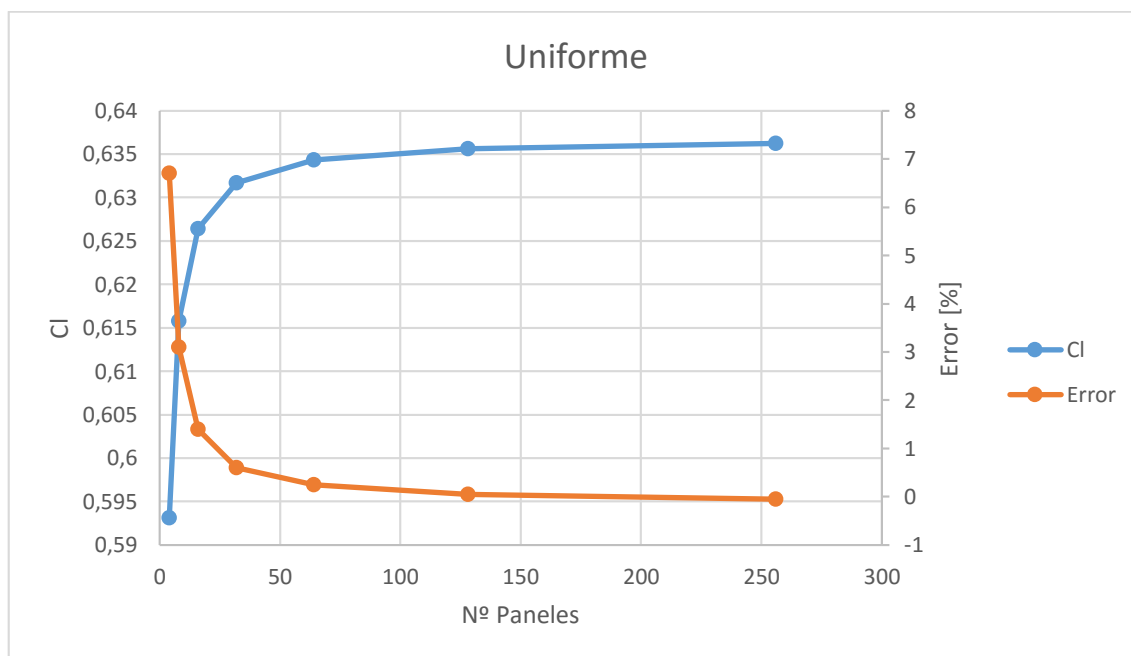


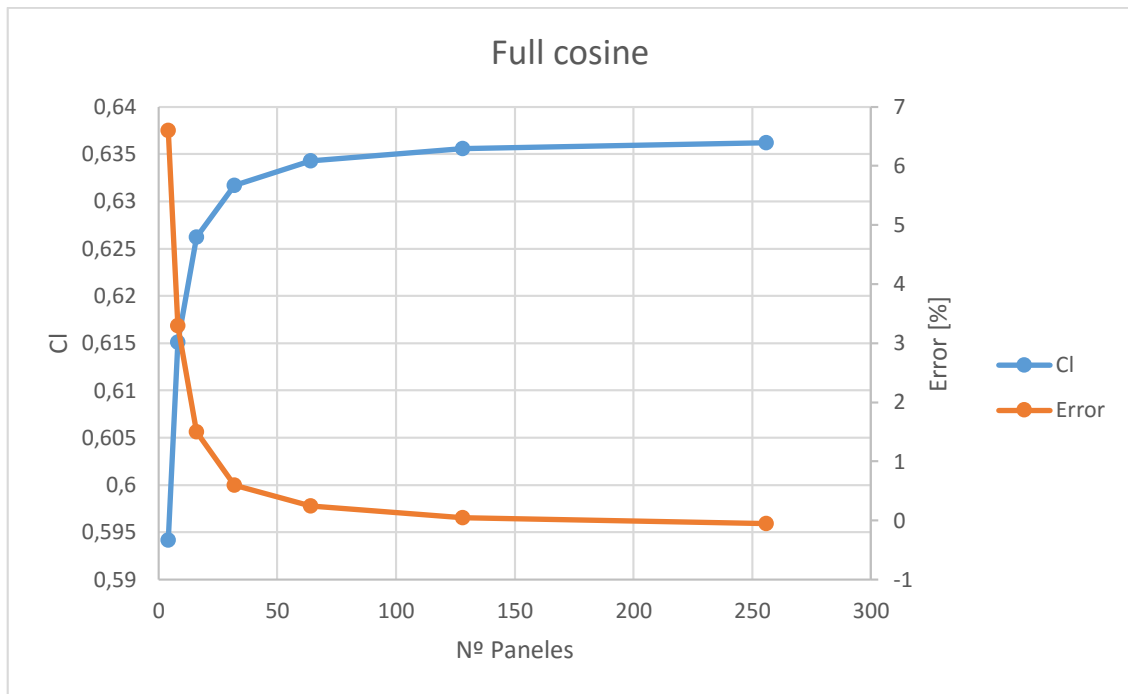
1- Verificación

Usando la teoría de perfiles delgados para el perfil NACA 2212 y un ángulo de ataque de 4° , se obtienen los coeficientes A_0 y A_1 , que permiten el cálculo del coeficiente de sustentación, el cual da $C_l = 0.6359$. Tras ello, usamos el código creado para obtener los valores del C_l variando la cantidad de paneles y aplicando distribuciones uniforme y “full cosine”, los cuales compararemos con el valor teórico con tal de determinar el error.

Nº paneles	C_l (uniforme)	C_l (full cosine)	Error [%] (unif)	Error [%] (f.cos)
4	0.5931	0.5942	6.7	6.6
8	0.6158	0.6151	3.1	3.3
16	0.6264	0.6262	1.4	1.5
32	0.6317	0.6317	0.6	0.6
64	0.6343	0.6343	0.25	0.25
128	0.6356	0.6356	0.05	0.05
