

## 1. Introducción

Para poder obtener un programa más optimizado y no gastar tiempo de procesamiento innecesario hicimos la convergencia de los parámetros:  $C_L, C_{Di}, C_{Mle}$ , para así saber el número de paneles óptimo que teníamos que utilizar para que el programa no tardara demasiado en realizar los cálculos pero que los resultados fueran a su vez válidos.

En la figura (1a) se ha realizado dicho análisis para un ala rectangular sin flecha ni torsión, con un perfil simétrico constante a lo largo de la envergadura y  $\alpha = 5^\circ$ . Se puede observar que con unos 100 paneles distribuidos uniformemente a lo largo de toda la envergadura, las variaciones de los coeficientes son ya muy pequeñas. Por este motivo para la realización de los siguientes apartados del trabajo se ha trabajado con 100 paneles.

Para asegurar que el código funcionaba correctamente se compararon los resultados obtenidos durante la convergencia con los presentados en (M3-3 pp. 21), obteniendo resultados satisfactorios.

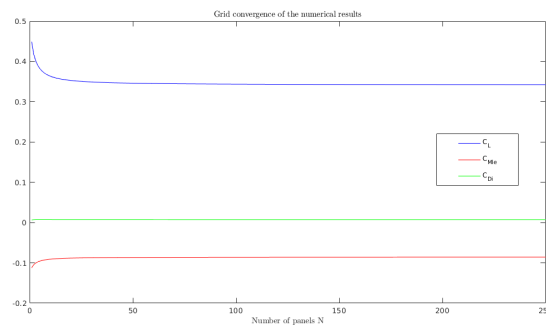


Figura 1: Convergencia del  $C_L, C_{Di}, C_{Mle}$

## 2. Apartado 1

En este primer apartado se ha analizado el comportamiento de la pendiente de sustentación frente a variaciones de alargamiento. Para la configuración alar se han tomado las indicaciones del enunciado y se ha supuesto además una distribución uniforme de perfil simétrico a lo largo de toda la envergadura.

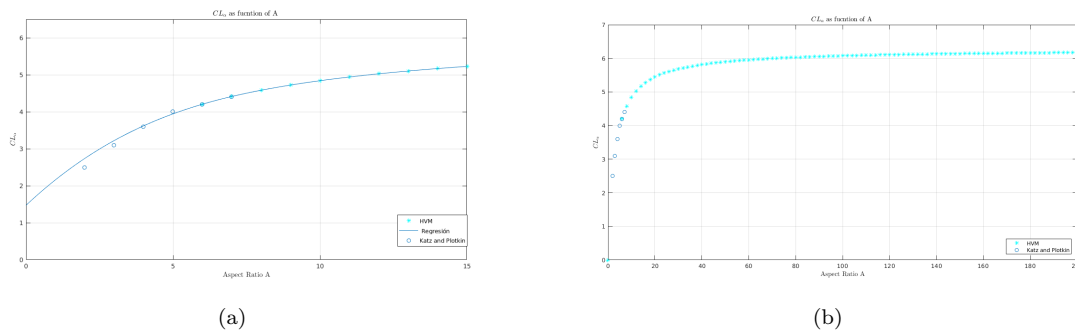


Figura 2: Comportamiento del  $C_{L\alpha}$  en función del alargamiento  $A$

En las figuras 2.a y 2.b se aprecia claramente como la pendiente de sustentación aumenta con el alargamiento. Además, en la figura (2.a) se puede ver que la diferencia entre los valores obtenidos con el HVM y los datos experimentales es poca, de esta forma los resultados obtenidos son una buena primera

aproximación para un primer estudio del ala.

Por otro lado se ha incluido también en la figura 2.b el análisis (tanto experimental como computacional) para alargamientos más pronunciados (inviabiles estructuralmente) para corroborar el comportamiento visto en clase y es que para valores de alargamiento muy altos la pendiente de sustentación  $C_{L\alpha}$  tiende a  $2\pi$  coincidiendo así con la de perfiles delgados.

