PROJECT 1

ECE 541 - Applied Electromagnetics

Elim Aschenberg

9/27/2025

I. Method of Moments (MoM) Analysis of an Air-Filled Microstrip Transmission Line with an Infinite Ground Plane – Using Point Matching Testing Procedure

1).

Write the theory and equations, and describe your MoM algorithm

Theory of point matching:

The theory of point matching is that by dividing the transmission line into several segments, we can find the charge density distribution, capacitance per unit length, and several other parameters. This is done by taking the center of each segment, finding its desired value, and then doing it for all the remaining segments on the transmission line.

Equations:

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

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

$$d = 2h$$

$$Z_{ij} = \frac{1}{2\pi\epsilon_0}\left[K_1\left(\frac{\Delta}{2},\, x_i - x_j\,,\, 0\right) - K_1\left(\frac{\Delta}{2},\, x_i - x_j\,,\, d\right)\right]$$

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

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

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

Empirical Formulas:

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

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

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

‘

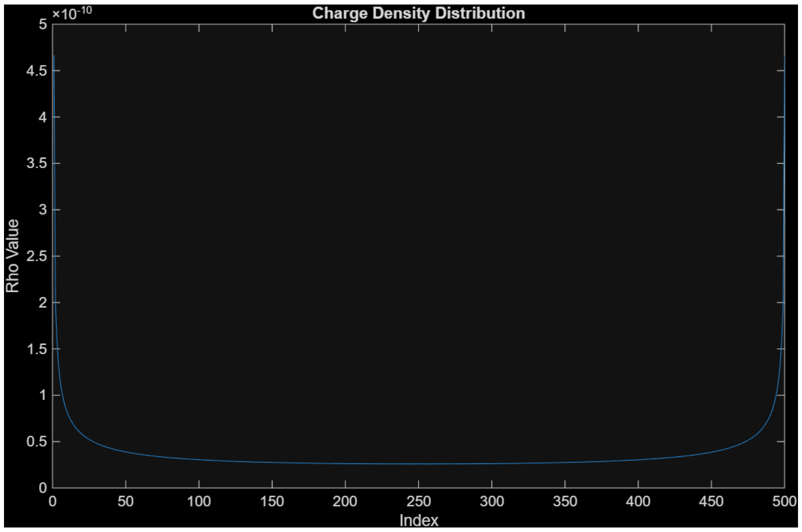
Describe the MoM algorithm:

Using a for loop, I created a $N \times N$ matrix for z(i,j). Then, knowing that voltage is constant across the transmission line, multiplied the inverse z(i,j) matrix by a voltage matrix of constant value. This gave the charge distribution across the transmission line. Using that data, I was able to then find Q’ and C’ as well. From there, I empirically solved for the same values, using different values for h to comply with the standard of having w/h ratios of <1, =1, and >1. I then compared my values for C’ from the empirical solution to the answers I got through point matching to achieve a %error of <1% in all cases.

2).

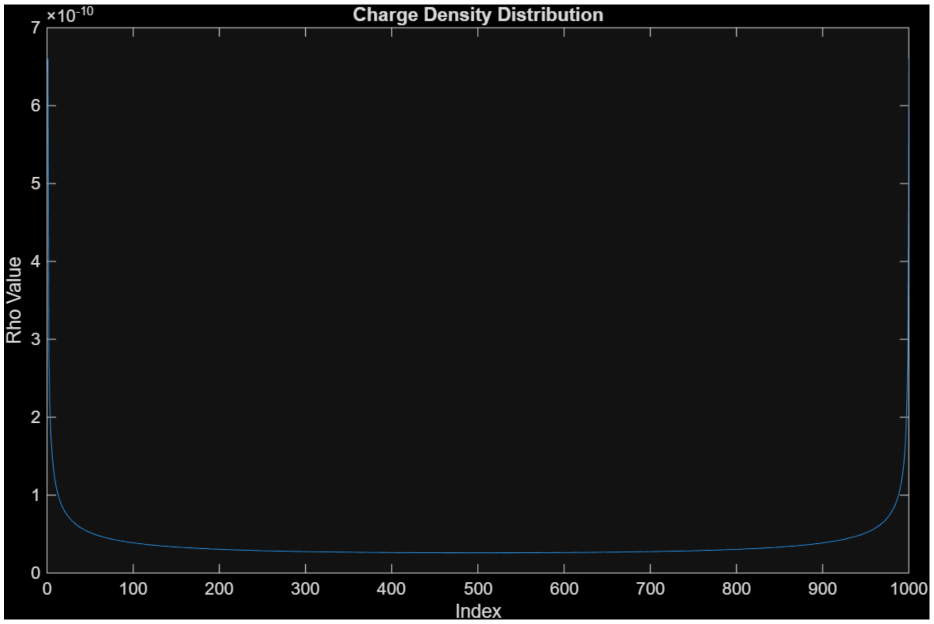
Find and plot the distribution of surface charges, ρ_s, along the upper strip, for several characteristic values of the w/h ratio, including cases w/h < 1, w/h = 1, and w/h > 1, and several values of N

Figure 1:



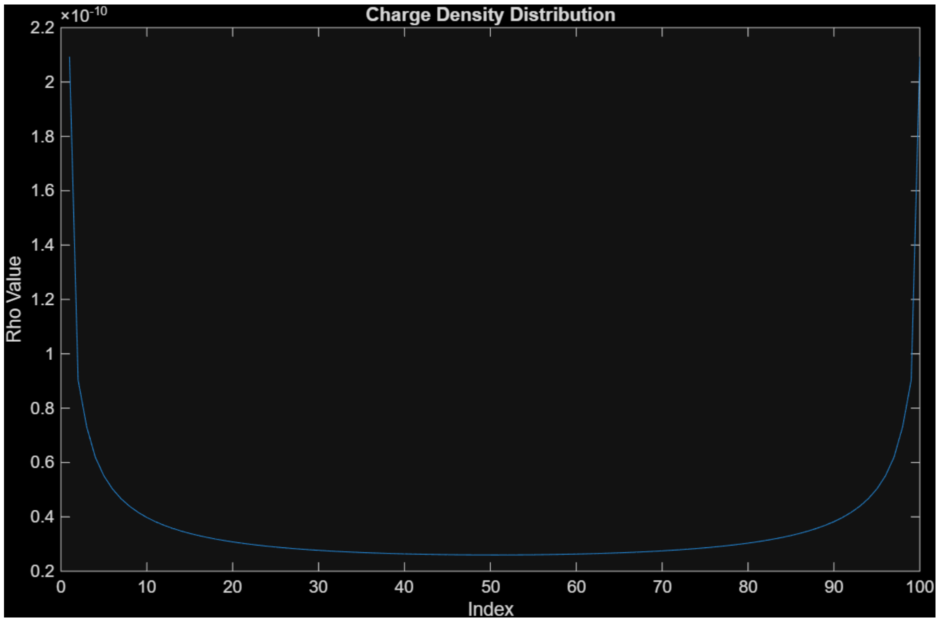
Characteristics of graph: w/h = 2, N = 500

Figure 2:



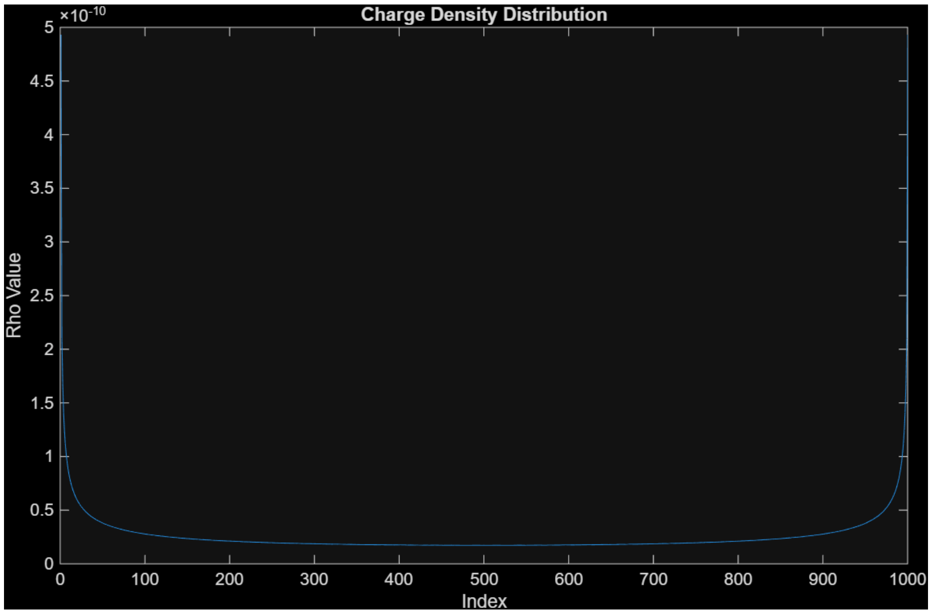
Characteristics of graph: w/h = 2, N = 1000