256	0.6362	0.6362	-0.05	-0.05

A partir de estos datos conseguimos las curvas del análisis de convergencia de paneles del C_l y del error en función del número de paneles en los casos de distribución uniforme y de full cosine.

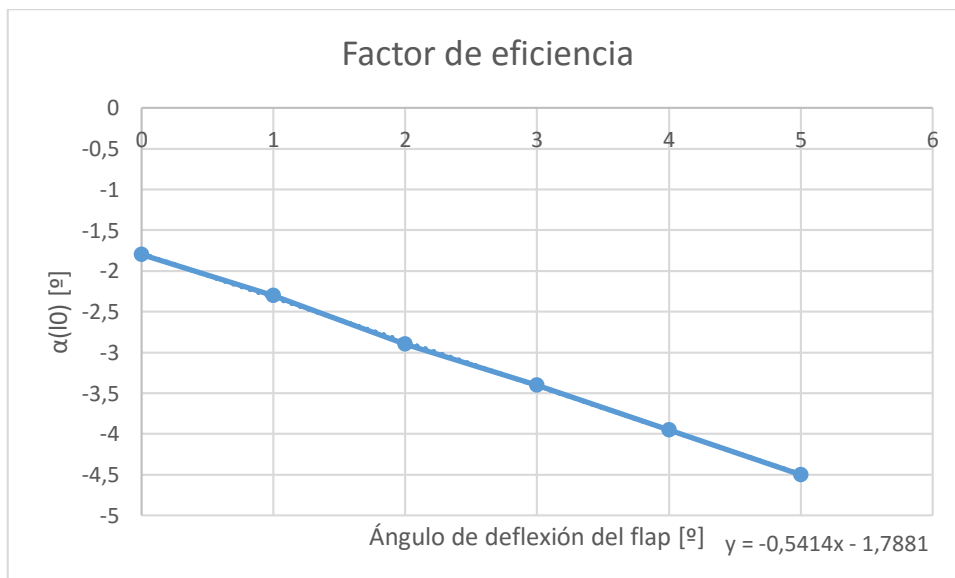




Se obtienen errores muy pequeños a partir de los 32 paneles, a partir de los cuales, ambos métodos de distribución (uniforme y full cosine) dan exactamente el mismo resultado. El error debería converger hacia el 0, pero pasa a ser negativo y converger hacia -0.14% (se ha probado con 5000 paneles).

En cuanto a la configuración del flap, optaremos por ponerlo en $x=0.8$, y utilizaremos 99 paneles (ya que necesitamos 100 puntos para el análisis y que así, al multiplicarlos por la posición del flap, obtengamos un número entero).

