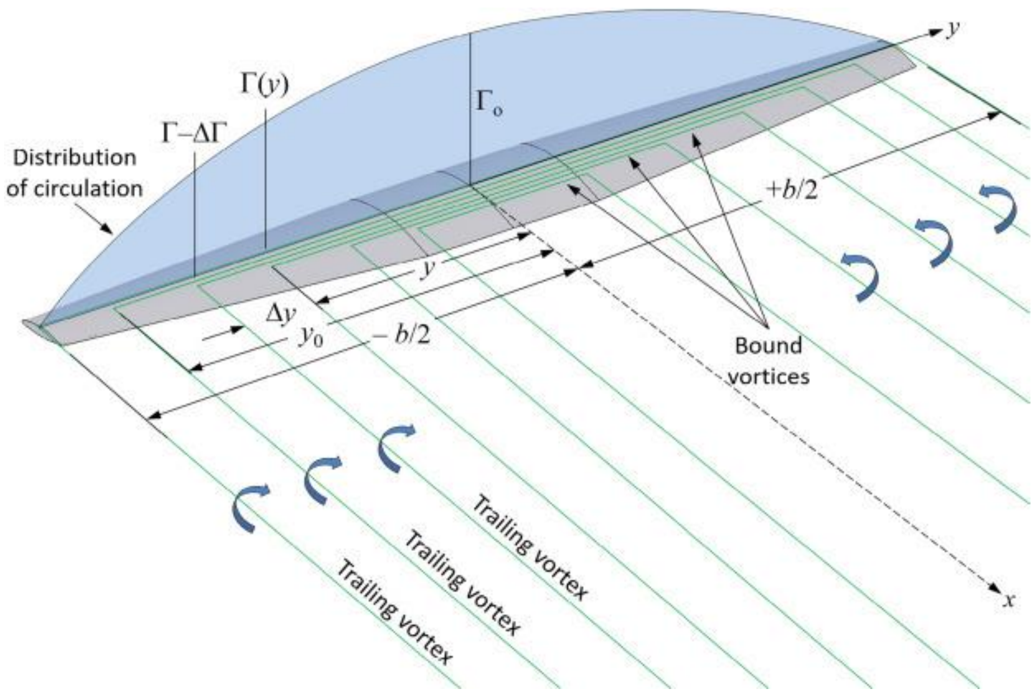
Numerical Lifting Line theory (NLLT) Computer Program Project

Class: ARO 3011 – Fluid Mechanics & Low Speed Aerodynamics

Section #02

Instructor: Dr. Tony Lin



Justin T. Millsap

Aerospace Engineering Department

California Polytechnic University, Pomona

November 30th, 2023

Summary:

The numerical lifting line theory is used to analyze and predict aerodynamic characteristics of a wing. This method represents a wing by a number of horseshoe vortices each with a different bound vortex.

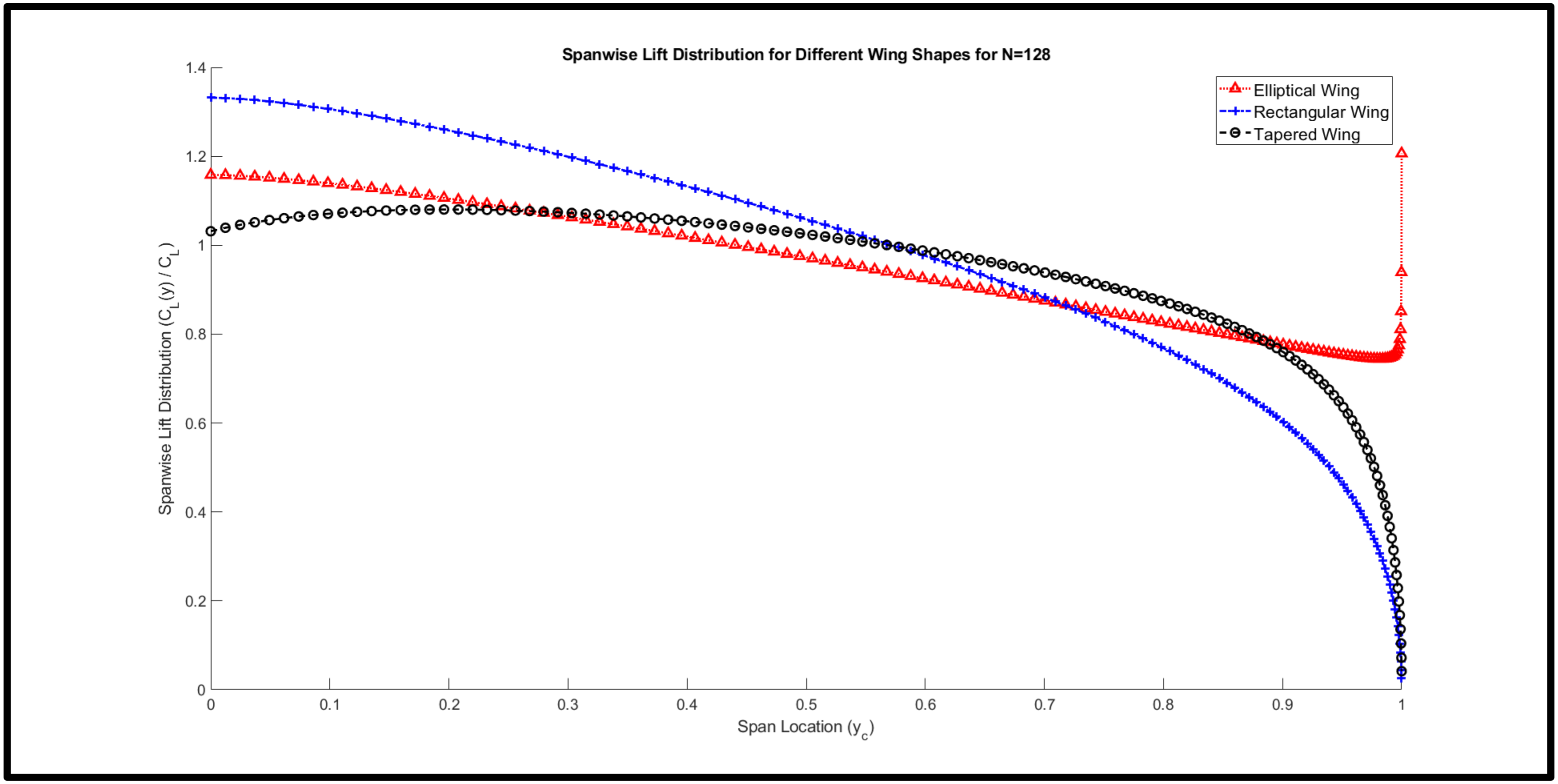
A computer program was created to computer total wing lift coefficient (C_L), total wing induced drag coefficient (C_{Di}), spanwise lift distribution normalized with total wing lift coefficient $\left({}^{C_l(y)}/_{C_L}\right)$, spanwise lift distribution of bound circulation $\Gamma(y)$, and δ. This program was created on MATLAB and can calculate these values for 3 different geometries such as a rectangular wing, tapered wing, and an elliptical wing.

The approach to create this code started off by defining known values consisting of twist (ε), angle of attack (α), aspect ratio, wingspan (b), and section-lift curve slope $(C_{L\alpha})$. The distance between each of the horseshoe vortices ($y_{c,i}$), were distributed into a "cosine space" fashion in order to get more precise data at the wing tips. We then set control points ($y_{c,i}$) along these cosine spaced horseshoe vortices. The angle of attack $\alpha(j)$ will vary at each $y_{c,i}$ value depending on the twist of the wing. After we obtained these values the code then proceeds by iterating values for the $C_{i,j}$ which will be used to iterate values for B_j and $A_{j,i}$ which are nxn matrices in which are vital in order to compute the vortex strengths (γ_i) using the governing equation $\gamma_i = \left[A_{j,i}\right]^{-1}[B_j]$. The code then calculates the down-wash velocity & down-wash angle of attack $(w_j$ & $(\alpha_{down-wash})_j)$. With all these values obtained above, the code is finalized with calculating (C_L), (C_{Di}), $\left({}^{C_l(y)}/_{C_L}\right)$, $\Gamma(y)$, and δ using equations in the numerical lifting line theory.

From the values produced by the computer program above, plots were produced that create a story of how different wing geometries result in different aerodynamic behaviors. To do this, the computer program generates a plot of ${}^{C_l(y)}/_{C_L})$ vs. y_c as well as $\Gamma(y)$ vs y_c which are listed below.

Upon examining the generated plots, it was evident that the rectangular wing exhibited the highest spanwise lift distribution with respect to the control points. However, it also converged to zero rapidly, resulting in the lowest spanwise bound circulation. The elliptical wing displayed the second highest spanwise lift distribution relative to the control points but encountered a singularity at $y_c = 1$, accompanied by the second highest spanwise bound circulation. Lastly, the tapered wing demonstrated the lowest spanwise lift distribution while having the highest spanwise bound circulation.