### 3. Apartado 2

En este apartado se ha analizado el efecto de la flecha sobre la pendiente de sustentación y sobre la posición del centro aerodinámico. La configuración alar se basa en un ala rectangular sin torsión, con  $A=6$  y con un perfil simétrico constante a lo largo de la envergadura ( $\alpha_{l0} = 0^\circ$ ).

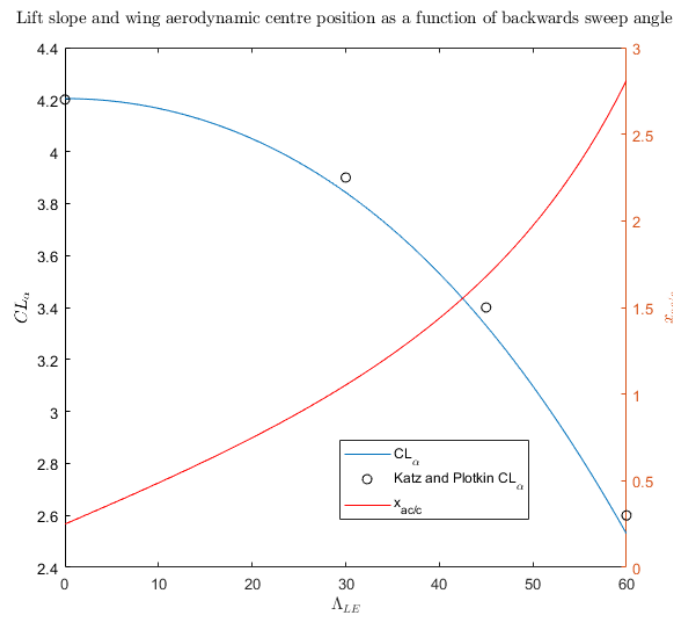
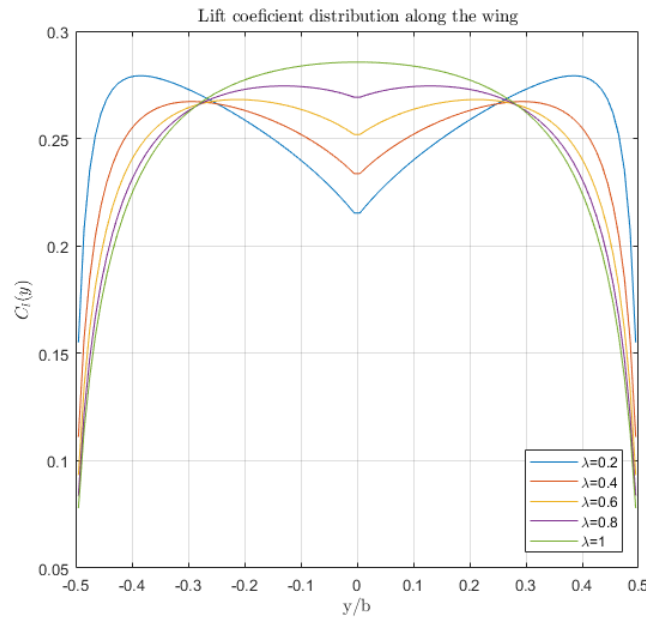


Figura 3:  $C_{L\alpha}$  y posición del centro aerodinámico en función del ángulo de flecha del borde de ataque

En la figura (3) se puede observar claramente como la pendiente de sustentación cae de forma progresiva al aumentar el ángulo de flecha del borde de ataque y como el centro aerodinámico se va retrasando al aumentar dicho parámetro. Conviene destacar que los valores calculados de  $C_{L\alpha}$  y los experimentales se aproximan bastante bien y que para un ángulo de flecha de  $0^\circ$ ,  $\frac{x_{ac}}{c}$  toma un valor de 0.25, el cual es un resultado esperado para esta configuración alar.