Figure 3:



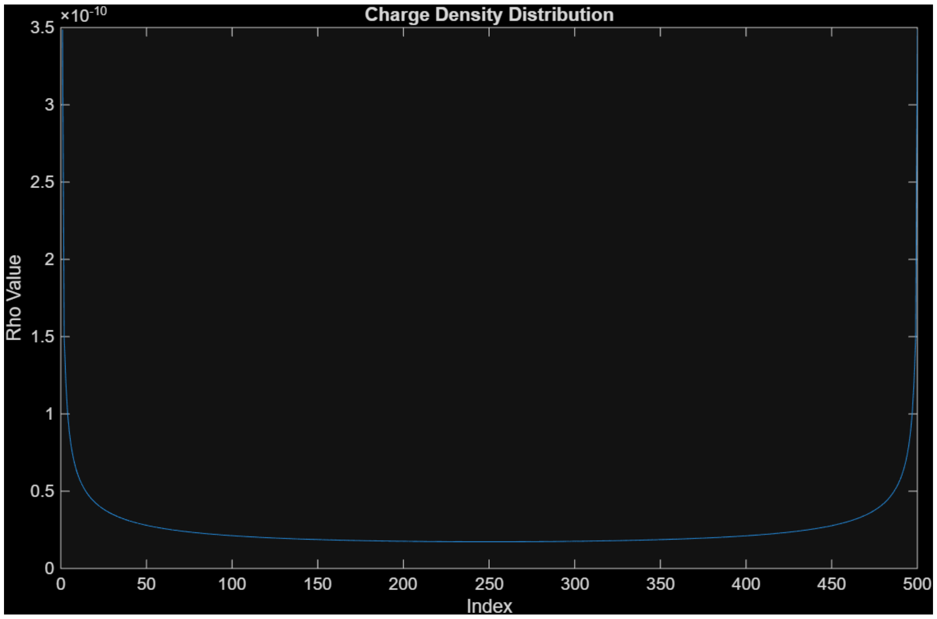
Characteristics of graph: w/h = 2, N = 100

Figure 4:



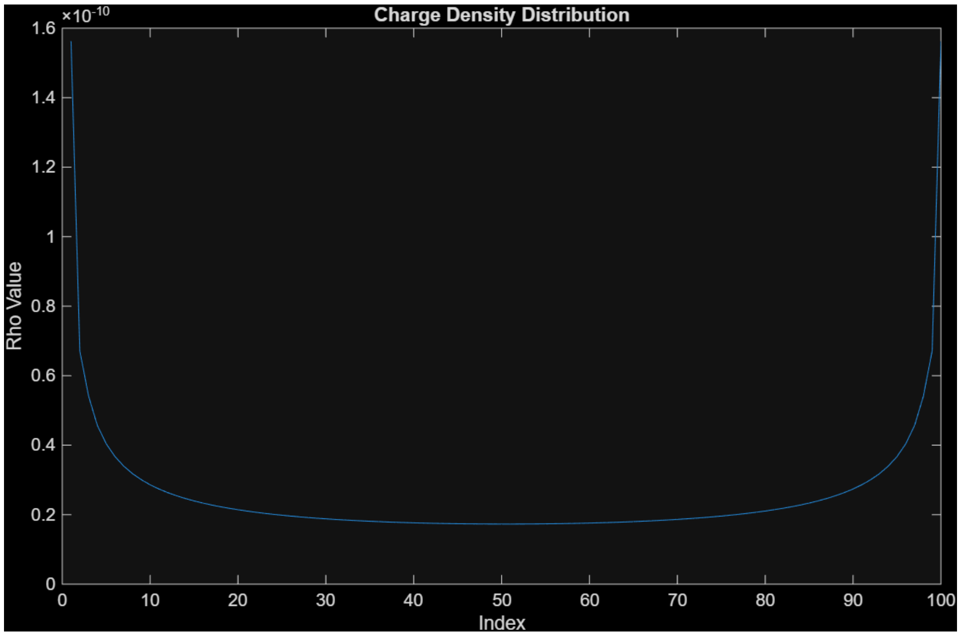
Characteristics of graph: w/h = 1, N = 1000

Figure 5:



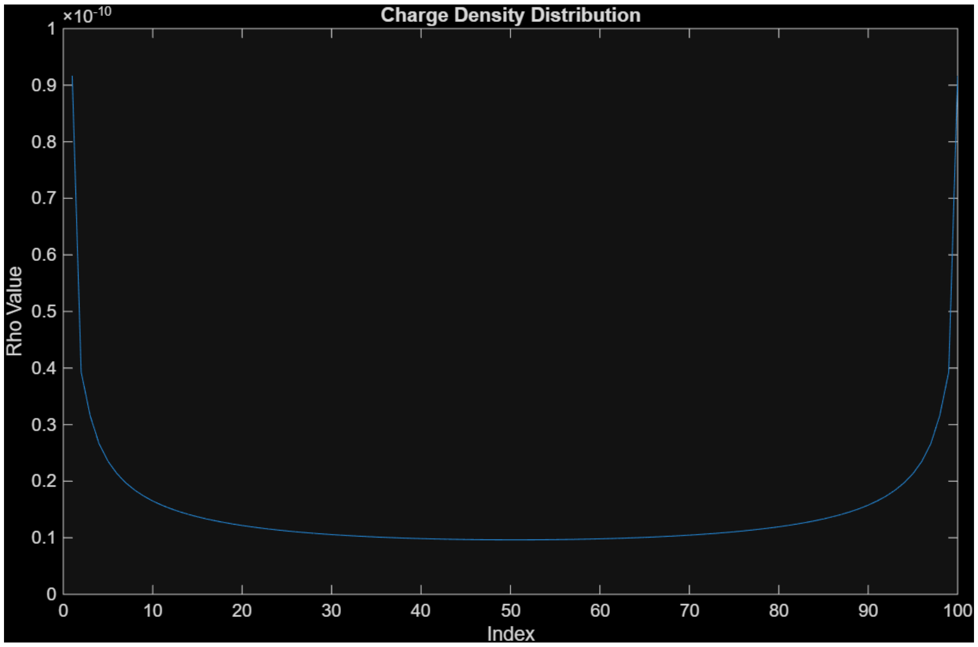
Characteristics of graph: w/h = 1, N = 500

Figure 6:



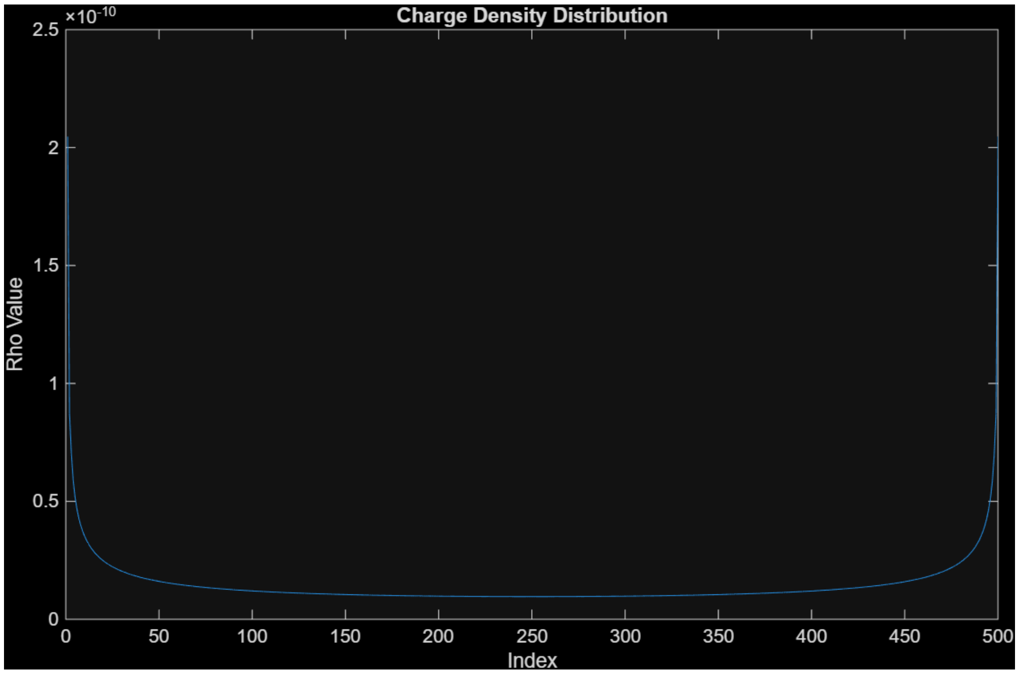
Characteristics of graph: w/h = 1, N = 100