In conclusion, various wing geometries come with their own set of advantages and disadvantages. The selection of a particular wing geometry should be pinpointed depending on specific objectives. The numerical lifting line theory proves to be a precise method for comprehending diverse aerodynamic behaviors, which vary based on the distinct characteristics of a wing. This outlines the importance of employing accurate techniques to analyze and visualize how different wing features impact overall performance.



<u>**Figure 1: Spanwise Lift Distribution vs y_c for Different Wing Shapes for N=128**</u>

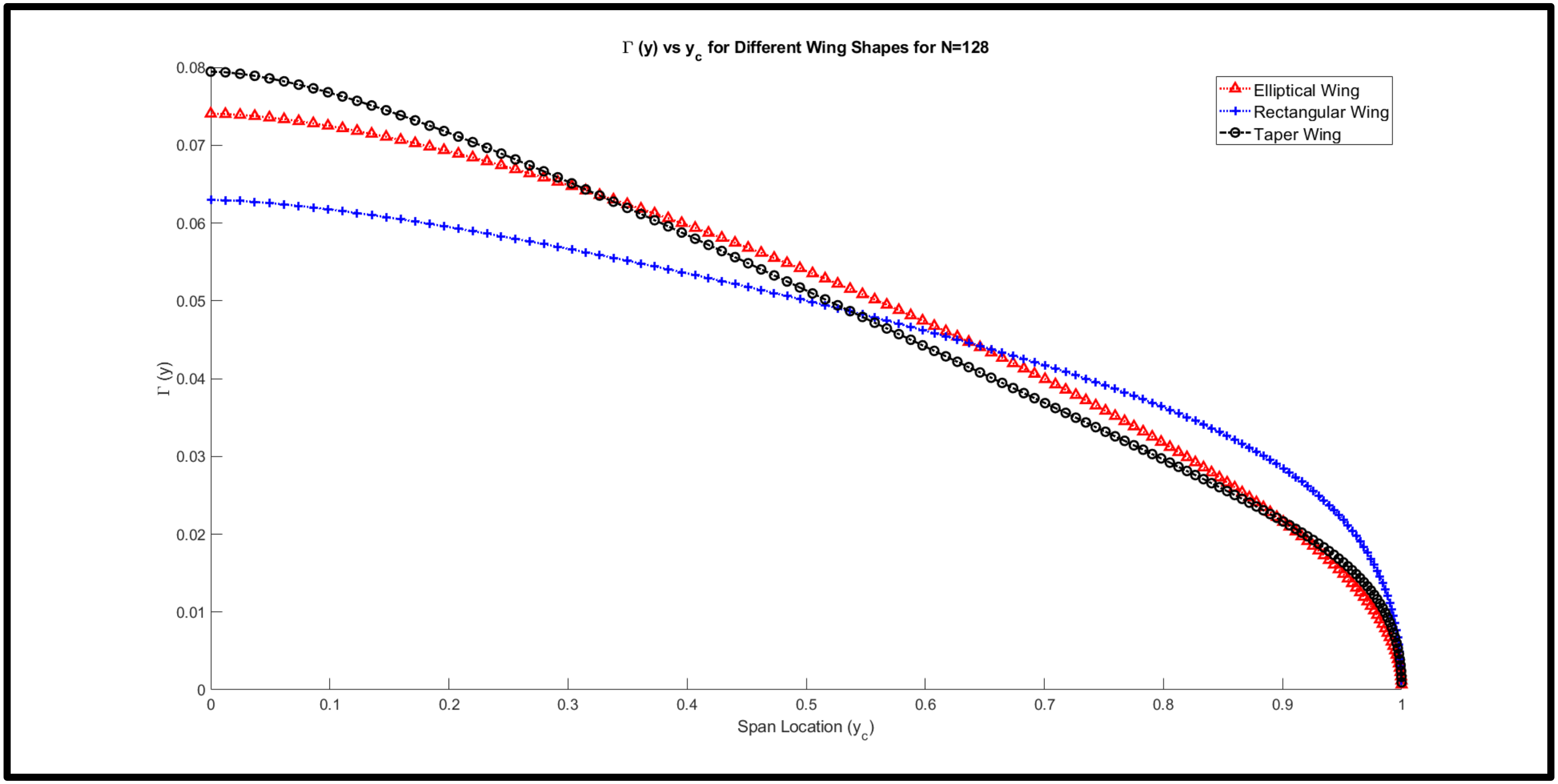


Figure 2: $\Gamma(y)$ vs y_c for Different Wing Shapes for N=128

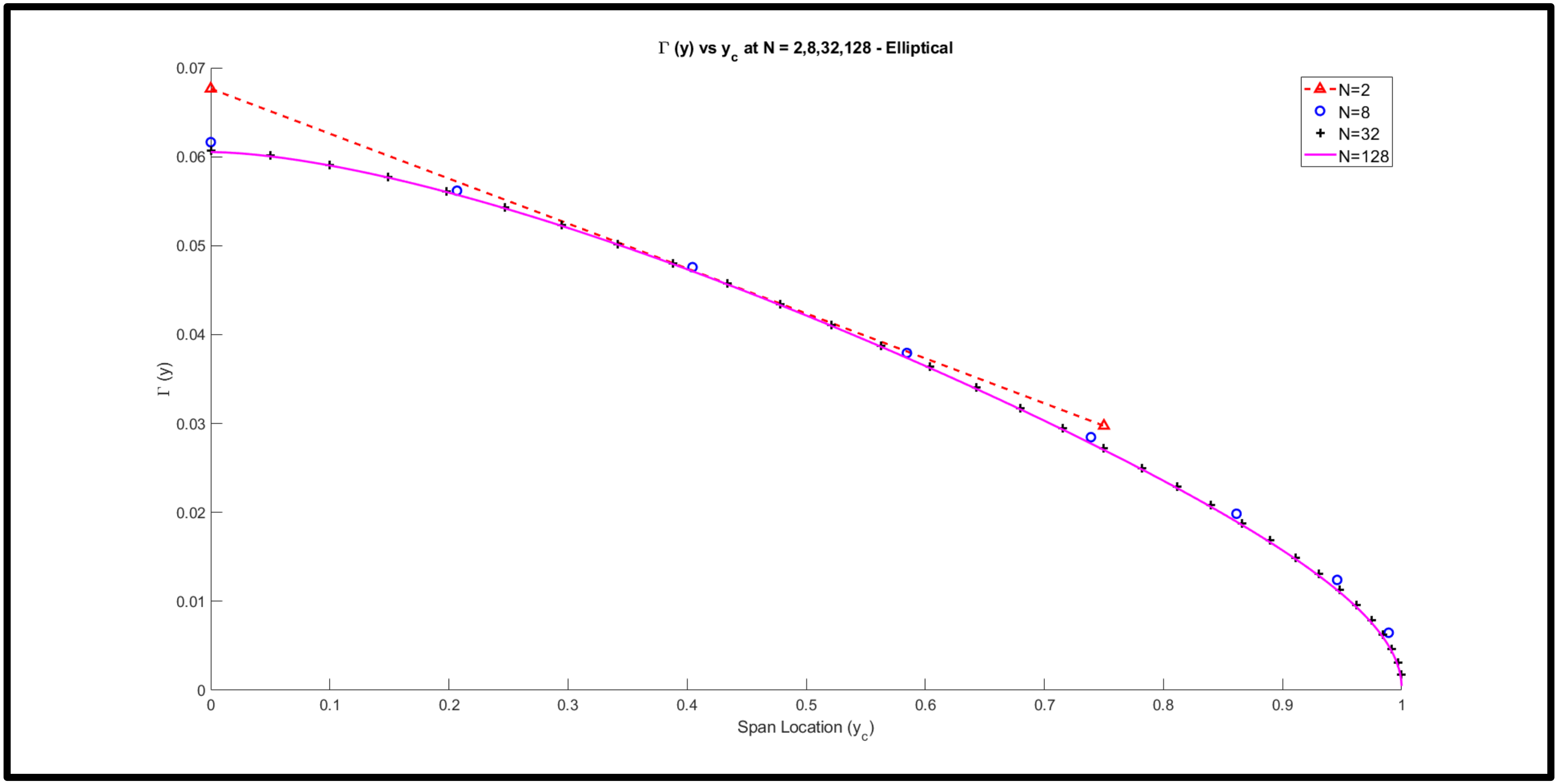


Figure 3: $\Gamma(y)$ vs y_c at N = 2, 8, 32, & 128 for an Elliptical Wing

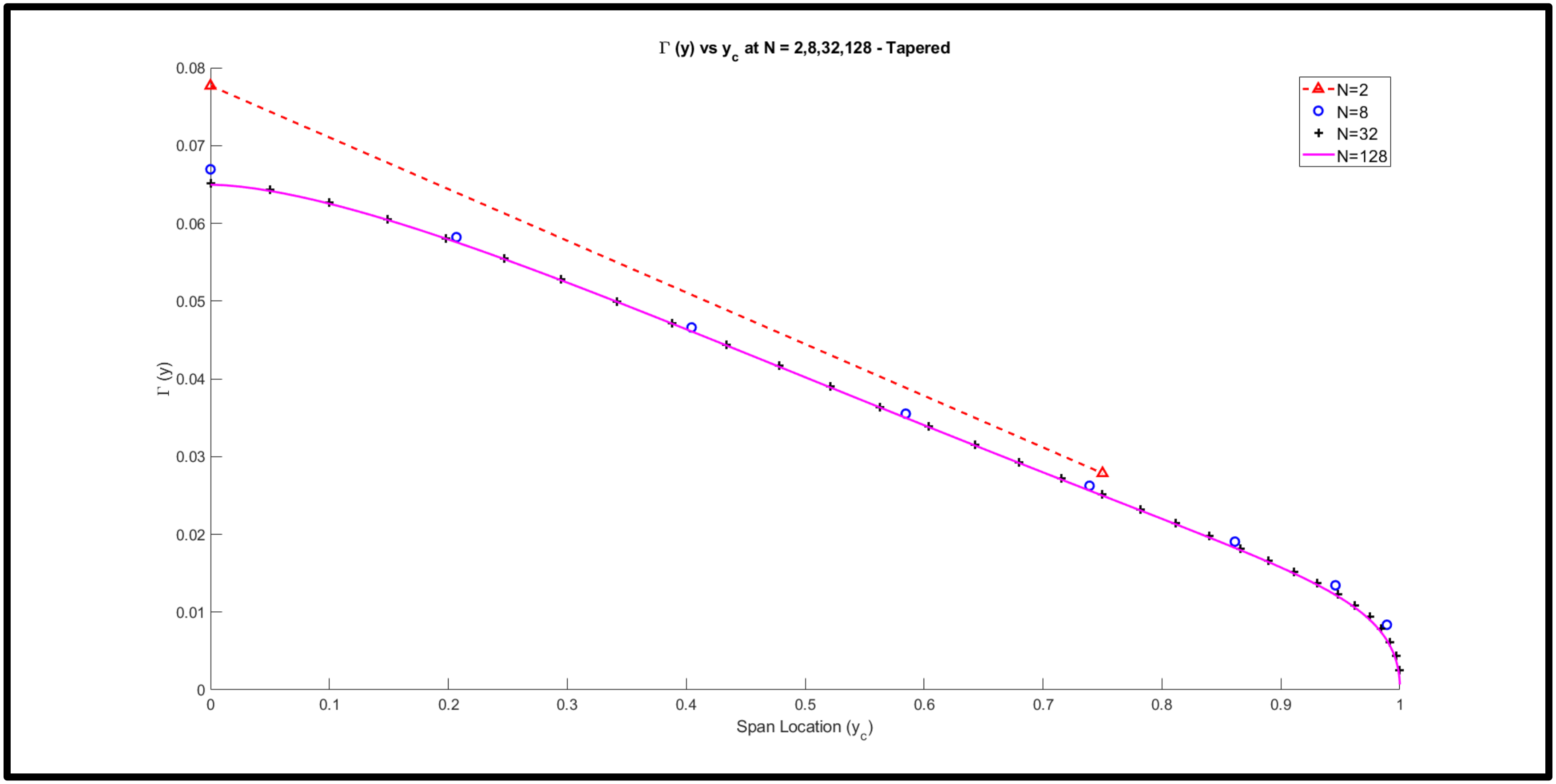


Figure 4: $\Gamma(y)$ vs y_c at N = 2, 8, 32, & 128 for a Tapered Wing

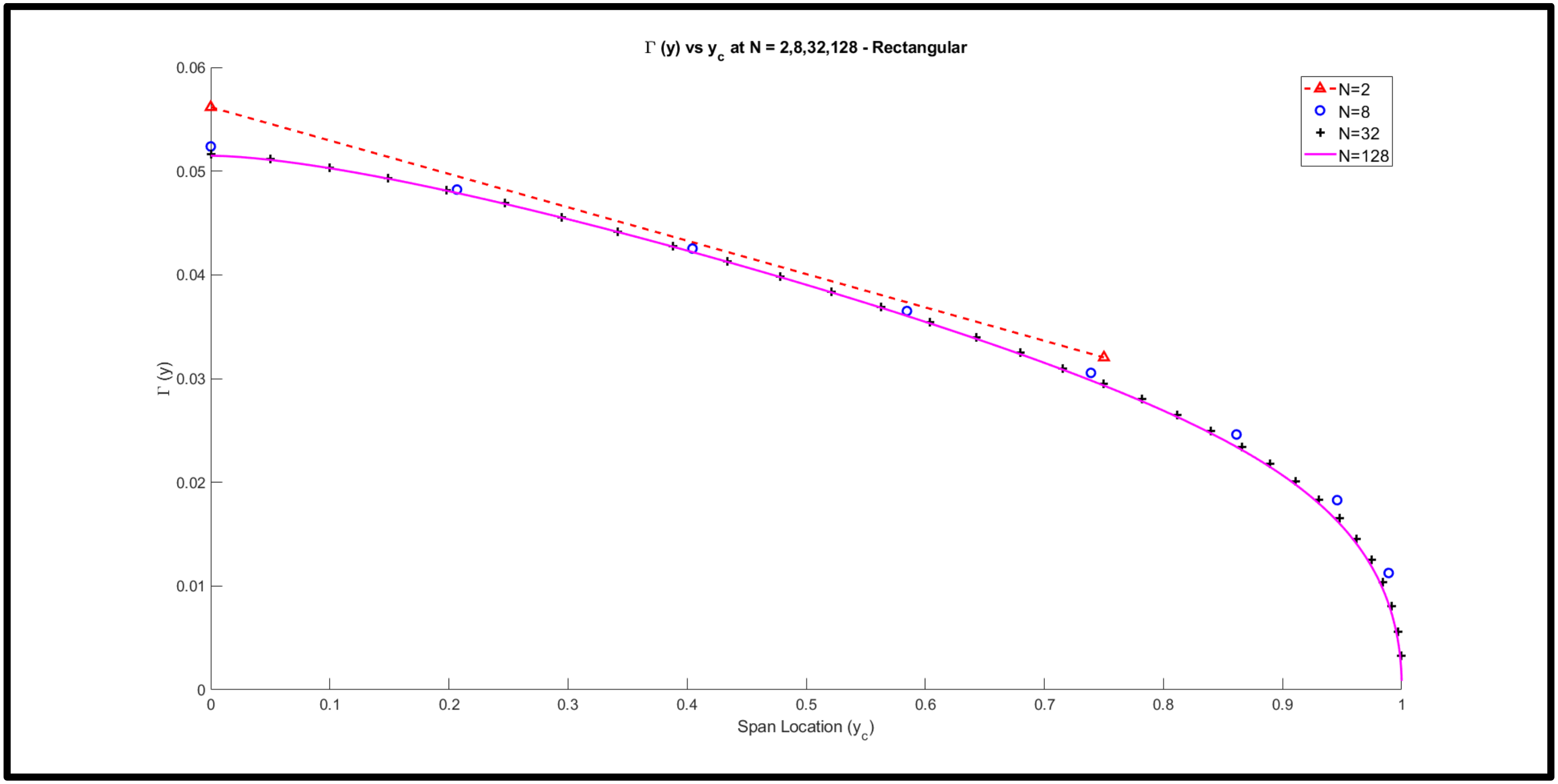


Figure 5: $\Gamma(y)$ vs y_c at N = 2, 8, 32, & 128 for a Rectangular Wing

Table 1: Spanwise distribution of bound circulation & total wing lift coefficient for all wing configurations for N = 128