### 4. Apartado 3

En este apartado se ha trabajado con un ala sin torsión ni flecha, con un perfil simétrico constante a lo largo de la envergadura y con valores de estrechamiento variables. Dado que el código programado opera a partir del ángulo de ataque geométrico del ala, para conseguir la condición  $C_L = 0.25$ , se debe encontrar el ángulo de ataque alar correspondiente. Para ello, se deben obtener dos  $C_L$  distintos para cada configuración, a partir de dos ángulos de ataque conocidos. Posteriormente con estos  $C_L$  podemos obtener la pendiente de sustentación del ala y iterando con distintos ángulos de ataque encontrar el que cumpla la condición anterior. Tal y como se puede observar en la gráfica, a medida que aumentamos el *taper ratio*  $\lambda$ , el  $C_l$  de las secciones cercanas a las puntas disminuye mientras que aumenta de forma local en la zona del encastramiento.

Figura 4: Distribución de  $C_l$  a lo largo de la envergadura del ala

## 5. Apartado 4

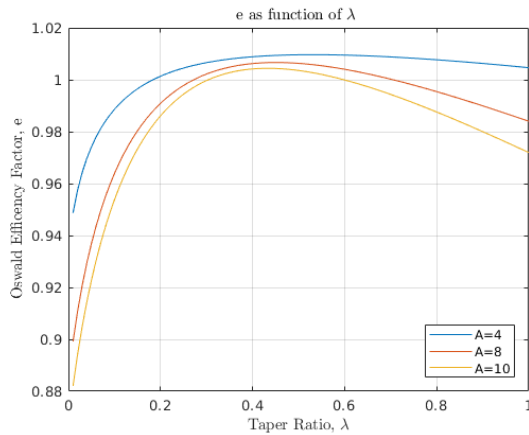
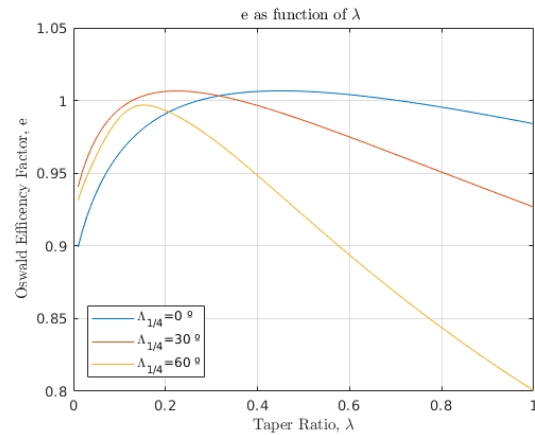
(a) Función de  $\lambda$  y  $A$ (b) Función de  $\lambda$  y  $\Lambda_{1/4}$ 

Figura 5: Factor de eficiencia de Oswald

En la figura (5a) a primera vista se puede comprobar que, para cualquier valor de *Taper Ratio*, cuanto menor es el *Aspect Ratio*, mayor es el Factor de Oswald. Así pues, vemos que para los tres casos el valor mínimo de  $e$  se alcanza cuando el *Taper Ratio* es 0 y el valor máximo se alcanza para un *Taper Ratio* de 0,4-0,5; además, a partir de estos valores, cuanto menor es el *Aspect Ratio*, el valor de  $e$  varía menos frente a aumentos de estrechamiento.

Por otra parte en la figura (5b) se puede apreciar que para un *Taper Ratio* cercano 0, el factor  $e$  tiende a aumentar con el ángulo de flecha. No obstante, esta tendencia se invierte a partir de valores de estrechamiento de 0.2 aproximadamente, de forma que cuanto menor es el ángulo de flecha, mayor es el

factor  $e$ . Se puede apreciar también que cuanto mayor es el ángulo de flecha, menor es el *Taper Ratio* para el cual se alcanza el factor  $e$  máximo.