Figure 7:



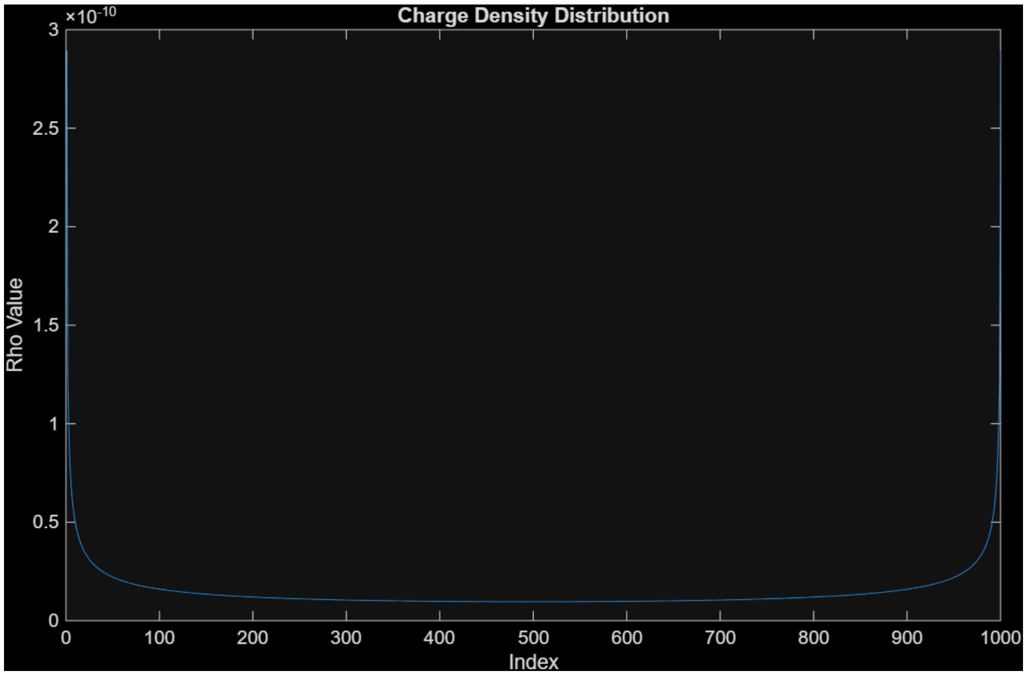
Characteristics of graph: w/h = 1/5, N = 100

Figure 8:



Characteristics of graph: w/h = 1/5, N = 500

Figure 9:

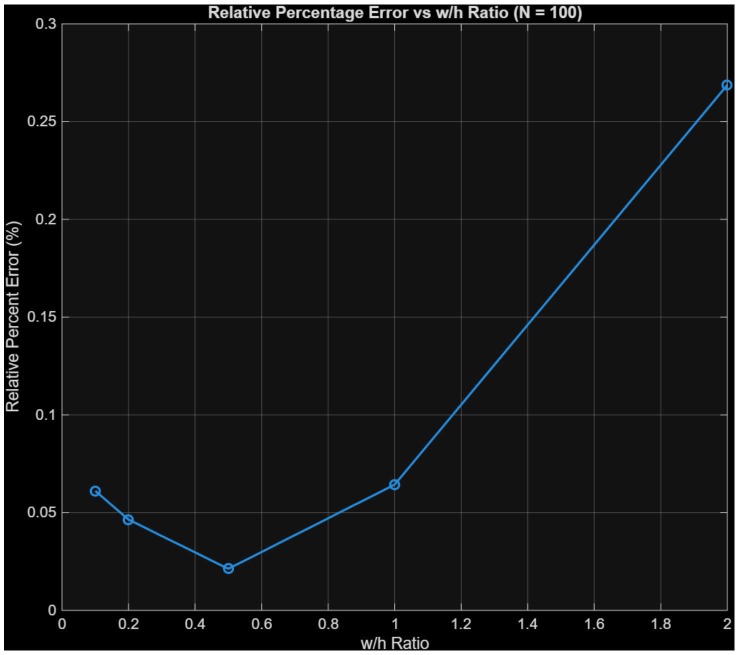


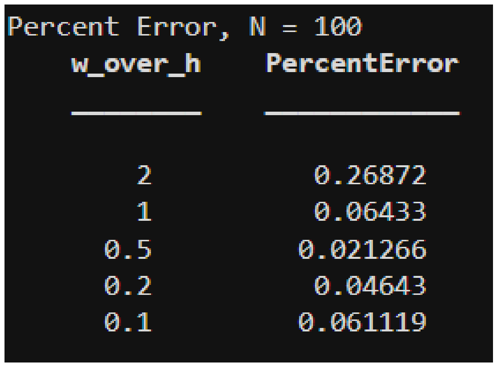
Characteristics of graph: L w/h = 1/5, N = 1000

3).

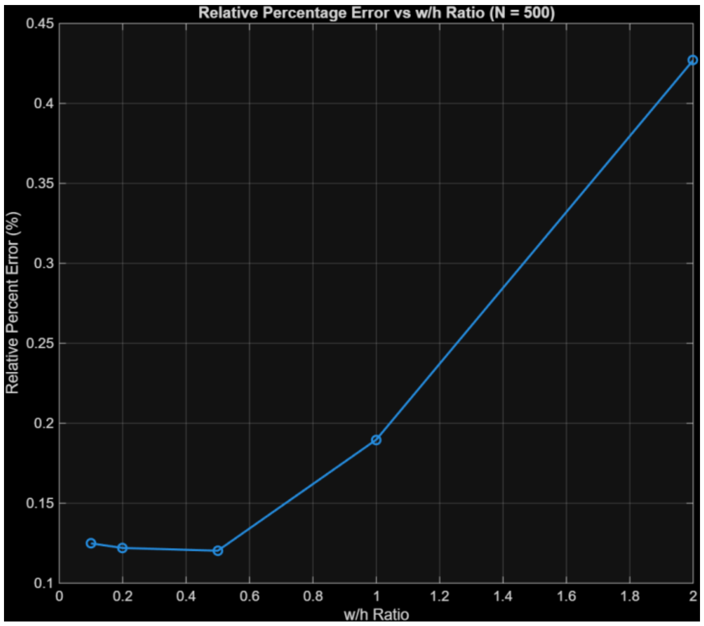
For several characteristic values of the w/h ratio, find the capacitance per unit length of the transmission line, C’, and tabulate (in a table) and plot (as a graph) the relative percentage error when compared to the result obtained by empirical formulas (p.569, Electromagnetics, Notaros –attached here)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

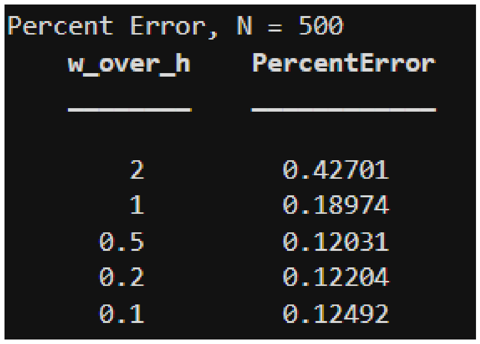
Figure 10-15 : Relative % Error with N = (100, 500, 1000) (Respectively) using point matching; with tables