Variando η y corrigiendo α_{i0} para que el Cl sea 0, obtenemos la siguiente gráfica, la pendiente de la cual es el factor de eficiencia $\frac{\partial \alpha}{\partial \eta} = -0.5414$

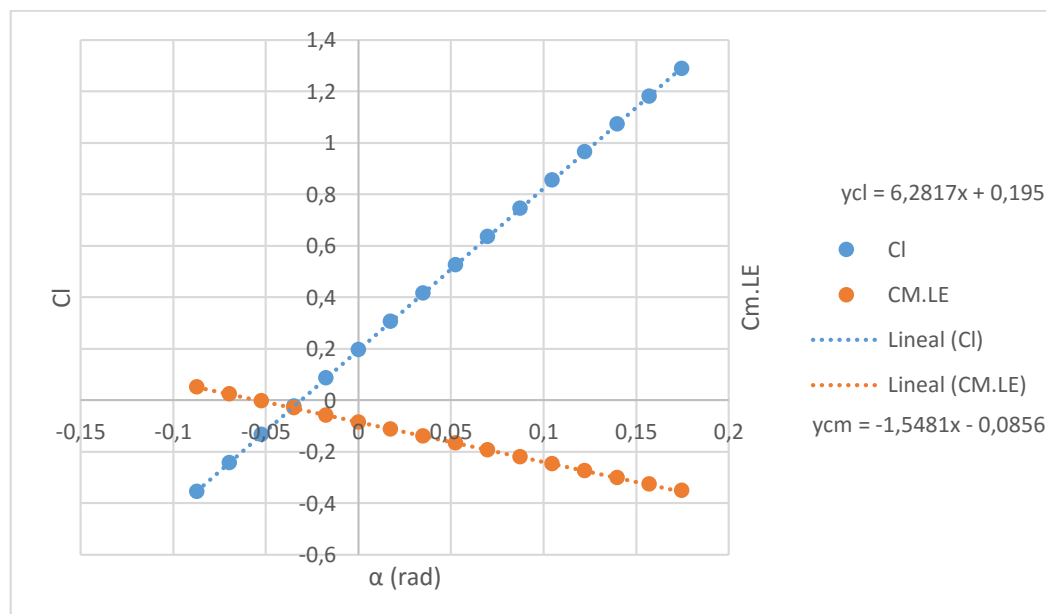


2- Validación

A continuación se calcularán la pendiente de la sustentación (Cl_α), el ángulo de sustentación nula (α_{i0}) y el coeficiente de momento con respecto al centro aerodinámico (Cm_0).

Obtendremos las pendientes ejecutando el código en el intervalo de ángulos de ataque de -5 a 10 grados

α (°)	α (rad)	Cl	Cm.LE
-5	0.08726646	-0.3549	0.0511
-4	0.06981317	-0.2449	0.0237
-3	0.05235988	-0.1348	-0.0037
-2	0.03490659	-0.0246	-0.0312
-1	0.01745329	0.0855	-0.0587
0	0	0.1956	-0.0862
1	0.01745329	0.3057	-0.1136
2	0.03490659	0.4157	-0.141
3	0.05235988	0.5255	-0.1682
4	0.06981317	0.6352	-0.1953
5	0.08726646	0.7447	-0.2222
6	0.10471976	0.854	-0.2488
7	0.12217305	0.963	-0.2753
8	0.13962634	1.0717	-0.3014
9	0.15707963	1.1801	-0.3272
10	0.17453293	1.2881	-0.3527



Resultados extraídos de la gráfica:

Cl_α	α_{l0} [rad]	C_{m0}
6.2817	-0.03	-0.05

Se comprueba que la pendiente de sustentación es cercana a 2π , y el cociente de la pendiente del coeficiente de momentos con respecto al borde de ataque y el Cl es 0.25 (Un cuarto de la cuerda, donde se halla el centro aerodinámico), por lo que se cumple la teoría de perfiles delgados.

Resultados experimentales (NACA report 460):

Cl_α	α_{l0} [°]	C_{m0}
5.17	-0.035	-0.03

Los resultados obtenidos distan bastante de los datos experimentales del informe NACA debido a las simplificaciones que hemos realizado, pese a que los datos obtenidos de las gráficas oficiales pueden no ser demasiado exactos (nitidez de las gráficas deja bastante que desear), podemos concluir que el desvío cometido es notable en Cl_α , mientras que en las demás medidas puede no serlo tanto.