## 6. Apartado 5

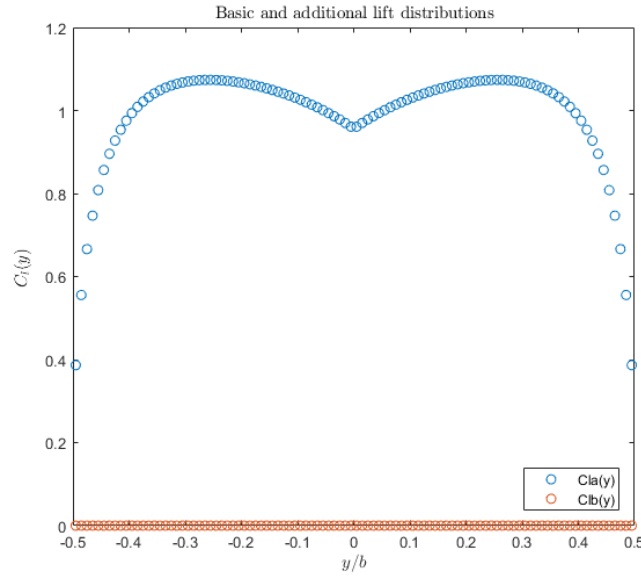


Figura 6: Distribución de sustentación básica y adicional

En primer lugar se ha calculado la distribución de sustentación adicional y básica de un ala sin torsión con un  $A = 8$ ,  $\lambda=0.5$ ,  $\Lambda_{c/2}=0^\circ$  y un perfil NACA 2408 constante; además se ha comprobado que la distribución adicional genera un  $C_L=1$  y la básica un  $C_L=0$ . De la figura (6) conviene destacar que el  $C_{lb}(y)$  es nulo a lo largo de toda la envergadura, lo que en este caso, comporta (al no haber torsión) que  $C_{Mac}$  del ala coincida con el  $C_{mac}$  del perfil. Los resultados del coeficiente de momento libre y de la posición del centro aerodinámico son los siguientes:

$$C_{M0} = -0,045 \quad (1)$$

$$\frac{X_{ac}}{c} = 0,3906 \quad (2)$$

En segundo lugar se ha estimado el  $C_{Lmax}$  y el correspondiente  $\alpha_{max}$  para dos deflexiones de flap distintas  $0^\circ$  y  $10^\circ$ . Para ello se ha supuesto la siguiente configuración: el flap se extiende desde  $\frac{y_{inicial}}{b} = -0,26$  hasta  $\frac{y_{final}}{b} = 0,26$  y el eje de charnela se encuentra situado en  $\frac{x_h}{c} = 0,8$ ; dicha posición de charnela permite aprovechar datos experimentales del libro *Theory of Wing Sections de Abbott*.

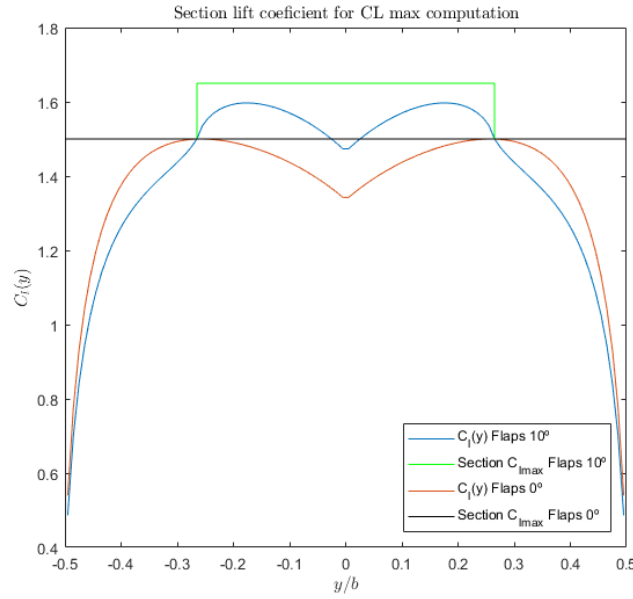


Figura 7: Distribución de  $C_l(y)$  para el  $C_{Lmax}$  del ala

Por lo que se refiere a la figura (7) destacar que el incremento de  $C_{lmax}(y)$  en las secciones afectadas por el flap cuando este presenta una deflexión de  $10^\circ$  es notorio, pasando de 1.5 a 1.65. En términos de  $C_{Lmax}$  y  $\alpha_{max}$  se han obtenido los siguientes resultados:

Deflexión de flap( $^\circ$ )	$C_{Lmax}$	$\alpha_{max}$ ( $^\circ$ )
0	1.3971	15.15
10	1.4351	12.84

Tabla 1:  $C_{Lmax}$  y  $\alpha_{max}$  para  $0^\circ$  y  $10^\circ$  de deflexión

Con la deflexión del flap, el  $C_{Lmax}$  del ala ha aumentado en torno al 3 por ciento, pero con ello y como es de esperar, el ángulo de ataque geométrico máximo se ha visto reducido de  $15.15^\circ$  a  $12.84^\circ$ . Esto se debe al desplazamiento de la curva de sustentación hacia ángulos más pequeños, que induce este elemento hipersustentador.

## 7. Apartado 6

Aprovechando parte del código del apartado anterior y con valores experimentales de la resistencia parásita para  $Re = 6M$  se ha computado la distribución de  $C_D = f(C_L)$  o curva polar. Para ello, es necesario primero calcular los coeficientes de la curva polar ( $C_{dp}$ ) del perfil NACA 2408 mediante los valores experimentales citados anteriormente.

$$C_{dp} = 0,0075C_l^2 - 0,0032C_l + 0,0055 \quad (3)$$

A continuación, realizando una discretización de la ecuación (42) pp.23 M3 – 2 y añadiendo el  $C_{Di}$  se ha obtenido la expresión de la curva polar del ala, que se muestra a continuación en la figura (8).

Se puede apreciar que la curva está ligeramente desplazada hacia la derecha de igual forma que la polar del perfil. Esto es debido a que el perfil del ala (NACA 2408) presenta una curvatura no nula. Destacar también que el  $C_{D0}$  del ala y el  $C_{d0}$  del perfil son muy parecidos.

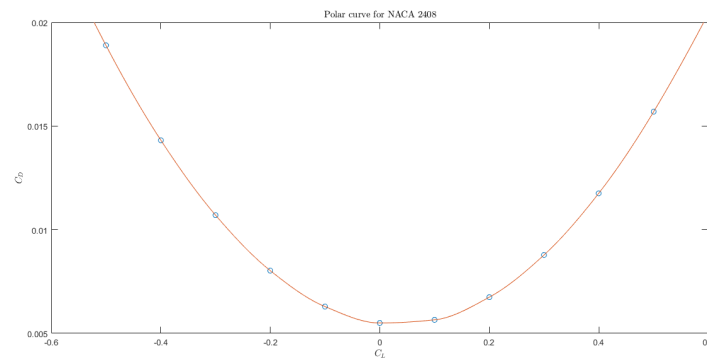


Figura 8: Curva polar del ala

## 8. Discussion

Empezando por el estrechamiento, a medida que disminuye  $\lambda$ , se reduce la superficie en las puntas del ala y por ende el momento flector, lo cual es beneficioso desde un punto de vista estructural. No obstante, conforme disminuye  $\lambda$  también lo hace el *downwash* en las puntas Figura (4) lo que comporta un aumento del  $C_l$  local en esta zona. Este aumento local provoca que entren en pérdida primero las secciones cercanas a las puntas, lo que supone la posibilidad de poder perder el control sobre los alerones (desventaja aerodinámica).

Respecto al ángulo de flecha (regresiva), cuanto mayor es dicho ángulo menor es el brazo de palanca de la sustentación y por ende menor es el momento flector (ventaja estructural). No obstante ello viene acompañado por una disminución de la pendiente de sustentación Figura (3) lo que supone una clara desventaja aerodinámica en vuelo subsónico.