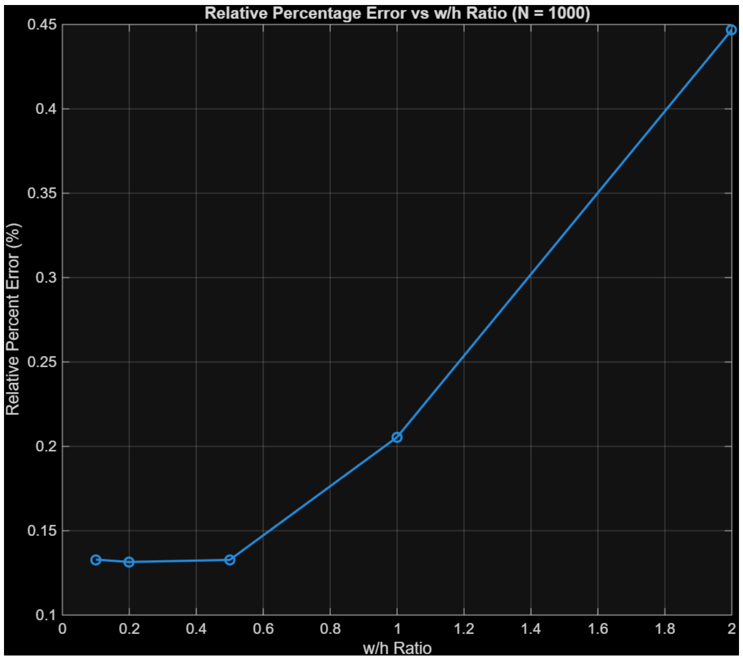


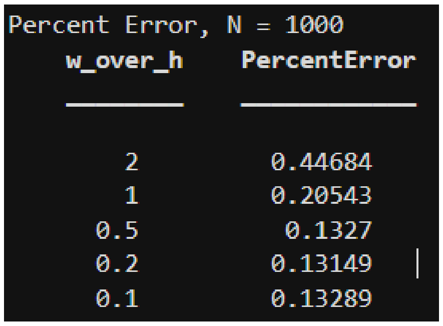
```
Percent Error, N = 100
    w_over_h      PercentError
    ________      ____________

        2            0.26872
        1            0.06433
      0.5           0.021266
      0.2            0.04643
      0.1           0.061119
```





```
Percent Error, N = 500
    w_over_h      PercentError
    ________      ____________

        2            0.42701
        1            0.18974
      0.5            0.12031
      0.2            0.12204
      0.1            0.12492
```



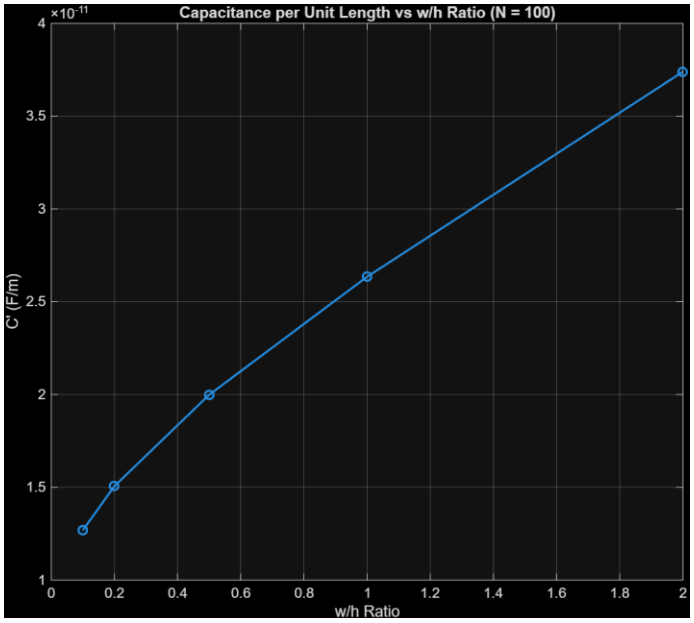


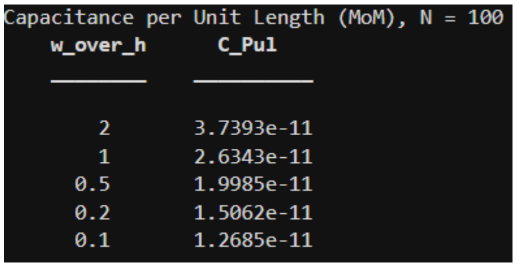
```
Percent Error, N = 1000
     w_over_h      PercentError
     ________      ____________

         2           0.44684
         1           0.20543
       0.5            0.1327
       0.2           0.13149
       0.1           0.13289
```

Discuss the convergence of MoM results with increasing N:

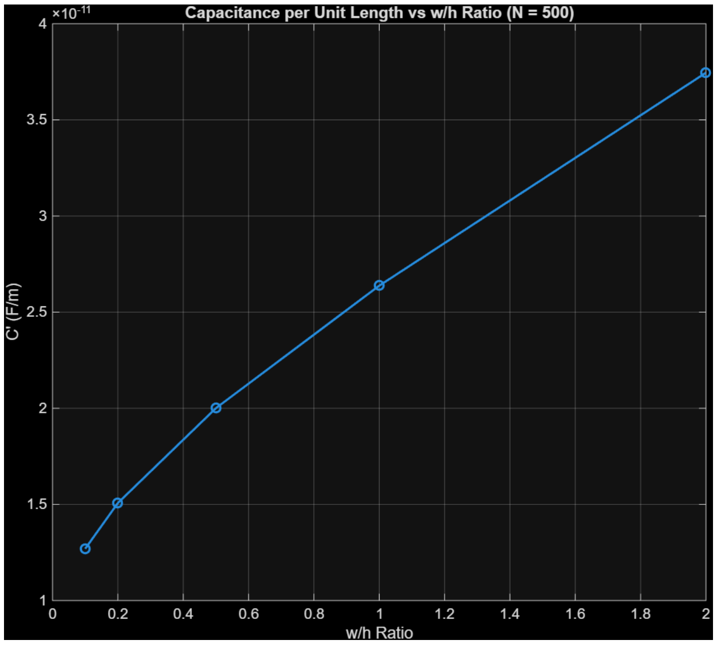
The %error graph becomes more linear as N increases. Additionally, the % Error increases as N increases. On the upper limit (w/h = 2), the graph appears to converge to 0.45% error as N increases.

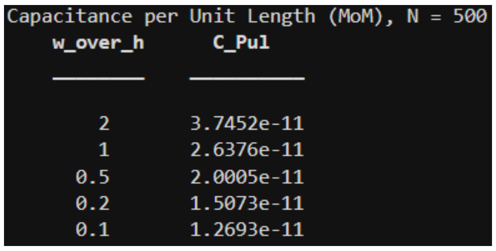
Figures 16-21: C' with respect to w/h and N = (100, 500, 1000) (Respectively) using point matching; with tables



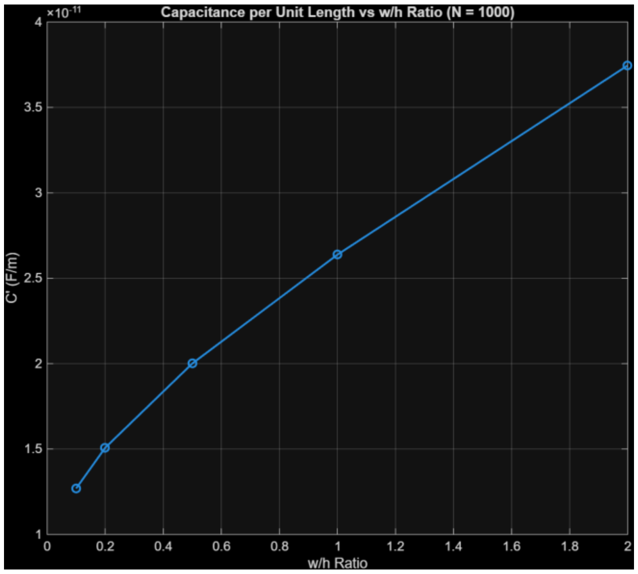


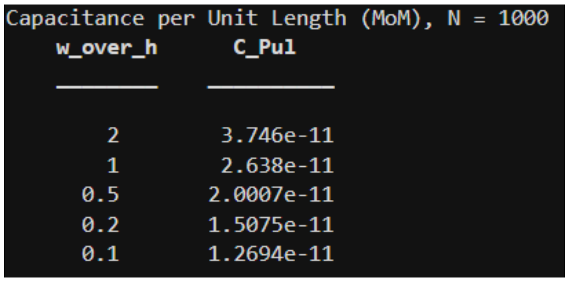
```
Capacitance per Unit Length (MoM), N = 100
    w_over_h        C_Pul    
    ________    ___________

        2       3.7393e-11
        1       2.6343e-11
      0.5       1.9985e-11
      0.2       1.5062e-11
      0.1       1.2685e-11
```





```
Capacitance per Unit Length (MoM), N = 500
    w_over_h        C_Pul    
    ________    ___________

        2       3.7452e-11
        1       2.6376e-11
      0.5       2.0005e-11
      0.2       1.5073e-11
      0.1       1.2693e-11
```





```
Capacitance per Unit Length (MoM), N = 1000
    w_over_h        C_Pul
    ________     ___________

        2          3.746e-11
        1          2.638e-11
      0.5         2.0007e-11
      0.2         1.5075e-11
      0.1         1.2694e-11
```

Discuss the convergence of MoM results with increasing N:

The graphs tend to retain their shape as N increases. Interestingly, as N increases, the upper and lower bounds of the graph (w/h = 2, 0.1, respectively) marginally increase.