			Rectangular			Tapered			Elliptical	
i	YC		Γ	Cl/CL		Γ	Cl/CL		Γ	Cl/CL
1	0.000		0.051	1.390		0.065	1.068		0.061	1.201
2	0.012		0.051	1.389		0.065	1.076		0.060	1.200
3	0.025		0.051	1.386		0.065	1.083		0.060	1.198
4	0.037		0.051	1.383		0.064	1.088		0.060	1.196
5	0.049		0.051	1.379		0.064	1.093		0.060	1.193
6	0.062		0.051	1.375		0.064	1.097		0.060	1.190
7	0.074		0.051	1.370		0.063	1.100		0.060	1.186
8	0.086		0.051	1.364		0.063	1.103		0.059	1.182
9	0.098		0.050	1.358		0.063	1.105		0.059	1.177
10	0.111		0.050	1.352		0.062	1.107		0.059	1.173
11	0.123		0.050	1.345		0.062	1.108		0.058	1.168
12	0.135		0.050	1.338		0.061	1.109		0.058	1.163
13	0.147		0.049	1.331		0.061	1.109		0.058	1.157
14	0.159		0.049	1.324		0.060	1.109		0.057	1.152
15	0.172		0.049	1.316		0.059	1.109		0.057	1.147
16	0.184		0.048	1.308		0.059	1.108		0.057	1.141
17	0.196		0.048	1.300		0.058	1.107		0.056	1.135
18	0.208		0.048	1.292		0.058	1.106		0.056	1.129
19	0.220		0.048	1.284		0.057	1.105		0.055	1.123
20	0.232		0.047	1.275		0.056	1.103		0.055	1.117
21	0.244		0.047	1.267		0.056	1.101		0.054	1.111
22	0.256		0.047	1.258		0.055	1.099		0.054	1.105
23	0.268		0.046	1.249		0.054	1.097		0.053	1.099
24	0.280		0.046	1.240		0.054	1.095		0.053	1.092
25	0.291		0.046	1.231		0.053	1.092		0.052	1.086

26	0.303		0.045	1.222		0.052	1.089		0.052	1.080
27	0.315		0.045	1.213		0.051	1.086		0.051	1.073
28	0.327		0.045	1.204		0.051	1.083		0.051	1.067
29	0.338		0.044	1.194		0.050	1.079		0.050	1.060
30	0.350		0.044	1.185		0.049	1.076		0.050	1.054
31	0.361		0.044	1.175		0.049	1.072		0.049	1.047
32	0.373		0.043	1.166		0.048	1.068		0.049	1.041
33	0.384		0.043	1.156		0.047	1.064		0.048	1.034
34	0.395		0.042	1.147		0.047	1.060		0.048	1.027
35	0.407		0.042	1.137		0.046	1.056		0.047	1.021
36	0.418		0.042	1.127		0.045	1.052		0.046	1.014
37	0.429		0.041	1.118		0.045	1.048		0.046	1.008
38	0.440		0.041	1.108		0.044	1.043		0.045	1.001
39	0.451		0.041	1.098		0.043	1.038		0.045	0.995
40	0.462		0.040	1.088		0.043	1.034		0.044	0.988
41	0.473		0.040	1.079		0.042	1.029		0.044	0.982
42	0.484		0.040	1.069		0.041	1.024		0.043	0.975
43	0.495		0.039	1.059		0.041	1.019		0.042	0.969
44	0.505		0.039	1.049		0.040	1.014		0.042	0.962
45	0.516		0.038	1.039		0.039	1.009		0.041	0.956
46	0.526		0.038	1.029		0.039	1.004		0.041	0.949
47	0.537		0.038	1.019		0.038	0.999		0.040	0.943
48	0.547		0.037	1.009		0.037	0.993		0.040	0.937
49	0.557		0.037	1.000		0.037	0.988		0.039	0.931
50	0.568		0.037	0.990		0.036	0.983		0.038	0.924
51	0.578		0.036	0.980		0.035	0.977		0.038	0.918
52	0.588		0.036	0.970		0.035	0.972		0.037	0.912
53	0.598		0.036	0.960		0.034	0.966		0.037	0.906
54	0.608		0.035	0.950		0.034	0.960		0.036	0.900

55	0.617		0.035	0.940		0.033	0.955		0.035	0.894
56	0.627		0.034	0.930		0.032	0.949		0.035	0.888
57	0.636		0.034	0.920		0.032	0.943		0.034	0.882
58	0.646		0.034	0.910		0.031	0.937		0.034	0.876
59	0.655		0.033	0.900		0.031	0.931		0.033	0.870
60	0.665		0.033	0.890		0.030	0.925		0.033	0.864
61	0.674		0.033	0.880		0.030	0.919		0.032	0.858
62	0.683		0.032	0.870		0.029	0.913		0.031	0.853
63	0.692		0.032	0.860		0.028	0.907		0.031	0.847
64	0.701		0.031	0.850		0.028	0.901		0.030	0.842
65	0.709		0.031	0.840		0.027	0.895		0.030	0.836
66	0.718		0.031	0.830		0.027	0.889		0.029	0.831
67	0.726		0.030	0.820		0.026	0.882		0.029	0.825
68	0.735		0.030	0.810		0.026	0.876		0.028	0.820
69	0.743		0.030	0.800		0.025	0.870		0.027	0.815
70	0.751		0.029	0.790		0.025	0.863		0.027	0.810
71	0.759		0.029	0.780		0.024	0.857		0.026	0.805
72	0.767		0.029	0.770		0.024	0.850		0.026	0.799
73	0.775		0.028	0.760		0.023	0.844		0.025	0.795
74	0.783		0.028	0.750		0.023	0.837		0.025	0.790
75	0.791		0.027	0.739		0.023	0.830		0.024	0.785
76	0.798		0.027	0.729		0.022	0.823		0.024	0.780
77	0.805		0.027	0.719		0.022	0.817		0.023	0.775
78	0.813		0.026	0.708		0.021	0.810		0.023	0.771
79	0.820		0.026	0.698		0.021	0.803		0.022	0.766
80	0.827		0.025	0.688		0.020	0.795		0.022	0.762
81	0.834		0.025	0.677		0.020	0.788		0.021	0.758
82	0.840		0.025	0.667		0.020	0.781		0.021	0.753
83	0.847		0.024	0.656		0.019	0.773		0.020	0.749

84	0.853		0.024	0.645		0.019	0.765		0.020	0.745
85	0.860		0.024	0.635		0.018	0.758		0.019	0.741
86	0.866		0.023	0.624		0.018	0.750		0.019	0.737
87	0.872		0.023	0.613		0.018	0.742		0.018	0.733
88	0.878		0.022	0.602		0.017	0.733		0.018	0.730
89	0.884		0.022	0.591		0.017	0.725		0.017	0.726
90	0.890		0.021	0.580		0.016	0.716		0.017	0.722
91	0.895		0.021	0.568		0.016	0.707		0.016	0.719
92	0.901		0.021	0.557		0.016	0.698		0.016	0.716
93	0.906		0.020	0.545		0.015	0.689		0.015	0.712
94	0.911		0.020	0.534		0.015	0.679		0.015	0.709
95	0.916		0.019	0.522		0.015	0.669		0.014	0.706
96	0.921		0.019	0.510		0.014	0.659		0.014	0.703
97	0.926		0.018	0.498		0.014	0.648		0.013	0.700
98	0.930		0.018	0.486		0.014	0.637		0.013	0.697
99	0.935		0.018	0.473		0.013	0.626		0.012	0.695
100	0.939		0.017	0.461		0.013	0.614		0.012	0.692
101	0.943		0.017	0.448		0.012	0.602		0.012	0.690
102	0.947		0.016	0.435		0.012	0.590		0.011	0.687
103	0.951		0.016	0.422		0.012	0.577		0.011	0.685
104	0.955		0.015	0.408		0.011	0.563		0.010	0.683
105	0.958		0.015	0.395		0.011	0.549		0.010	0.681
106	0.962		0.014	0.381		0.011	0.534		0.009	0.679
107	0.965		0.014	0.367		0.010	0.519		0.009	0.678
108	0.968		0.013	0.353		0.010	0.503		0.009	0.676
109	0.971		0.013	0.339		0.009	0.487		0.008	0.675
110	0.974		0.012	0.324		0.009	0.470		0.008	0.674
111	0.977		0.011	0.309		0.009	0.452		0.007	0.673
112	0.979		0.011	0.294		0.008	0.433		0.007	0.672

113	0.982		0.010	0.279		0.008	0.414		0.006	0.671
114	0.984		0.010	0.263		0.007	0.394		0.006	0.671
115	0.986		0.009	0.247		0.007	0.373		0.006	0.671
116	0.988		0.009	0.231		0.007	0.352		0.005	0.671
117	0.990		0.008	0.215		0.006	0.329		0.005	0.672
118	0.992		0.007	0.198		0.006	0.306		0.004	0.673
119	0.993		0.007	0.181		0.005	0.282		0.004	0.675
120	0.995		0.006	0.164		0.005	0.257		0.004	0.678
121	0.996		0.005	0.147		0.004	0.232		0.003	0.682
122	0.997		0.005	0.130		0.004	0.205		0.003	0.688
123	0.998		0.004	0.112		0.003	0.178		0.002	0.696
124	0.998		0.003	0.094		0.003	0.150		0.002	0.708
125	0.999		0.003	0.076		0.002	0.122		0.002	0.728
126	1.000		0.002	0.058		0.002	0.093		0.001	0.764
127	1.000		0.001	0.040		0.001	0.065		0.001	0.843
128	1.000		0.001	0.023		0.001	0.037		0.000	1.083

Table 2: Total wing lift coefficient (C_L), total wing induced drag coefficient (C_{Di}). & δ for a Rectangular Wing

Rectangular Wing			
N	C_L	C_Di	δ
2	0.3528	0.0036	-0.2777
8	0.3033	0.0034	-0.0600
32	0.2975	0.0035	-0.0056
128	0.2963	0.0035	0.0081

Table 3: Total wing lift coefficient (C_L), total wing induced drag coefficient (C_{Di}). & δ for a Tapered Wing

Tapered Wing			
N	C_L	C_Di	δ
2	0.4222	0.0060	-0.1508
8	0.3242	0.0044	0.0520
32	0.3174	0.0044	0.1025
128	0.3163	0.0044	0.1165

Table 4: Total wing lift coefficient (C_L), total wing induced drag coefficient (C_{Di}). & δ for a Elliptical Wing

Elliptical Wing			
N	C_L	C_Di	δ
2	0.3895	0.0047	-0.2241
8	0.3234	0.0041	-0.0065
32	0.3177	0.0042	0.0513
128	0.3167	0.0043	0.0658

APPENDIX A

Table of Contents


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                %
%                                                                %
% Justin Millsap | ARO 3011 | Computer Assignment | Dr. Tony Lin %
%                                                                %
%                                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A) Develop a computer program that uses numerical lifting-line theory to calculate the following aerodynamic characteristics.