## 9. Código del programa

A continuación se presenta el código base que se ha utilizado para la realización de todos los puntos del trabajo.

```

1 disp( 'WING PARAMETERS' );
2 % Aspect Ratio
3 inputwing =0;
4 while (inputwing==0)
5
6     aspect_ratio=input( 'Introduce a valid aspect ratio A: ' );
7     if ( aspect_ratio<0)
8         disp ( 'Invalid Aspect Ratio, introduce another one ' );
9         fprintf( '\n' );
10        inputwing =0;
11    elseif (aspect_ratio<5)
12        warning( 'Horshoe Vortex Method may not provide accurate results for
13                such a low aspect ratio' );
14        inputwing=1;
15    else
16        S=1/aspect_ratio;
17        inputwing=1;
18    end
19 end
20 %Taper Ratio

```

```

20 inputwing =0;
21 while (inputwing==0)
22
23     taper_ratio=input('Taper ratio \lamda: ');
24     if ( taper_ratio==0 || taper_ratio>1)
25         disp('Invalid Taper Ratio, set another one ');
26         fprintf('\n');
27         inputwing =0;
28     else
29         cr=(2/aspect_ratio)/(1+taper_ratio);
30         ct=cr*taper_ratio;
31         inputwing=1;
32     end
33 end
34 %Quarter chord swep angle (delta25)
35 delta25=input('Quarter chord swep angle (grados): ');
36 delta25=deg2rad(delta25);
37
38 disp('GEOMETRIC TWIST');
39 epsilon_ct = input('Tip airfoil epsilon (grados): ');
40 epsilon_ct= deg2rad(epsilon_ct);
41
42 disp('AERODYNAMIC CONDITIONS');
43 alphas0t = input('Zerolift tip alpha (grados): ');
44 alphas0t=-deg2rad(alphas0t);
45 alphas0r = input('Zerolift root alpha(grados): ');
46 alphas0r= -deg2rad(alphas0r);
47 yflapinicial=input('Initial flap y-position (y/b): ');
48 yflapfinal=input('Final flap y-position (y/b): ');
49 incrementalalpha0=input('Increment alpha0 due to flap: ');
50 disp('Geometric angle of attack');
51 alpha = input('alpha (grados): ');
52 alpha=deg2rad(alpha);
53 beta=deg2rad(0);
54 Vinf = [cos(alpha)*cos(beta) cos(alpha)*sin(beta) sin(alpha)];
55
56 disp('Simulation parameters ');
57 N=input('Number of panels ');
58
59 tandelta75=2*((0.75*cr)-(0.25*cr+0.5*tan(delta25)+0.5*ct));
60 y=-0.5:(1/N):0.5;
61 xvhm=zeros(1,N); %%X Vortex head midpoints
62 yvhm=zeros(1,N); %%Y Vortex head midpoints
63 for i=1:N
64     yvhm(i)=y(i)+((1/N)/2);
65     xvhm(i)=(cr/4)+(abs(yvhm(i))*tan(delta25));
66     xcp(i)=((3*cr)/4)-(abs(yvhm(i))*tandelta75);
67     ycp(i)= yvhm(i);
68     epsilon(i)=epsilon_ct*2*abs(ycp(i));
69     alphas0(i)=(2*abs(ycp(i))*(alphas0t-alphas0r))+alphas0r;
70 end
71 for i=1:N
72     yb(i)=yvhm(i)-((1/N)*0.5);
73     xb(i)=xvhm(i);
74     yc(i)=yvhm(i)+((1/N)*0.5);
75     xc(i)=xvhm(i);