4).

Provide a complete listing of your MATLAB code.

```
function K = K1_function(a,x,y)
z = x + 1i*y;
K = real((z-a)*(log(z-a)-1)) - real((z+a)*(log(z+a)-1));
end
% Define Constants
epsilon0 = 8.8541878188*10^-12;
w = 1;
N = 500;
h = 1;
eta0 = 377;
c0 = 3E8;
epsilon_r = 1;
v1=1;
v2 = 0;
% Initalize variables
delta = w/N;
d = 2*h;
% intialize matrices
v_matrx = v1*ones(1,N);
```

```
z = zeros(N,N);
% Populate the matrix z with some values <- Gemini prompt (MatLab R2025b)
for i = 1:N
xi = delta*i;
for j = 1:N
xj = delta*j;
z(i,j) = 1/(2*pi*epsilon0)*(K1_function(delta/2, xi-xj, 0)-K1_function(delta/2, xi-
xj,d));
end
end
% create charge distribution matrix
Rho = v_matrx * inv(z);
% Plotting
figure;
plot(Rho);
title('Charge Density Distribution for point matching');
xlabel('Index');
ylabel('Rho Value');
% Calculate C'
Charge_Pul = sum(Rho * delta);
Cap_Pul = Charge_Pul/(v1-v2);
% Begin empirical comparison by initalizing matricies
h_val = [0.5,1,2,5,10];
results_vector = zeros(2,length(h_val));
z0_emp = zeros(1, length(h_val));
Cap_Pul_emp = zeros(1, length(z0_emp));
Percent_Error = zeros(2, length (h_val));
for k = 1: length(h_val)
h = h_val(k);
ratio = w/h;
% Empirical Calculations
if ratio < 1
p = 0.04*(1-ratio)^2;
else
p = 0;
end
epsilon_reff = (epsilon_r + 1)/2 + ((epsilon_r - 1)/2) * ((1 + 12/ratio)^-(1/2) + p);
if ratio > 1
z0_emp (k) = eta0/sqrt(epsilon_reff) * (ratio + 1.393 + 0.667 * log (ratio +
1.444))^-1;
Cap_Pul_emp (k) = 1/(z0_emp (k) * c0);
else
z0_emp (k) = eta0/(2*pi*sqrt(epsilon_reff)) * log (8/ratio + ratio/4);
Cap_Pul_emp (k) = 1/(z0_emp (k) * c0);
end
% Redefine d for new h values
d = 2*h;
```

```
for i = 1:N
xi = delta*i;
for j = 1:N
xj = delta*j;
% recalculate z for every new h
z(i,j) = 1/(2*pi*epsilon0)*(K1_function(delta/2, xi-xj, 0)-K1_function(delta/2, xi-
xj,d));
end
end
Rho = v_matrx * inv(z);
Charge_Pul = 0;
for j = 1:N
Charge_Pul = Charge_Pul + (Rho(j) * delta) ;
end
% Calculate %error
Cap_Pul = Charge_Pul/(v1-v2);
results_vector(1, k) = ratio;
results_vector(2, k) = Cap_Pul;
Percent_Error (1, k) = ratio;
Percent_Error(2, k) = abs(results_vector(2,k)-Cap_Pul_emp(k)) / Cap_Pul_emp(k) * 100;
end
% ----- Plot C' vs w/h -----
figure;
plot(results_vector(1,:), results_vector(2,:),'o-','LineWidth',1.5);
grid on;
xlabel('w/h Ratio');
ylabel('C'' (F/m)');
title(sprintf('Capacitance per Unit Length vs w/h Ratio (N = %d)', N));
% ----- Plot % Error vs w/h -----
figure;
plot(Percent_Error(1,:), Percent_Error(2,:),'o-','LineWidth',1.5);
grid on;
xlabel('w/h Ratio');
ylabel('Relative Percent Error (%)');
title(sprintf('Relative Percentage Error vs w/h Ratio (N = %d)', N));
% ----- Labeled tables -----
fprintf('Capacitance per Unit Length (MoM), N = %d\n', N);
T_Cpul = table(results_vector(1,:).', results_vector(2,:).', ...
'VariableNames', {'w_over_h','C_Pul'});
disp(T_Cpul);
fprintf('Percent Error, N = %d\n', N);
T_err = table(Percent_Error(1,:).', Percent_Error(2,:).', ...
'VariableNames', {'w_over_h','PercentError'});
disp(T_err);
```

II. Method of Moments (MoM) Analysis of an Air-Filled Microstrip Transmission Line with an Infinite Ground Plane – Using Galerkin Testing Method

1).

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

Theory of Galerkin method:

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

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

$$d\ =\ 2h$$

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

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

$$Z_{ij}\ =\ -\frac{1}{2\pi\varepsilon_0}\big[K2\big(a,\ x_i\ -\ (x_j+a),\ 0\ ,\ x_i\ -\ (x_j-a),\ 0\big) - K_2\big(a,\ x_i\ -\ (x_j+a),\ d\ ,\ x_i\ -\ (x_j-a),\ d\big)\big]$$

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

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

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

Empirical Formulas:

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

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

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

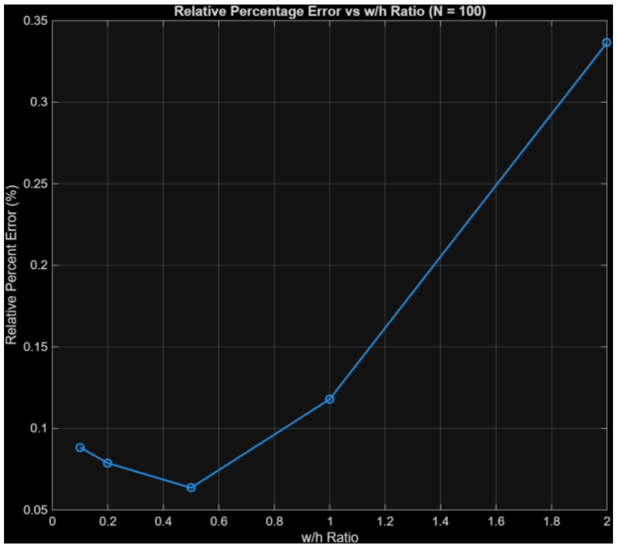
Describe the MoM algorithm:

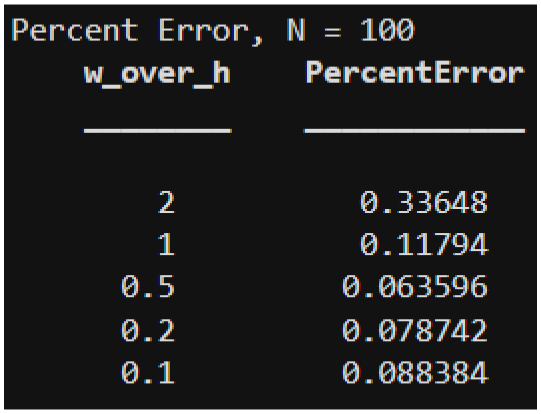
Using a for loop, I created a N×N matrix for z(i,j). Then, knowing that voltage is constant across the transmission line, multiplied the inverse z(i,j) matrix by a voltage matrix of constant value. This gave the charge distribution across the transmission line. Using that data, I was able to then find Q' and C' as well. From there, I empirically solved for the same values, using different values for h to comply with the standard of having w/h ratios of <1, =1, and >1. I then compared my values for C' from the empirical solution to the remaining segments on the transmission line. This process is nearly identical to point matching.

2).