1) Total wing lift coefficient C_L 2) Total wing induced drag coefficient C_di 3) Spanwise lift distribution normalized with total wing lift coefficient C_l (y) / C_L 4) Spandwise distribution of bound circulation GAMMA(y) 5) Delta = (pi*A*C_Di)/(C_L)^2 - 1

A.1) Rectangular Wing - Pilatus PC-6 Turbo Porter

```
clear;clc; clear all

%%%%%   INPUTS [ EDIT ] %%%%%

epsilon_t_deg = -3;                          % Twist [ deg ]
epsilon_t_rad = epsilon_t_deg*pi/180;        % Twist [ rad ]
C_L_alpha_rad = 2*pi*0.96;                   % Section lift curve slope [ 1 /
 rad ]
SemiSpanLength = 1;                         % b/2
b = SemiSpanLength*2;                       % Wing Span
AR = 8;                                      % Aspect Ratio
AOA_absolute_deg = 5;                       % Wing Absolute AOA (at center
 section) [ deg ]
AOA_absolute_rad = AOA_absolute_deg*pi/180;  % Wing Absolute AOA (at center
 section) [ rad ]
```

```
rho = 1;
V = 1;
q = (1/2)*rho*V^2;                                    % Dynamic Pressure
N = 128;                                               % Number of itterations

%%%%%% Equations [ DO NOT EDIT ]    %%%%%
% Total wing lift coefficient C_L

c = b/AR;                                            % Chord Length
S = b*c;                                             % Wing Area

i = 1:N;
j = 1:N;

%  Determining k_ij values
%  if i => j -----> k = 1
%  if i < j  -----> k =0

k = zeros(N,N);
k = k(i,j);
for i = 1:N;
    for j = 1:N;
        if i>= j;
            k(i,j) = 1;
        else i < j ;
            k(i,j) = 0;
        end
    end
end



% Cosine Spacing y_vi

for i = 1:N;
   m = 2*(N-1) + 1;
   deltaTheta = pi/m;
   y_v(i) = (b/2)*cos((1 - i + (1/2)*(m-1))*deltaTheta);
end

% Determining y_ci  Span Location of computation of downwash

 for i = 2:N;

y_c(1) = 0;
y_c(i) = (1/2)*(y_v(i) + y_v(i-1));

 end
 % y_c_128 = y_c

 % save('y_c_128.mat','y_c_128')

% Determining AOA_j
for j = 1:N
```

```
    for i = 1:N

AOA(i) = epsilon_t_rad*y_c(i) + AOA_absolute_rad;

    end
end


% Determining C_ij values
for i = 1:N;
    for j = 1:N;

C(i,j) = (1/(2*pi)) * (y_v(i)) / ( (y_v(i))^2 - (y_c(j))^2 );

    end
end

% Defining Chord Length "c"

for j = 1:N

    c(j) = b/AR ;

end

% Defining A

for i = 1:N
    for j = 1:N
        A(j,i) = (1/2)*(c(j))*(C_L_alpha_rad)*C(i,j) + k(i,j);
    end

end


% Defining B

for j = 1:N
    B(j) = (1/2)*c(j)*(C_L_alpha_rad)*AOA(j);
end
B = B' ;

% Defining gamma_lower
gamma_lower = (A)^-1 * B;


% Defining w ( down - wash velocity )
w = zeros(N,1);
for j = 1:N
    for i = 1:N

        w(j) = w(j) + C(i,j)*gamma_lower(i);

    end
```

```
end


% Defining AOA_dw ( Down-wash Angle of Attack)
AOA_dw = zeros(N , 1);
for j = 1:N;
    for i = 1:N;
        AOA_dw(j) = w(j)/V;
    end
end


% Define delta_y

delta_y = zeros(N,1);

delta_y(1) = (y_v(1)+y_v(1));

for i = 2:N
        delta_y(i) =2*( y_v(i) - y_v(i-1));
end


% Calculate lift coefficients C_Li

for j = 1:N;

    C_l(j) =  C_L_alpha_rad*(AOA(j) - AOA_dw(j));

end

% Define gamma_cap for a rectangular wing
gamma_cap_rect = zeros(N, 1);  % Initialize gamma_cap as a column vector of
 zeros

for i = 1:N
    gamma_cap_rect(i) = (1/2)*c(i)*C_l(i);
end

    gamma_cap_y_rect = gamma_cap_rect;
% Calculate Lift Coefficient
for i = 1:N;
    L(i) = rho*V*(gamma_cap_rect(i)*delta_y(i));
end

C_L_rect = sum(L)/(q*S);

%Calculate Drag Coefficient CD_i
C_D_rect = zeros(N,1);
    C_D_rect(1) = gamma_cap_rect(1)*y_v(1)*w(1);
for i = 2:N;
      C_D_rect(i) = gamma_cap_rect(i)* (y_v(i) - y_v(i-1))*w(i);
end
C_D_i_rect = AR*( C_D_rect(1) + sum(C_D_rect(2:N)) );
```

```
% Spanwise lift distribution C_ly
C_ly = sum(C_l);



C_ly_CL_rect =  C_l/ C_L_rect;



C_lyCl = C_ly/C_L_rect;

% Determine Delta
Delta_rect = ( (pi*AR*C_D_i_rect)/(C_L_rect^2) ) - 1;


% gamma_cap_y_rect_2 = gamma_cap_y_rect
% gamma_cap_y_rect_8 = gamma_cap_y_rect
% gamma_cap_y_rect_32 = gamma_cap_y_rect
% gamma_cap_y_rect_128 = gamma_cap_y_rect
% %
% %
%  save('Gamma_cap_rect_N_128.mat','gamma_cap_y_rect_128')
```