```

```

76     ya(i)=yb(i);
77     xa(i)=20;
78     yd(i)=yc(i);
79     xd(i)=20;
80     chord1(i)=cr-((cr-ct)*abs(yb(i))*2);
81     chord2(i)=cr-((cr-ct)*abs(yc(i))*2);
82     s(i)=(chord1(i)+chord2(i))*(1/N)*0.5;
83 end
84 contador=1;
85 while (contador<=N)
86     if ((yb(contador))>=(yflapinicial)) && ((yc(contador))<=yflapfinal)
87         alphas0(contador)=alphas0(contador)-incrementalphas0;
88         contador=contador+1;
89     else
90         contador=contador+1;
91     end
92 end
93 %%Cálculo del vector n para cada panel (sin tener en cuenta flap)
94 for i=1:N
95     alphas0tot(i)=epsilon(i)+alphas0(i);
96     nx(i)=sin(alphas0tot(i));
97     ny(i)=0;
98     nz(i)=cos(alphas0tot(i));
99 end
100 %% Induced Velocity (Control Point)
101 for i=1:N
102     for j=1:N
103         % ab vortex
104         a=0;
105         b=0;
106         c=((xcp(i)-xa(j))*(ycp(i)-yb(j)))-((ycp(i)-ya(j))*(xcp(i)-xb(j)));
107         d=a*a+b*b+c*c;
108         r1=sqrt(((xcp(i)-xa(j))*(xcp(i)-xa(j)))+(ycp(i)-ya(j))*(ycp(i)-ya(j))));
109         r2=sqrt(((xcp(i)-xb(j))*(xcp(i)-xb(j)))+(ycp(i)-yb(j))*(ycp(i)-yb(j))));
110         ror1=(xb(j)-xa(j))*(xcp(i)-xa(j))+(yb(j)-ya(j))*(ycp(i)-ya(j));
111         ror2=(xb(j)-xa(j))*(xcp(i)-xb(j))+(yb(j)-ya(j))*(ycp(i)-yb(j));
112         com=(1/(4*pi*d))*((ror1/r1)-(ror2/r2));
113         u1(i,j)=a*com;
114         v1(i,j)=b*com;
115         w1(i,j)=c*com;
116     end
117 end
118 % bc vortex
119 for i=1:N
120     for j=1:N
121         a=0;
122         b=0;
123         c=((xcp(i)-xb(j))*(ycp(i)-yc(j)))-((ycp(i)-yb(j))*(xcp(i)-xc(j)));
124         d=a*a+b*b+c*c;
125         r1=sqrt(((xcp(i)-xb(j))*(xcp(i)-xb(j)))+(ycp(i)-yb(j))*(ycp(i)-yb(j))));
126         r2=sqrt(((xcp(i)-xc(j))*(xcp(i)-xc(j)))+(ycp(i)-yc(j))*(ycp(i)-yc(j))));
127         ror1=((xc(j)-xb(j))*(xcp(i)-xb(j)))+(yc(j)-yb(j))*(ycp(i)-yb(j));

```



```

128         ror2=((xc(j)-xb(j))*(xcp(i)-xc(j)))+(yc(j)-yb(j))*(ycp(i)-yc(j)));
129         com=(1/(4*pi*d))*((ror1/r1)-(ror2/r2));
130         u2(i,j)=a*com;
131         v2(i,j)=b*com;
132         w2(i,j)=c*com;
133     end
134 end
135 %cd vortex
136 for i=1:N
137     for j=1:N
138         a=0;
139         b=0;
140         c=((xcp(i)-xc(j))*(ycp(i)-yd(j)))-((ycp(i)-yc(j))*(xcp(i)-xd(j)));
141         d=a*a+b*b+c*c;
142         r1= sqrt(((xcp(i)-xc(j))*(xcp(i)-xc(j)))+(ycp(i)-yc(j))*(ycp(i)-yc
143             (j))));
144         r2= sqrt(((xcp(i)-xd(j))*(xcp(i)-xd(j)))+(ycp(i)-yd(j))*(ycp(i)-yd
145             (j))));
146         ror1=((xd(j)-xc(j))*(xcp(i)-xc(j)))+(yd(j)-yc(j))*(ycp(i)-yc(j));
147         ror2=((xd(j)-xc(j))*(xcp(i)-xd(j)))+(yd(j)-yc(j))*(ycp(i)-yd(j));
148         com=(1/(4*pi*d))*((ror1/r1)-(ror2/r2));
149         u3(i,j)=a*com;
150         v3(i,j)=b*com;
151         w3(i,j)=c*com;
152     end
153 end
154 % Velocidades inducidas totales en el control point
155 U=u1+u2+u3;
156 V=v1+v2+v3;
157 W=w1+w2+w3;
158 %% Induced Velocity (Vortex Head Midpoint)
159 for i=1:N
160     for j=1:N
161         % ab vortex
162         a=0;
163         b=0;
164         c=((xvhm(i)-xa(j))*(yvhm(i)-yb(j)))-((yvhm(i)-ya(j))*(xvhm(i)-xb(j)
165             ));
166         d=a*a+b*b+c*c;
167         r1= sqrt(((xvhm(i)-xa(j))*(xvhm(i)-xa(j)))+(yvhm(i)-ya(j))*(yvhm(i)
168             -ya(j))));
169         r2= sqrt(((xvhm(i)-xb(j))*(xvhm(i)-xb(j)))+(yvhm(i)-yb(j))*(yvhm(i)
170             -yb(j))));
171         ror1=(xb(j)-xa(j))*(xvhm(i)-xa(j))+(yb(j)-ya(j))*(yvhm(i)-ya(j));
172         ror2=(xb(j)-xa(j))*(xvhm(i)-xb(j))+(yb(j)-ya(j))*(yvhm(i)-yb(j));
173         com=(1/(4*pi*d))*((ror1/r1)-(ror2/r2));
174         ulm(i,j)=a*com;
175         vlm(i,j)=b*com;
176         wlm(i,j)=c*com;
177     end
178 end
179 end
180 %cd vortex
181 for i=1:N
182     for j=1:N
183         a=0;