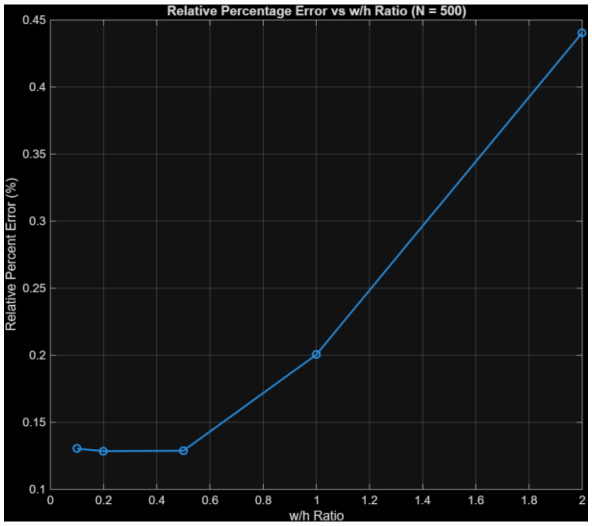
For several characteristic values of the w/h ratio (w/h < 1, w/h = 1, and w/h > 1), find the capacitance per unit length of the transmission line, C’, and tabulate (in a table) and plot (as a graph) the relative percentage error when compared to the result obtained by empirical formulas (p.569, Electromagnetics, Notaros – attached here) as a function of N. Discuss the convergence of MoM results with increasing N.

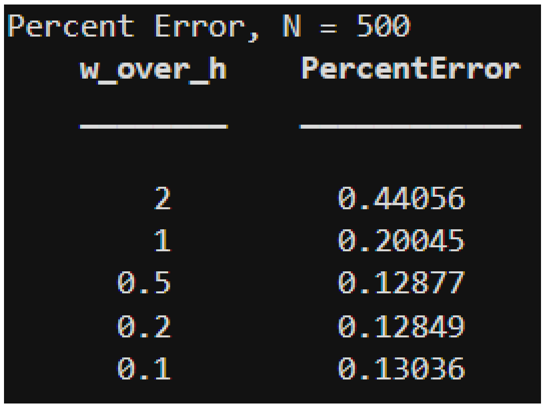
Figure 22-27 : Relative % Error with N = (100, 500, 1000) (Respectively) using point matching; with tables



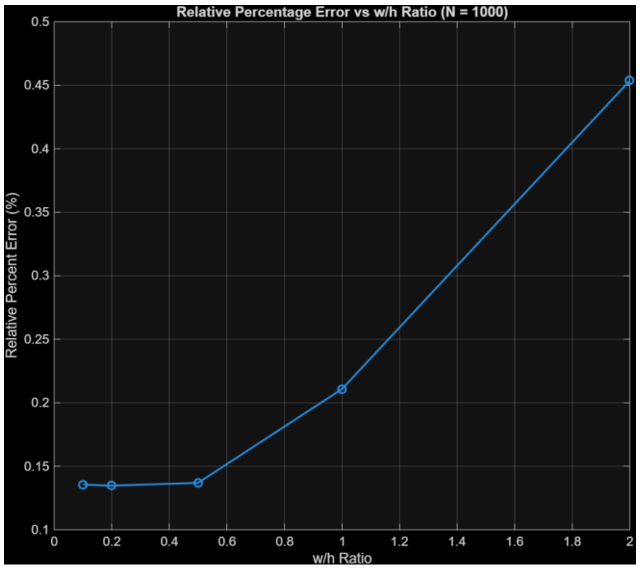


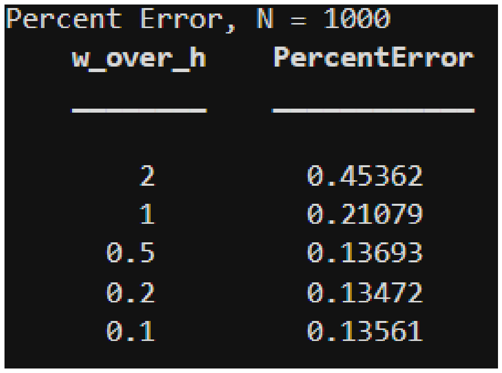
```
Percent Error, N = 100
     w_over_h      PercentError
     ________      ____________

        2            0.33648
        1            0.11794
      0.5           0.063596
      0.2           0.078742
      0.1           0.088384
```





```
Percent Error, N = 500
     w_over_h      PercentError
     ________      ____________

        2            0.44056
        1            0.20045
      0.5            0.12877
      0.2            0.12849
      0.1            0.13036
```



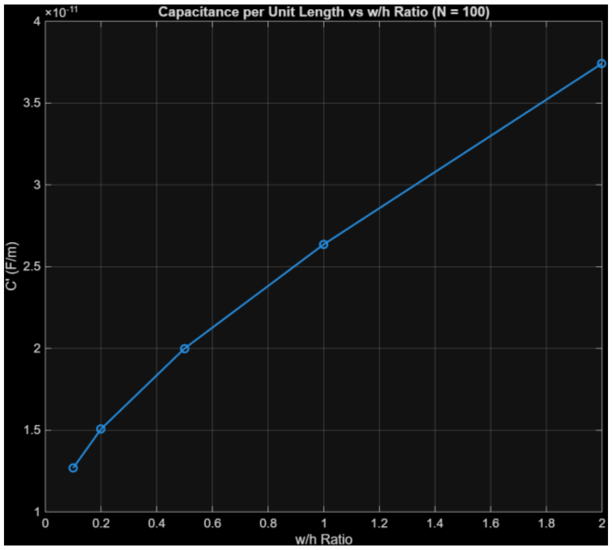


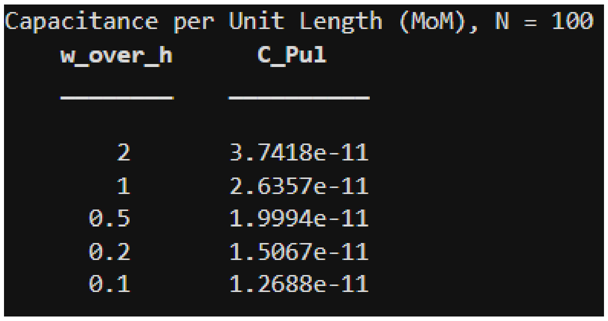
```
Percent Error, N = 1000
    w_over_h      PercentError
    ________      ____________

        2           0.45362
        1           0.21079
      0.5           0.13693
      0.2           0.13472
      0.1           0.13561
```

Discuss the convergence of MoM results with increasing N:

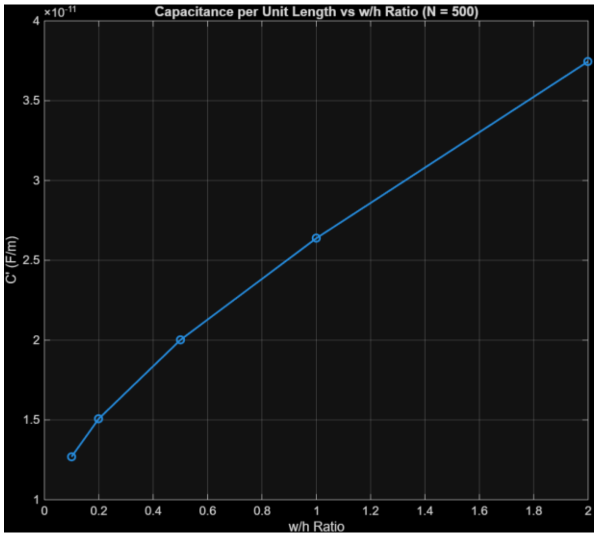
As N increases, the graph again becomes more linear. Additionally, the % Error also increases.

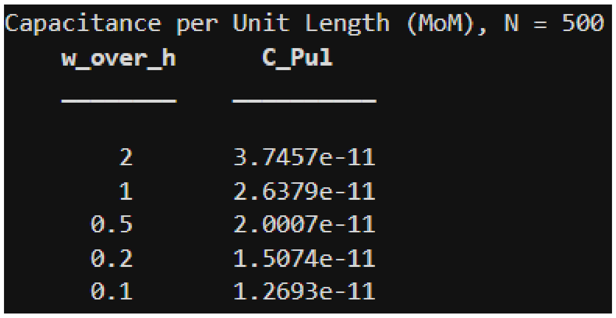
Figures 28-33: C' with respect to w/h and N = (100, 500, 1000) (Respectively) using Galerkin Method; with tables