A.2) Tapered Wing - Cessna Cition Bravo

```
lambda = 0.3;

%%%%%% Equations [ DO NOT EDIT ]    %%%%%
% Total wing lift coefficient C_L

S = (b^2)/AR;


c_root = (2*S)/(b*(1+lambda));
c_tip = lambda*c_root;




i = 1:N;
j = 1:N;

%  Determining k_ij values
%  if i => j -----> k = 1
%  if i < j  -----> k =0

k = zeros(N,N);
k = k(i,j);
for i = 1:N;
    for j = 1:N;
        if i>= j;
            k(i,j) = 1;
```

```
        else i < j ;
            k(i,j) = 0;
        end
    end
end



% Cosine Spacing y_vi

for i = 1:N;
   m = 2*(N-1) + 1;
   deltaTheta = pi/m;
   y_v(i) = (b/2)*cos((1 - i + (1/2)*(m-1))*deltaTheta);
end



% Determining AOA_j
for j = 1:N
    for i = 1:N

AOA(i) = epsilon_t_rad*y_c(i) + AOA_absolute_rad;

    end
end


% Determining C_ij values
for i = 1:N;
    for j = 1:N;

C(i,j) = (1/(2*pi)) * (y_v(i)) / ( (y_v(i))^2 - (y_c(j))^2 );

    end
end

% Define chord length for a Tapered Wing
c_tapered = zeros(N, 1);
for i = 1:N;
    y = y_c(i);
    c_tapered(i) = c_root * (1 - (2 * y / b) * (1 - lambda));
end
% Defining A

for i = 1:N;
    for j = 1:N;
        A(j,i) = (1/2)*(c_tapered(j))*(C_L_alpha_rad)*C(i,j) + k(i,j);
    end

end


% Defining B
```

```
for j = 1:N
    B(j) = (1/2)*c_tapered(j)*(C_L_alpha_rad)*AOA(j);
end
B = B' ;



% Defining gamma_lower
gamma_lower = A \ B(:);




% Defining w ( down - wash velocity )
w = zeros(N,1);
for j = 1:N;
    for i = 1:N;

        w(j) = w(j) + C(i,j)*gamma_lower(i);

    end
end


% Defining AOA_dw ( Down-wash Angle of Attack)
AOA_dw = zeros(N , 1);
for j = 1:N;
    for i = 1:N;
        AOA_dw(j) = w(j)/V;
    end
end


% Define delta_y

delta_y = zeros(N,1);

delta_y(1) = (y_v(1)+y_v(1));

for i = 2:N
        delta_y(i) =2*( y_v(i) - y_v(i-1));
end





% Calculate lift coefficients C_Li

for j = 1:N;

    C_l(j) =  C_L_alpha_rad*(AOA(j) - AOA_dw(j));
```

```
end

% Define gamma_cap for a rectangular wing
gamma_cap_taper = zeros(N, 1);  % Initialize gamma_cap as a column vector of
 zeros

for i = 1:N;
    gamma_cap_taper(i) = (1/2)*c_tapered(i)*C_l(i);
end;

    gamma_cap_y_taper = gamma_cap_taper;

% Calculate Lift Coefficient
for i = 1:N;
    L(i) = rho*V*(gamma_cap_taper(i)*delta_y(i));
end

C_L_taper= sum(L)/(q*S);

%Calculate Drag Coefficient CD_i

C_D_taper = zeros(N,1);
    C_D_taper(1) = gamma_cap_taper(1)*y_v(1)*w(1);
for i = 2:N;
      C_D_rect(i) = (gamma_cap_taper(1) * y_v(1) * w(1) + gamma_cap_taper(i)*
 (y_v(i) - y_v(i-1))*w(i));
end

C_D_i_taper = AR*( C_D_taper(1) + sum(C_D_taper(2:N)) );


% Spanwise lift distribution C_ly
C_ly = sum(C_l);




C_ly_CL_taper =  C_l/ C_L_taper;




C_lyCl = C_ly/C_L_taper;

% Determine Delta
Delta_taper= ( (pi*AR*C_D_i_taper)/(C_L_taper^2) ) - 1;

% Determine gamma_cap_y
gamma_cap_y_taper = gamma_cap_taper;
%  gamma_cap_y_taper_2 = gamma_cap_y_taper
% gamma_cap_y_taper_8 = gamma_cap_y_taper
% gamma_cap_y_taper_32 = gamma_cap_y_taper
% gamma_cap_y_taper_128 = gamma_cap_y_taper
% %
%  save('Gamma_cap_taper_N_128.mat','gamma_cap_y_taper_128')
```

A.3) Elliptical Wing - Supermarine Spitfire

```
%%%%%% Equations [ DO NOT EDIT ]    %%%%
% Total wing lift coefficient C_L


S = (b^2)/AR;                                    % Wing Area

i = 1:N;
j = 1:N;

%  Determining k_ij values
%  if i => j -----> k = 1
%  if i < j  -----> k =0

k = zeros(N,N);
k = k(i,j);
for i = 1:N;
    for j = 1:N;
        if i>= j;
            k(i,j) = 1;
        else i < j ;
            k(i,j) = 0;
        end
    end
end



% Cosine Spacing y_vi

for i = 1:N;
   m = 2*(N-1) + 1;
   deltaTheta = pi/m;
   y_v(i) = (b/2)*cos((1 - i + (1/2)*(m-1))*deltaTheta);
end
% Determining y_ci  Span Location of computation of downwash

 i = 2:N;

y_c(1) = 0;
y_c(i) = (1/2)*(y_v(i) + y_v(i-1));

% y_c_32 = y_c
%
% save('y_c_32.mat','y_c_32')

%Calculating chord length for an elliptical wing


for i = 1:N;
    y = y_c(i);
    c_elliptical(i) = (4*S/(pi*b)) * sqrt(1 - (2*y/b)^2);
```

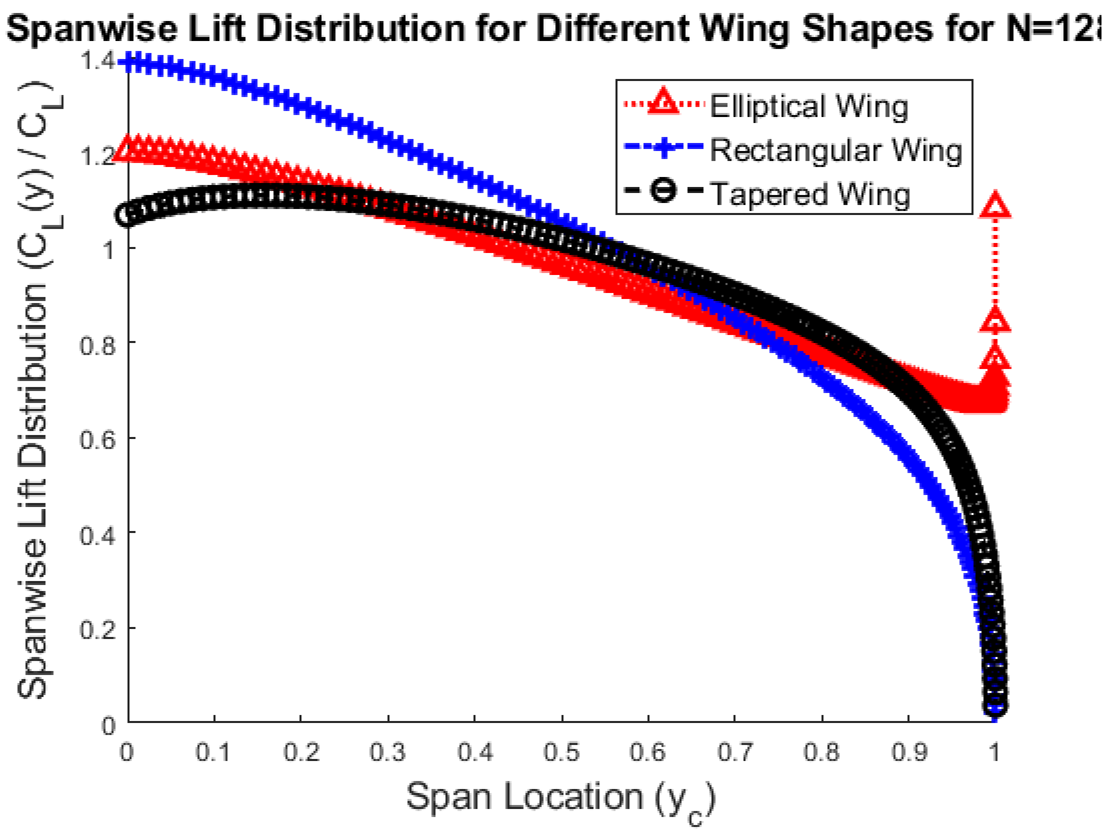
```
end



% Determining AOA_j
for j = 1:N
    for i = 1:N

AOA(i) = epsilon_t_rad*y_c(i) + AOA_absolute_rad;

    end
end




% Determining C_ij values
for i = 1:N;
    for j = 1:N;

C(i,j) = (1/(2*pi)) * (y_v(i)) / ( (y_v(i))^2 - (y_c(j))^2 );

    end
end



% Defining A

for i = 1:N
    for j = 1:N
        A(j,i) = (1/2)*(c_elliptical(j))*(C_L_alpha_rad)*C(i,j) + k(i,j);
    end

end


% Defining B

for j = 1:N
    B(j) = (1/2)*c_elliptical(j)*(C_L_alpha_rad)*AOA(j);
end
B = B' ;



% Defining gamma_lower
gamma_lower = A \ B(:);



% Defining w ( down - wash velocity )
w = zeros(N,1);
for j = 1:N;
```