```

```

179         b=0;
180         c=((xvhm(i)-xc(j))*(yvhm(i)-yd(j)))-((yvhm(i)-yc(j))*(xvhm(i)-xd(j)
           ));
181         d=a*a+b*b+c*c;
182         r1= sqrt(((xvhm(i)-xc(j))*(xvhm(i)-xc(j)))+(yvhm(i)-yc(j))*(yvhm(i)
           -yc(j))));
183         r2= sqrt(((xvhm(i)-xd(j))*(xvhm(i)-xd(j)))+(yvhm(i)-yd(j))*(yvhm(i)
           -yd(j))));
184         ror1=((xd(j)-xc(j))*(xvhm(i)-xc(j)))+(yd(j)-yc(j))*(yvhm(i)-yc(j))
           );
185         ror2=((xd(j)-xc(j))*(xvhm(i)-xd(j)))+(yd(j)-yc(j))*(yvhm(i)-yd(j))
           );
186         com=(1/(4*pi*d))*((ror1/r1)-(ror2/r2));
187
188         u3m(i,j)=a*com;
189         v3m(i,j)=b*com;
190         w3m(i,j)=c*com;
191
192     end
193 end
194
195 % Velocidades inducidas totales en el control point
196 Um=u1m+u3m;
197 Vm=v1m+v3m;
198 Wm=w1m+w3m;
199 %% Matriz de coeficientes
200 for i=1:N
201     for j=1:N
202         A(i,j)=U(i,j)*nx(i)+V(i,j)*ny(i)+W(i,j)*nz(i);
203     end
204     B(i)=-Vinf(1,1)*nx(i)-Vinf(1,2)*ny(i)-Vinf(1,3)*nz(i);
205 end
206
207 %% Fuerzas
208 Circulacion=linsolve(A,B);
209 cmg=(2/3)*cr*((1+taper_ratio+(taper_ratio)^2)/(1+taper_ratio));
210 for i=1:N
211     CL=(2/S)*Circulacion(i)*(1/N)+sumaCL;
212     sumaCL=CL;
213     CMle=-(2/(S*cmg))*Circulacion(i)*(xvhm(i))*(1/N)+sumaCMle;
214     sumaCMle=CMle;
215     Cly(i)=(2/s(i))*Circulacion(i)*(1/N);
216 for j=1:N
217     CDi=-(2/S)*Circulacion(i)*(1/N)*Circulacion(j)*Wm(i,j)+sumaCDi;
218     sumaCDi=CDi;
219 end
220 end

```