, N1
Point Matching — w1=0.5 m, h=0.5 m
Galerkin — w1=0.5 m, h=0.5 m
Point Matching — w1=0.5 m, h=1 m
Galerkin — w1=0.5 m, h=1 m
Point Matching — w1=0.5 m, h=2 m
Galerkin — w1=0.5 m, h=2 m
Point Matching — w1=1 m, h=0.5 m
Galerkin — w1=1 m, h=0.5 m
Point Matching — w1=1 m, h=1 m
Galerkin — w1=1 m, h=1 m
Point Matching — w1=1 m, h=2 m
Galerkin — w1=1 m, h=2 m
Point Matching — w1=2 m, h=0.5 m
Galerkin — w1=2 m, h=0.5 m
Point Matching — w1=2 m, h=1 m
Galerkin — w1=2 m, h=1 m
Point Matching — w1=2 m, h=2 m
Galerkin — w1=2 m, h=2 m


Relative Percentage Error: Point Matching vs Galerkin (w2 - w1 > 10h)
Relative Percent Error, |C'PM - C'Gal| / C'Gal × 100
Number of Top Conductor Segments, N1
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
50
100
150
200
w1=0.5 m, h=0.5 m (w2=w1+10h)
w1=0.5 m, h=1 m (w2=w1+10h)
w1=1 m, h=0.5 m (w2=w1+10h)
w1=1 m, h=1 m (w2=w1+10h)
w1=2 m, h=0.5 m (w2=w1+10h)
w1=2 m, h=1 m (w2=w1+10h)

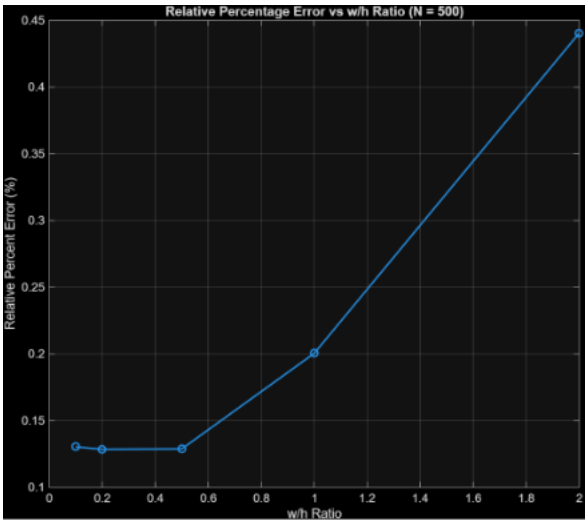
III. Comparison of different testing techniques, Galerkin and point matching, and different treatments of the ground plane, finite and infinite

1. Compare the relative percentage errors obtained by the Galerkin method and the point matching method, respectively (approaches I and II above). Also, compare these results with the corresponding results for the analysis with an infinite ground plane and image theory (Project 1). What values of w1, w2, and h result in good agreement between the analyses with finite and infinite ground planes, respectively?

Point Matching:



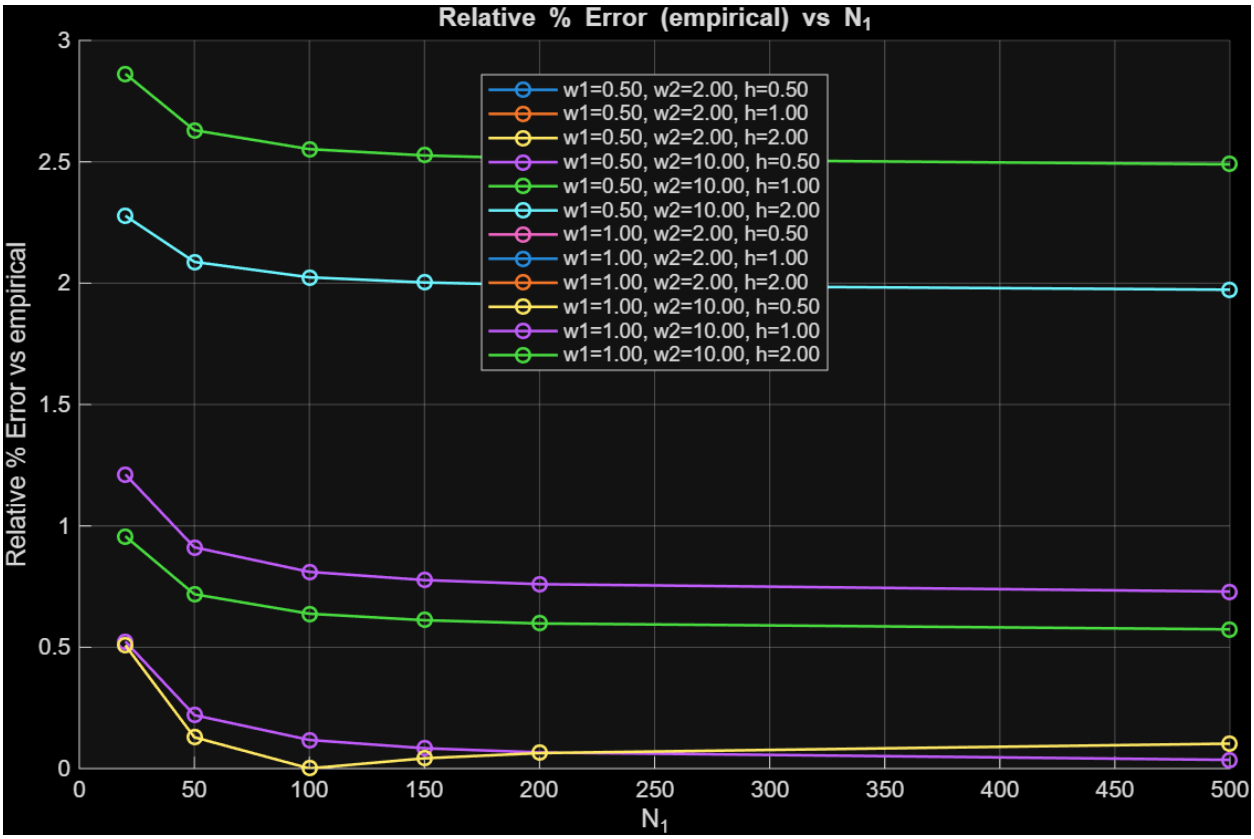
Project1 Point Matching with N=500:



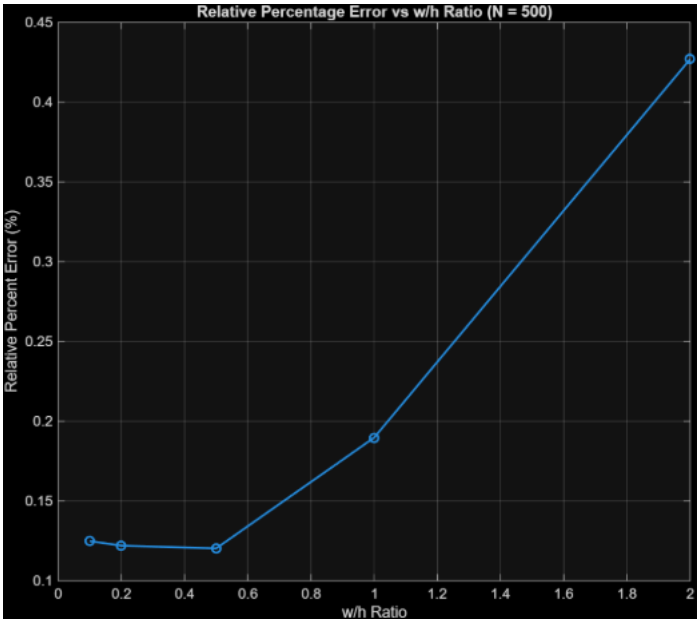

```
0.5          10          0.5     500     10000     10500     2.6314e-11          2.6326e-11          0.046482
```

The values for w1, w2, and h that result in a good agreement between the analyses with the finite and infinite ground planes are 0.5, 10, and 0.5, respectively. This is because the smaller the height and w1 with respect to w2, the more closely the ground resembles an infinite ground plane.