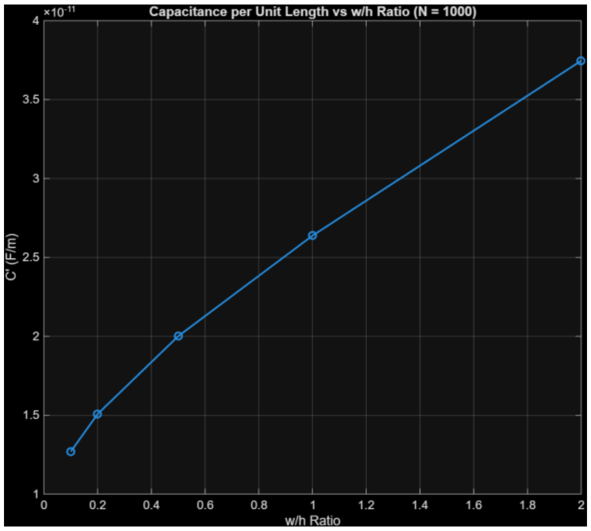


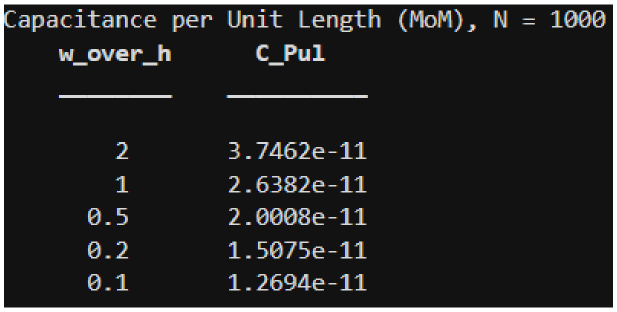
```
Capacitance per Unit Length (MoM), N = 100
    w_over_h       C_PuL
    ________    ___________

        2       3.7418e-11
        1       2.6357e-11
      0.5       1.9994e-11
      0.2       1.5067e-11
      0.1       1.2688e-11
```





```
Capacitance per Unit Length (MoM), N = 500
    w_over_h        C_Pul
    ________      __________

        2         3.7457e-11
        1         2.6379e-11
      0.5         2.0007e-11
      0.2         1.5074e-11
      0.1         1.2693e-11
```





```
Capacitance per Unit Length (MoM), N = 1000
    w_over_h        C_Pul
    ________      __________

        2         3.7462e-11
        1         2.6382e-11
      0.5         2.0008e-11
      0.2         1.5075e-11
      0.1         1.2694e-11
```

Discuss the convergence of MoM results with increasing N:

The shapes of the graphs remain relatively similar as N increases. Additionally, the percent error does increase with N, but the difference is almost negligible.

3).

Provide a complete listing of your MATLAB code.

```
function K = K2_function(a,x1,y1,x2,y2)
z1 = x1 + 1i*y1;
```

```
z2 = x2 + 1i*y2;
r = sqrt((x2 - x1)^2 + (y2 - y1)^2);
l_hat = ((x2 - x1) + 1i*(y2 - y1))/r;
% When using integral K2, the ln x × x2 is zero for x = 0, so you can just use if
statement
% to check whether argument is 0 and equate the whole expression to 0 in that case
if z2 + a == 0
F1 = 0;
else
F1 = 0.5*(z2 + a)^2*(log(z2 + a) - 3/2);
end
if z2 - a == 0
F2 = 0;
else
F2 = 0.5*(z2 - a)^2*(log(z2 - a) - 3/2);
end
if z1 + a == 0
F3 = 0;
else
F3 = 0.5*(z1 + a)^2*(log(z1 + a) - 3/2);
end
if z1 - a == 0
F4 = 0;
else
F4 = 0.5*(z1 - a)^2*(log(z1 - a) - 3/2);
end
K = real(conj(l_hat) * ((F1 - F2) - (F3 - F4)));
end
% initalizing constants
epsilon0 = 8.854187817e-12;
w = 1;
h = 1;
N = 1000;
v1 = 1;
v2 = 0;
epsilon_r = 1;
eta0 = 377;
c0 = 3E8;
% initalizing variables
d = 2 * h;
delta = w/N;
a = delta/2;
% creating matricies
x = zeros(N,1);
v = delta*ones(N,1);
z = zeros(N,N);
% calculate x positions at the center of each segment
```

```
for i = 1:N
x(i) = (i-0.5) * delta;
end
for i = 1:N
xi = x(i);
for j = 1:N
xj = x(j);
x_right = xi - (xj + a);
x_left = xi - (xj - a);
K_self = K2_function(a, x_right, 0, x_left, 0);
K_image = K2_function(a, x_right, d, x_left, d);
z(i,j) = -1/(2*pi*epsilon0) * (K_self - K_image);
end
end
% Z(i,j) = Z(j,i) -> matrix is symmetric
for i = 1:N
for j = 1:N
z(i,j) = 0.5 * (z(i,j) + z(j,i));
end
end
% Calculations for C'
Rho = z \ v;
Charge_Pul = sum(Rho * delta);
Cap_Pul = Charge_Pul/(v1-v2);
% Plotting
figure;
plot(x, Rho/max(abs(Rho)),'-');
grid on
xlabel('x');
ylabel('\rho(x)');
title('Charge Density Distribution');
% Setting up matricies
h_val = [0.5,1,2,5,10];
results_vector = zeros(2,length(h_val));
z0_emp = zeros(1, length(h_val));
Cap_Pul_emp = zeros(1, length(z0_emp));
Percent_Error = zeros(2, length (h_val));
for k = 1: length(h_val)
h = h_val(k);
% Redefine variables reliant upon h
ratio = w/h;
d = 2 * h;
% empirical calulations
if ratio < 1
p = 0.04*(1-ratio)^2;
else
p = 0;
```

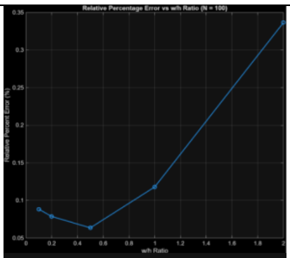
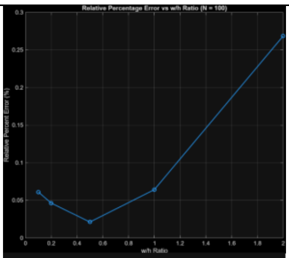
```
end
epsilon_reff = (epsilon_r + 1)/2 + ((epsilon_r - 1)/2) * ((1 + 12/ratio)^-(1/2) + p);
if ratio > 1
z0_emp (k) = eta0/sqrt(epsilon_reff) * (ratio + 1.393 + 0.667 * log (ratio +
1.444))^-1;
Cap_Pul_emp (k) = 1/(z0_emp (k) * c0);
else
z0_emp (k) = eta0/(2*pi*sqrt(epsilon_reff)) * log (8/ratio + ratio/4);
Cap_Pul_emp (k) = sqrt(epsilon_reff)/(z0_emp (k) * c0);
end
% Repeating the Galerkin code above
for i = 1:N
x(i) = (i-0.5) * delta;
end
for i = 1:N
xi = x(i);
for j = 1:N
xj = x(j);
x_right = xi - (xj + a); y1s = 0;
x_left = xi - (xj - a);
K_self = K2_function(a, x_right, 0, x_left, 0);
K_image = K2_function(a, x_right, d, x_left, d);
z(i,j) = -1/(2*pi*epsilon0) * (K_self - K_image);
end
end
for i = 1:N
for j = 1:N
z(i,j) = 0.5 * (z(i,j) + z(j,i));
end
end
Rho = z \ v;
Charge_Pul = 0;
for j = 1:N
Charge_Pul = Charge_Pul + (Rho(j) * delta);
end
% Setting up for %error calculation
Cap_Pul = Charge_Pul/(v1-v2);
results_vector(1, k) = ratio;
results_vector(2, k) = Cap_Pul;
Percent_Error (1, k) = ratio;
Percent_Error(2, k) = abs(results_vector(2,k)-Cap_Pul_emp(k))...
/ Cap_Pul_emp(k) * 100;
end
% %Error plot with N in title
figure;
plot(Percent_Error(1,:), Percent_Error(2,:),'o-','LineWidth',1.5); grid on;
xlabel('w/h Ratio');
```

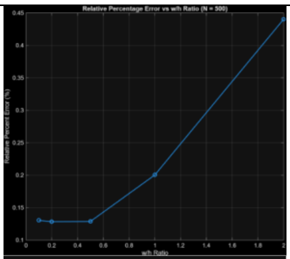
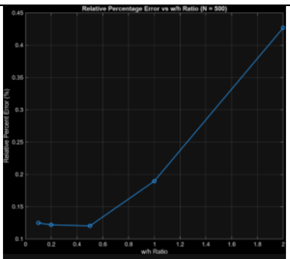
```
ylabel('Relative Percent Error (%)');
title(sprintf('Relative Percentage Error vs w/h Ratio (N = %d)', N));
% C' vs w/h with N in title
figure;
plot(results_vector(1,:), results_vector(2,:),'o-','LineWidth',1.5); grid on;
xlabel('w/h Ratio');
ylabel('C'' (F/m)');
title(sprintf('Capacitance per Unit Length vs w/h Ratio (N = %d)', N));
% labeled tables with N in header
fprintf('Capacitance per Unit Length (MoM), N = %d\n', N);
T_Cpul = table(results_vector(1,:).', results_vector(2,:).', ...
'VariableNames', {'w_over_h','C_Pul'});
disp(T_Cpul);
fprintf('Percent Error, N = %d\n', N);
T_err = table(Percent_Error(1,:).', Percent_Error(2,:).', ...
'VariableNames', {'w_over_h','PercentError'});
disp(T_err);
```