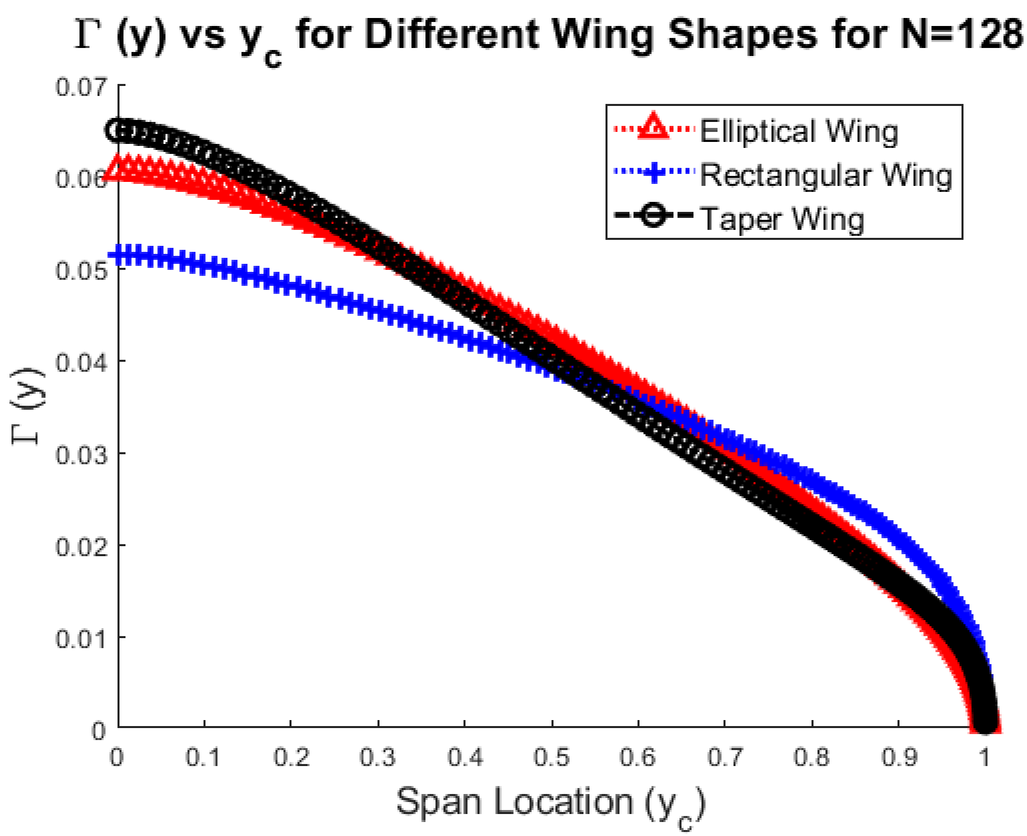
```
    for i = 1:N;

        w(j) = w(j) + C(i,j)*gamma_lower(i);

    end
end


% Defining AOA_dw ( Down-wash Angle of Attack)
AOA_dw = zeros(N , 1);
for j = 1:N;
    for i = 1:N;
        AOA_dw(j) = w(j)/V;
    end
end


% Define delta_y

delta_y = zeros(N,1);

delta_y(1) = (y_v(1)+y_v(1));

for i = 2:N
        delta_y(i) =2*( y_v(i) - y_v(i-1));
end



% Calculate lift coefficients C_Li

for j = 1:N;

    C_l(j) =  C_L_alpha_rad*(AOA(j) - AOA_dw(j));

end

% Define gamma_cap for a Elliptical Wing
gamma_cap_ellip = zeros(N, 1);  % Initialize gamma_cap as a column vector of
 zeros

for i = 1:N;
    gamma_cap_ellip(i) = (1/2)*c_elliptical(i)*C_l(i);
end

    gamma_cap_y_ellip = gamma_cap_ellip;

 % Calculate Lift Coefficient
for i = 1:N;
    L(i) = rho*V*(gamma_cap_ellip(i)*delta_y(i));
end

C_L_ellip = sum(L)/(q*S);
```

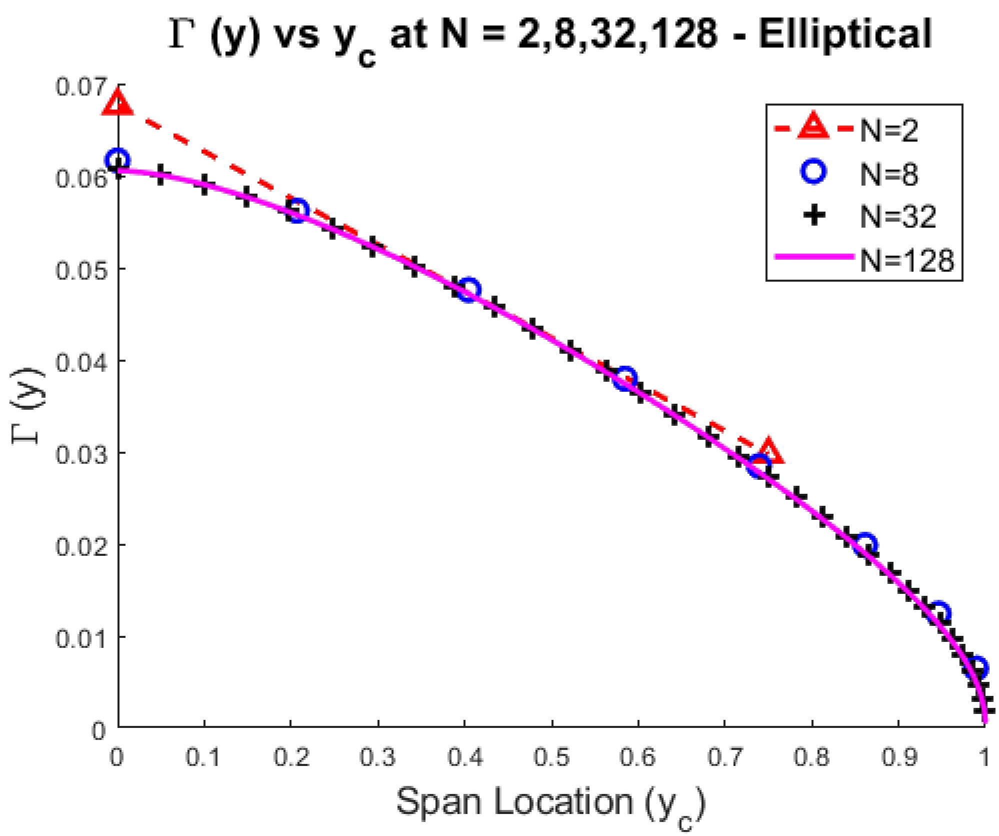
```
% Spanwise lift distribution C_ly
C_ly = sum(C_l);

%Calculate Drag Coefficient CD_i

C_D_ellip = zeros(N,1);
    C_D_ellip(1) = gamma_cap_ellip(1)*y_v(1)*w(1);
for i = 2:N;
      C_D_ellip(i) = (gamma_cap_ellip(1) * y_v(1) * w(1) + gamma_cap_ellip(i)*
 (y_v(i) - y_v(i-1))*w(i));
end
C_D_i_ellip = AR*( C_D_ellip(1) + sum(C_D_ellip(2:N)) );


C_ly_CL_ellip =  C_l/ C_L_ellip;



C_lyCl = C_ly/C_L_ellip;

% Determine Delta
Delta_elliptical = ( (pi*AR*C_D_i_ellip)/(C_L_taper^2) ) - 1;

% Determine gamma_cap_y


%  gamma_cap_y_ellip_2 = gamma_cap_y_ellip
% gamma_cap_y_ellip_8 = gamma_cap_y_ellip
% gamma_cap_y_ellip_32 = gamma_cap_y_ellip
% gamma_cap_y_ellip_128 = gamma_cap_y_ellip
%
%  save('Gamma_cap_ellip_N_128.mat','gamma_cap_y_ellip_128')


load('Gamma_cap_rect_N_2.mat','gamma_cap_y_rect_2')
load('Gamma_cap_rect_N_8.mat','gamma_cap_y_rect_8')
load('Gamma_cap_rect_N_32.mat','gamma_cap_y_rect_32')
load('Gamma_cap_rect_N_128.mat','gamma_cap_y_rect_128')

load('Gamma_cap_taper_N_2.mat','gamma_cap_y_taper_2')
load('Gamma_cap_taper_N_8.mat','gamma_cap_y_taper_8')
load('Gamma_cap_taper_N_32.mat','gamma_cap_y_taper_32')
load('Gamma_cap_taper_N_128.mat','gamma_cap_y_taper_128')

load('Gamma_cap_ellip_N_2.mat','gamma_cap_y_ellip_2')
load('Gamma_cap_ellip_N_8.mat','gamma_cap_y_ellip_8')
load('Gamma_cap_ellip_N_32.mat','gamma_cap_y_ellip_32')
load('Gamma_cap_ellip_N_128.mat','gamma_cap_y_ellip_128')

load('y_c_2.mat','y_c_2')
load('y_c_8.mat','y_c_8')
load('y_c_32.mat','y_c_32')
load('y_c_128.mat','y_c_128')
```