Galerkin:



Project1 Point Matching with N=500:



Just like with point matching, the values for w1, w2, and h that result in a good agreement between the analyses with the finite and infinite ground planes are 0.5, 10, and 0.5,

respectively. Again, this is because the smaller the height and w1 with respect to w2, the more closely the ground resembles an infinite ground plane.

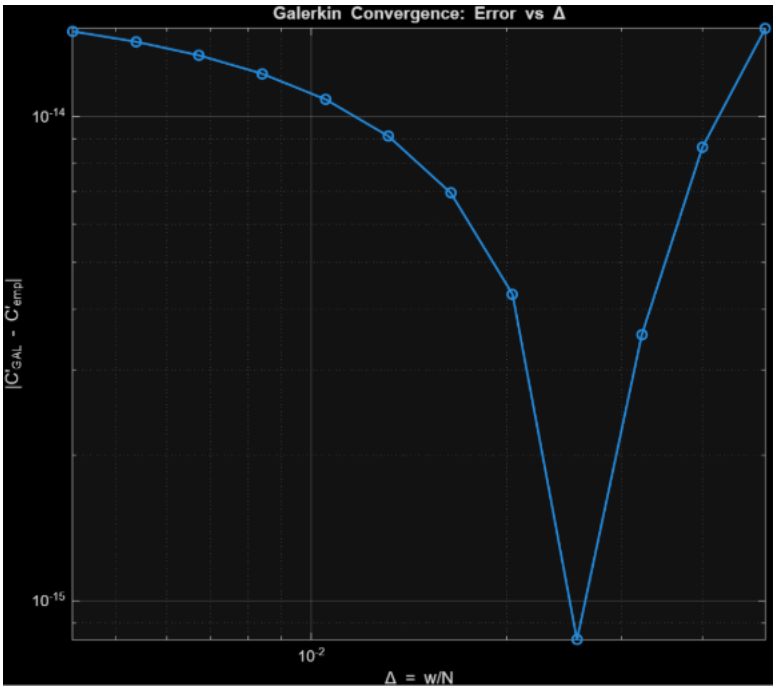
2.

Compare convergence rates obtained by the point matching method and the Galerkin method, respectively (approaches I and II), for at least 10 different values of (discretization segment lengths). Convergence rates should be calculated as follows:

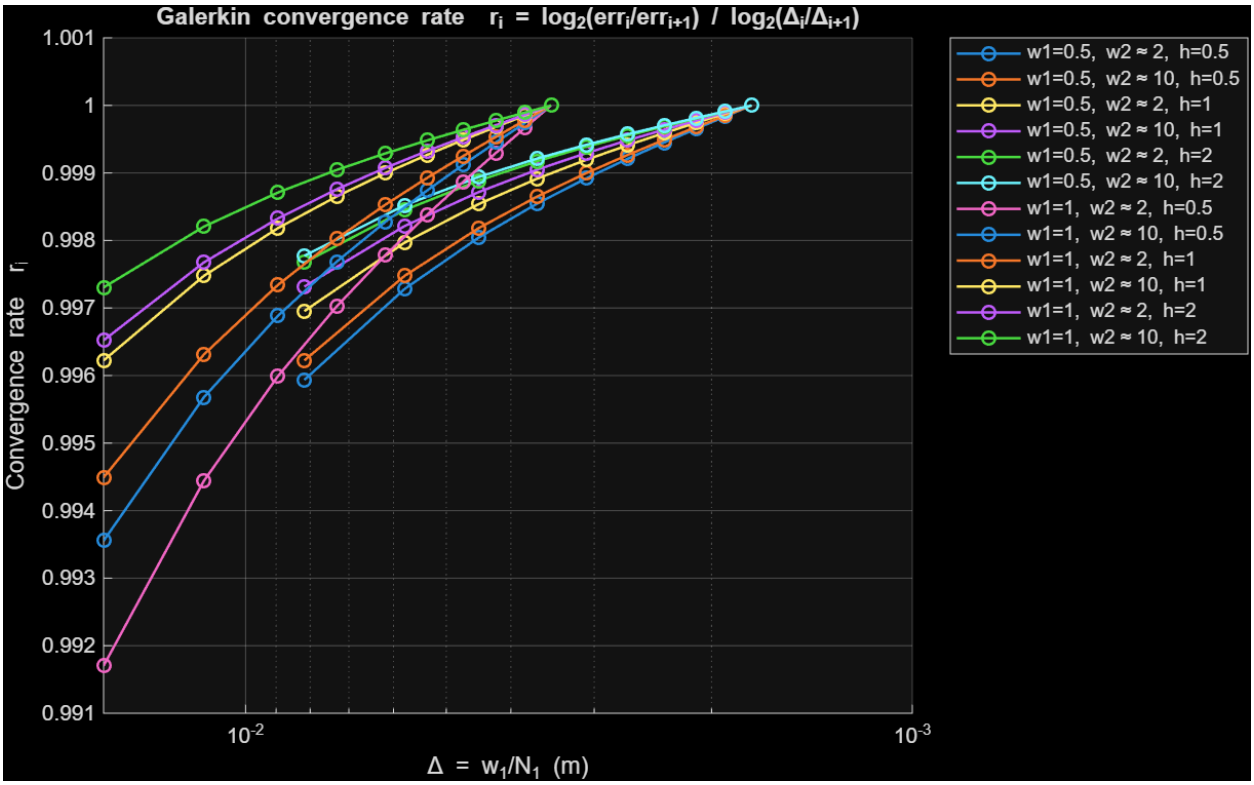
$\frac{\log_2\left(\frac{err_1}{err_{1+1}}\right)}{\log_2\left(\frac{\Delta_i}{\Delta_{i+1}}\right)}$ where $err_1, err_2, \ldots err_i \ldots$ are in descending order

Galerkin method:

with an infinite ground plane:

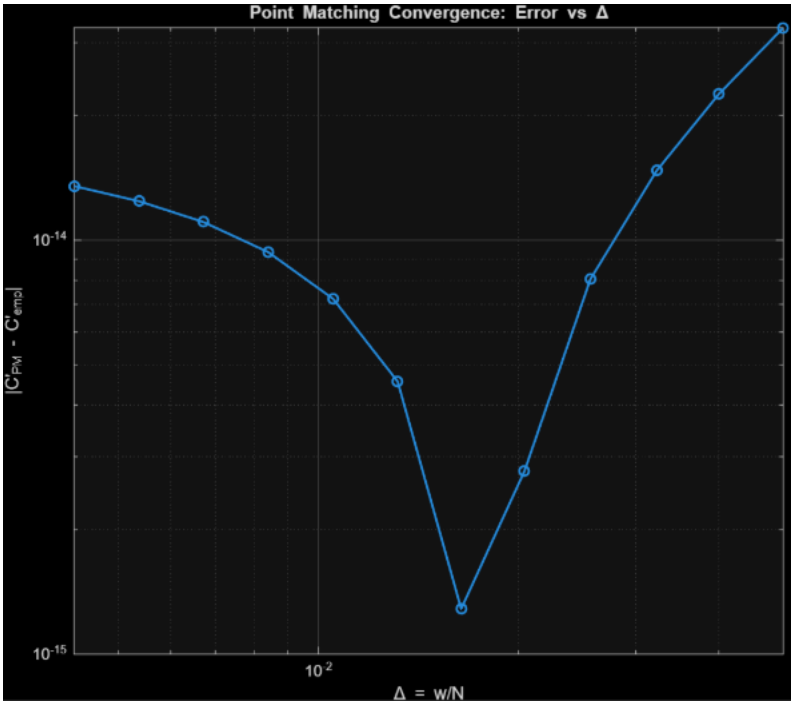


With a finite ground plane:

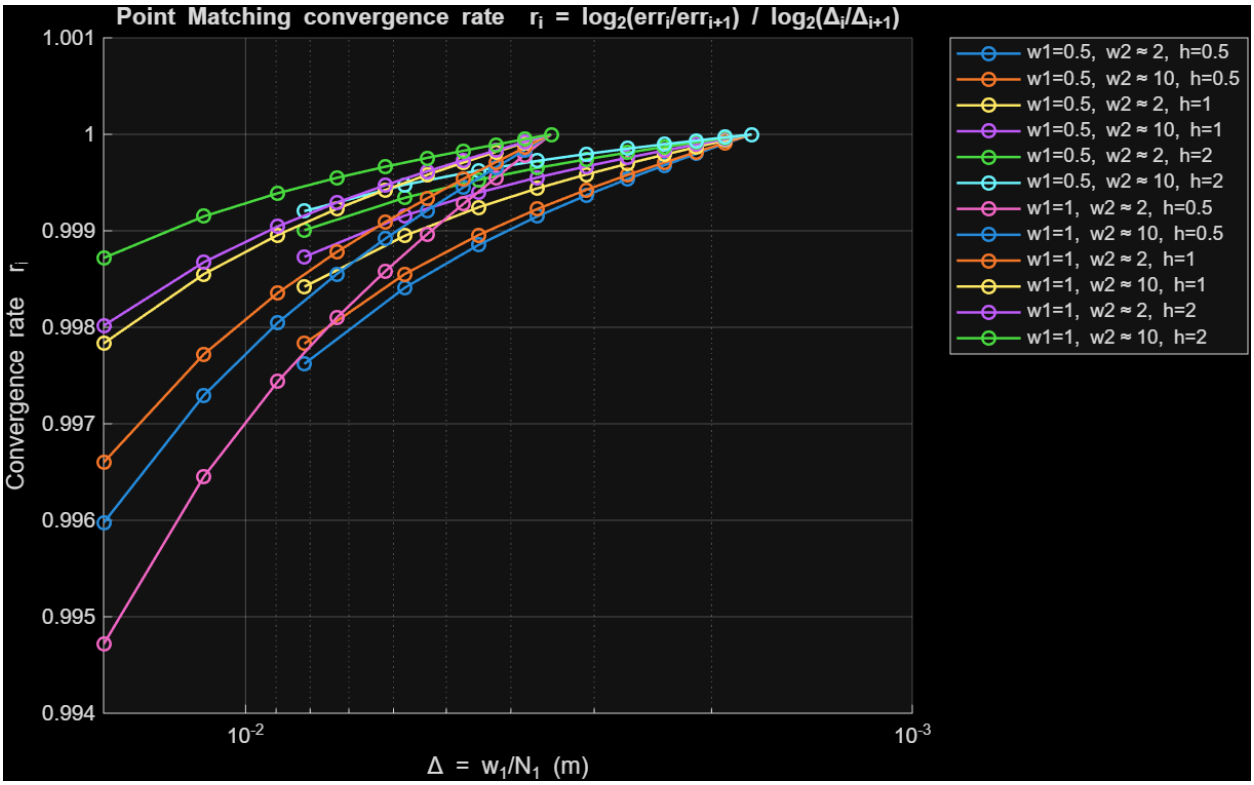


Point Matching:

with an infinite ground plane:



With a finite ground plane



Answer the following questions in a few sentences in the conclusions: Which testing procedure is more efficient and accurate? Why?

Between the two, the Galerkin method is more efficient and accurate. You can see that both methods approach a convergence rate near 1, but Galerkin consistently gives smaller errors for the same segment size Δ. This is because Galerkin uses weighted testing functions that better represent the continuous charge distribution, reducing numerical error. The Point Matching method still converges, but it needs finer segmentation (smaller Δ) to reach the same accuracy, so it's a bit less efficient overall.