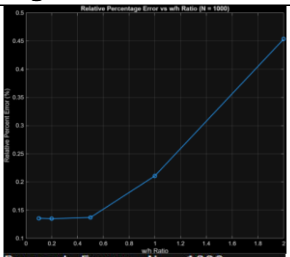
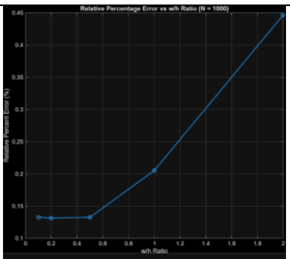
III. Convergence comparison of different testing techniques: point matching and Galerkin

1).

Compare the relative percentage errors obtained by the point matching method and the Galerkin method, respectively (approaches I and II). As the reference result, use the empirical formulas (p.569, Electromagnetics, Notaros – attached here). Have in mind that $Z_0 = 1 / (cC')$ where in your case $c = c_0$ = speed of light in free space. Also in your case, $\varepsilon_{reff} = 1$ (air filled MS line) For each approach, use a single set of characteristic values.

# of Trials	Galerkin Method	Point Matching	Comparison
100	 Percent Error, N = 100 w_over_h PercentError ________ ____________ 2 0.33648 1 0.11794 0.5 0.063596 0.2 0.078742 0.1 0.088384 Figures 22 and 23	 Percent Error, N = 100 w_over_h PercentError ________ ____________ 2 0.26872 1 0.06433 0.5 0.021266 0.2 0.04643 0.1 0.061119 Figures 10 and 11	The % Error in the Galerkin Method is higher than the point matching for every case of w/h.

500	
	`Percent Error, N = 500`
	`w_over_h    PercentError`
	`________    ____________`
	`   2          0.44056`
	`   1          0.20045`
	` 0.5          0.12877`
	` 0.2          0.12849`
	` 0.1          0.13036`
	Figures 24 and 25
		
		`Percent Error, N = 500`
		`w_over_h    PercentError`
		`________    ____________`
		`   2          0.42701`
		`   1          0.18974`
		` 0.5          0.12031`
		` 0.2          0.12204`
		` 0.1          0.12492`
		Figures 12 and 13
			The % Error in the Galerkin Method is higher than the point matching for every case of w/h. However, as N increases, the difference in error shrinks.

1000	
	`Percent Error, N = 1000`
	`w_over_h    PercentError`
	`________    ____________`
	`   2          0.45362`
	`   1          0.21079`
	` 0.5          0.13693`
	` 0.2          0.13472`
	` 0.1          0.13561`
	Figures 26 and 27
		
		`Percent Error, N = 1000`
		`w_over_h    PercentError`
		`________    ____________`
		`   2          0.44684`
		`   1          0.20543`
		` 0.5           0.1327`
		` 0.2          0.13149`
		` 0.1          0.13289`
		Figures 14 and 15
			The % Error in the Galerkin Method is higher than the point matching for every case of w/h. However, as N increases, the difference in error shrinks. At 1000 subdivisions, the accuracy is nearly identical.

Note: For these graphs/tables, h, w, ε_{reff}= 1 , c = c_0 = 3E8, and $Z_0 = \frac{1}{cC'}$

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 Δ (discretization segment lengths). Convergence rates should be calculated as follows: $\frac{\log_2\left(\frac{err_1}{err_{i+1}}\right)}{\log_2\left(\frac{\Delta_i}{\Delta_{i+1}}\right)}$ where err_1, err_2, … , err_i … are in descending order.

Figure 34

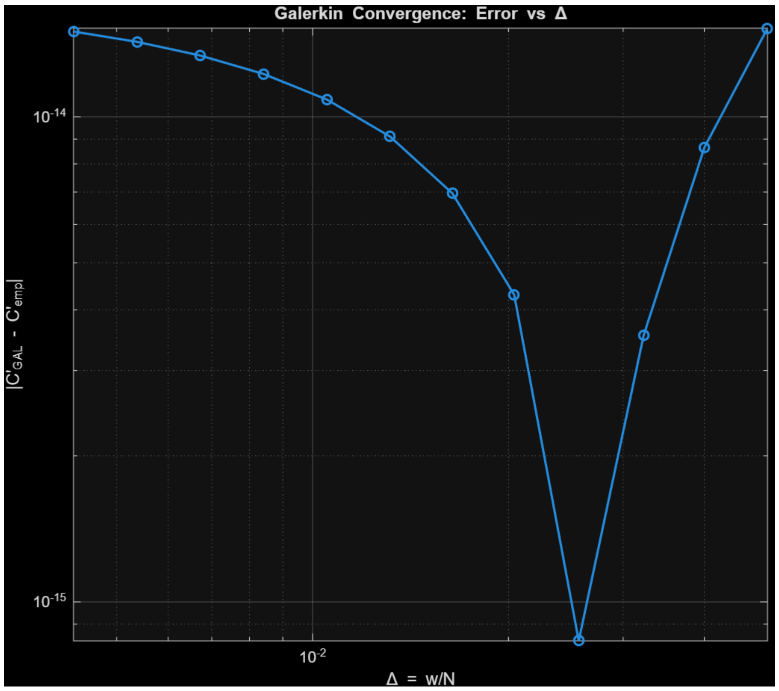
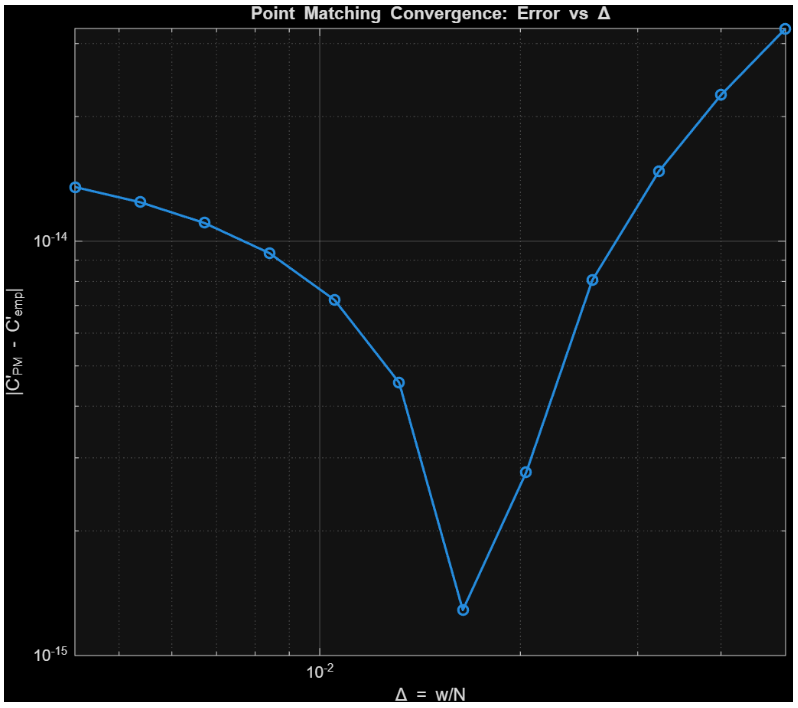

Galerkin Convergence: Error vs Δ
10^{-14}
10^{-15}
$|C'_{GAL} - C'_{empl}|$
10^{-2}
Δ = w/N

Figure 35


Point Matching Convergence: Error vs Δ
10^{-14}
10^{-15}
$|C'_{PM} - C'_{empl}|$
10^{-2}
Δ = w/N

3).

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

Galerkin is more efficient and accurate. For the same Δ, its error drops and it becomes precise. Additionally, is has a smoother and steeper decay than point matching. Point matching requires smaller Δ to reach similar precision and has more variability. This is because Galerkin minimizes any unused parts in an averaged way and giving a more balanced system, while point-matching only checks at certain points and ends up being more sensitive to local errors.