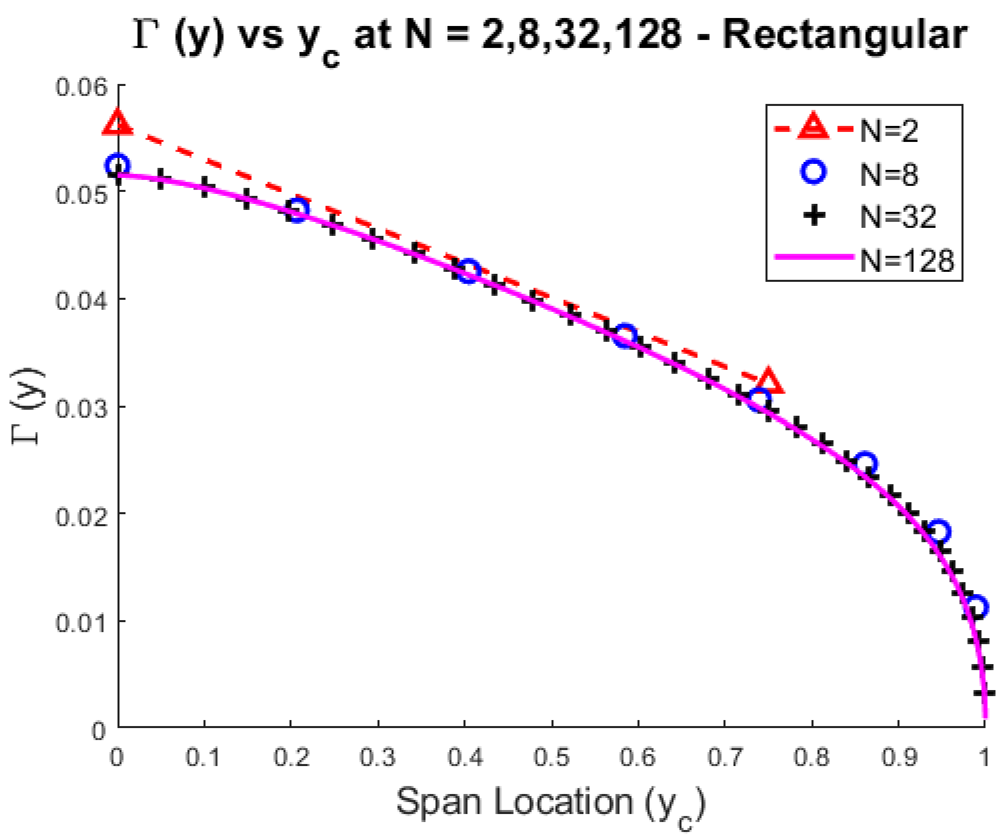
```
% Spanwise Lift distribution vs Y_c for all wing configurations

figure(1)

hold on

plot(y_c, C_ly_CL_ellip, "r:^", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "Elliptical Wing")
plot(y_c, C_ly_CL_rect, "b -.+", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "Rectangular Wing")
plot(y_c, C_ly_CL_taper, "k --o", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "Tapered Wing")
legend('show', 'FontSize', 12); % Set FontSize for legend
xlabel('Span Location (y_c)', 'FontSize', 14); % Set FontSize for xlabel
ylabel('Spanwise Lift Distribution (C_L(y) / C_L)', 'FontSize', 14); % Set
 FontSize for ylabel
title('Spanwise Lift Distribution for Different Wing Shapes for
 N=128', 'FontSize', 14); % Set FontSize for title

hold off

% Gamma(y) vs Y_c for all wing configuarions

figure(2)
hold on
plot(y_c, gamma_cap_y_ellip, "r:^", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "Elliptical Wing")
plot(y_c, gamma_cap_y_rect, "b :+", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "Rectangular Wing")
plot(y_c, gamma_cap_y_taper, "k -.o", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "Taper Wing")
legend('show', 'FontSize', 12);
xlabel('Span Location (y_c)', 'FontSize', 14);
ylabel('\Gamma (y)', 'FontSize', 14);
title('\Gamma (y) vs y_c for Different Wing Shapes for N=128', 'FontSize',
 16);

hold off

% Elliptical Wing

figure(3)
hold on

plot(y_c_2, gamma_cap_y_ellip_2, "r^--", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=2")
plot(y_c_8, gamma_cap_y_ellip_8, "bo", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=8")
plot(y_c_32, gamma_cap_y_ellip_32, "k +", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=32")
```

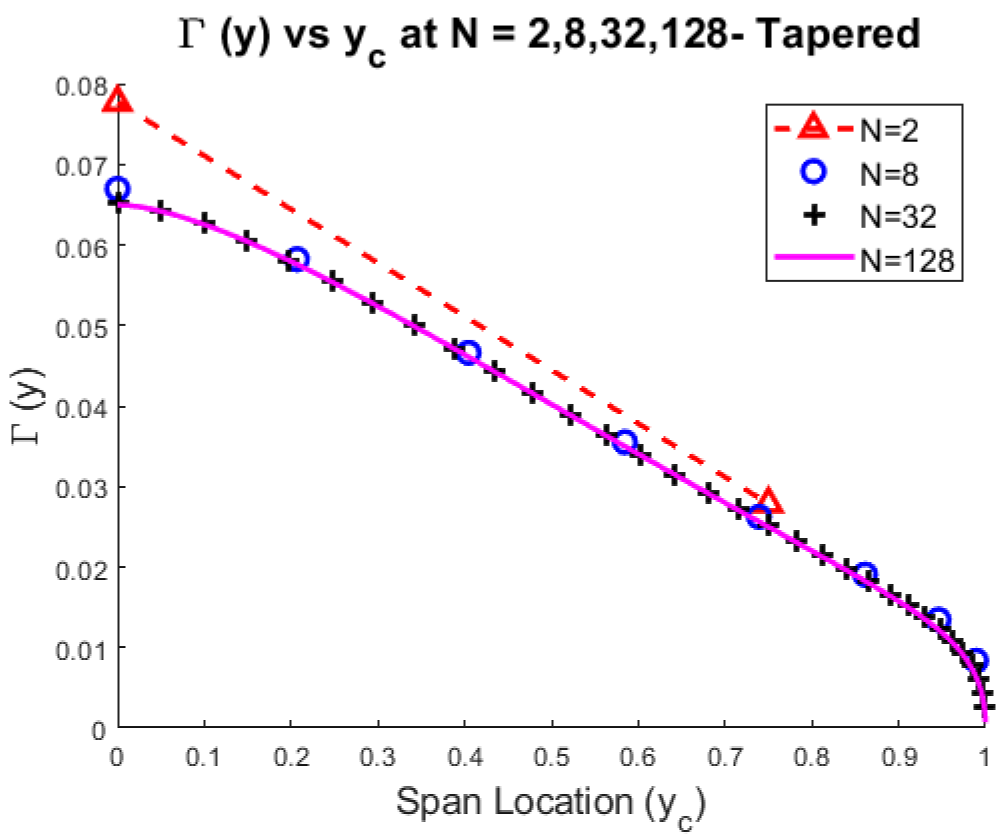
```
plot(y_c_128, gamma_cap_y_ellip_128, "m -", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=128")
xlabel('Span Location (y_c)', 'FontSize', 14);
ylabel('\Gamma (y)', 'FontSize', 14);
title('\Gamma (y) vs y_c at N = 2,8,32,128 - Elliptical', 'FontSize', 16);
legend('show', 'FontSize', 12);

hold off

% Rectangular Wing

figure(4)
hold on
plot(y_c_2, gamma_cap_y_rect_2, "r^--", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=2")
plot(y_c_8, gamma_cap_y_rect_8, "bo", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=8")
plot(y_c_32, gamma_cap_y_rect_32, "k +", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=32")
plot(y_c_128, gamma_cap_y_rect_128, "m -", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=128")
xlabel('Span Location (y_c)', 'FontSize', 14);
ylabel('\Gamma (y)', 'FontSize', 14);
title('\Gamma (y) vs y_c at N = 2,8,32,128 - Rectangular', 'FontSize', 16);
legend('show', 'FontSize', 12);

hold off

% Tapered Wing

figure(5)
hold on

plot(y_c_2, gamma_cap_y_taper_2, "r^--", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=2")
plot(y_c_8, gamma_cap_y_taper_8, "bo", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=8")
plot(y_c_32, gamma_cap_y_taper_32, "k +", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=32")
plot(y_c_128, gamma_cap_y_taper_128, "m -", "LineWidth", 2, "MarkerSize",
 8, "DisplayName", "N=128")
xlabel('Span Location (y_c)', 'FontSize', 14);
ylabel('\Gamma (y)', 'FontSize', 14);
title('\Gamma (y) vs y_c at N = 2,8,32,128- Tapered', 'FontSize', 16);
legend('show', 'FontSize', 12);

hold off
```


Spanwise Lift Distribution for Different Wing Shapes for N=128
Elliptical Wing
Rectangular Wing
Tapered Wing
Spanwise Lift Distribution (C_L(y) / C_L)
Span Location (y_c)


Γ (y) vs y_c for Different Wing Shapes for N=128
Elliptical Wing
Rectangular Wing
Taper Wing
Γ (y)
Span Location (y_c)


Γ (y) vs y_c at N = 2,8,32,128 - Elliptical
N=2
N=8
N=32
N=128
Γ (y)
Span Location (y_c)


Γ (y) vs y_c at N = 2,8,32,128 - Rectangular
N=2
N=8
N=32
N=128
Γ (y)
Span Location (y_c)



Published with MATLAB® R2021b

APPENDIX B