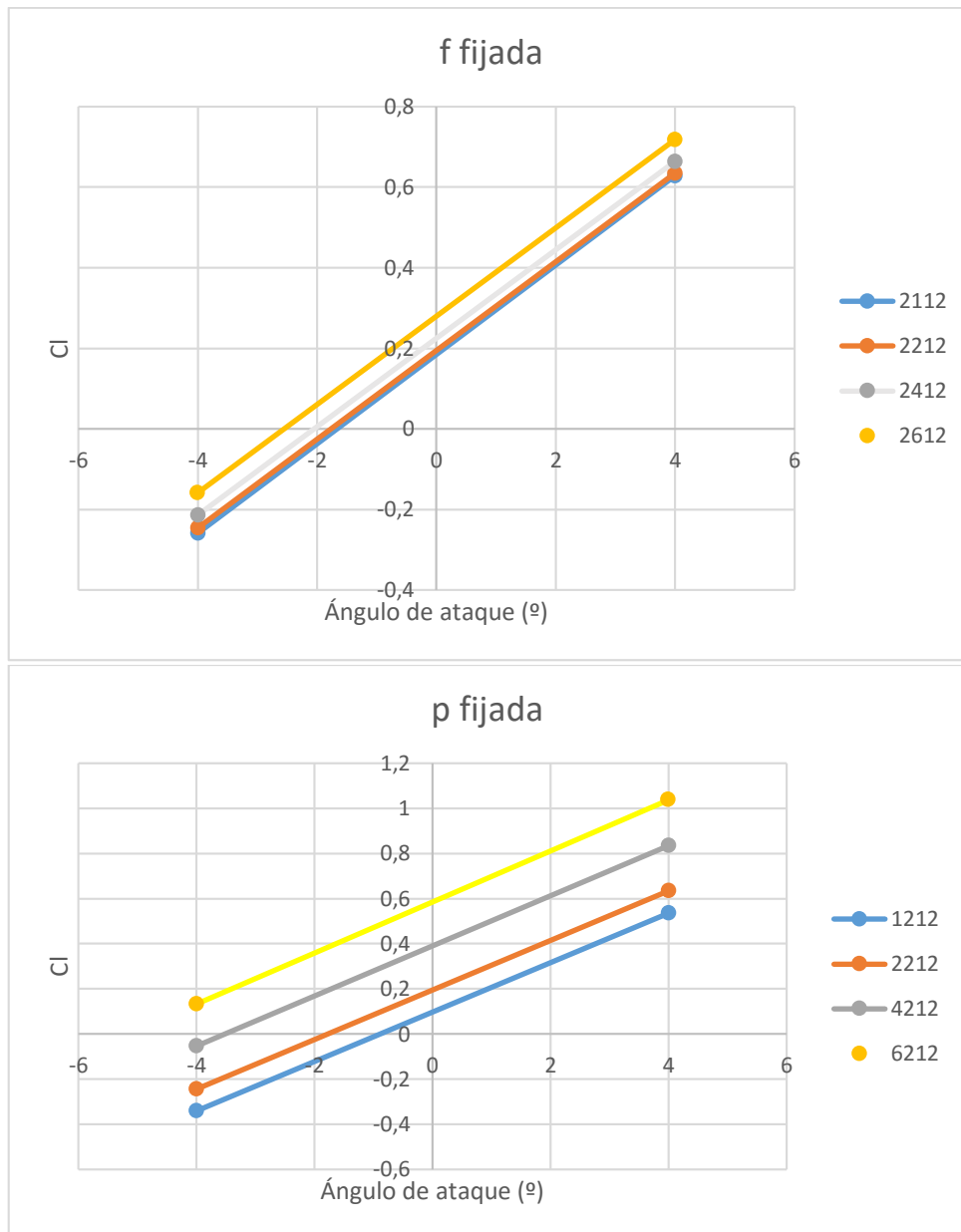
Tabla de errores:

Cl_α	α_{l0} [°]	C_{m0}
21.5%	14.3%	66.6%

3- Discusión de las variaciones de características aerodinámicas dadas por la variación del perfil.

Obtendremos los datos necesarios usando dos puntos para cada perfil

	Perfil	Cl		Cm	
		-4	4	-4	4
Fijar f	2112	-0.2585	0.6287	0.0342	-0.1854
	2212	-0.2449	0.6352	0.0237	-0.1953
	2412	-0.2135	0.6638	0	-0.2184
	2612	-0.1582	0.7177	-0.0349	-0.2531
Fijar p	1212	-0.3411	0.5363	0.0664	-0.1523
	2212	-0.2449	0.6352	0.0237	-0.1953
	4212	-0.0548	0.8352	-0.0612	-0.2813
	6212	0.1331	1.0379	-0.1454	-0.3671



Podemos observar que, en ambos casos, cuando aumentamos el camber máximo o su posición la recta se desplaza hacia arriba. Cuando aumentamos la combadura máxima aparece un aumento de C_l debido al aumento del pico de succión que genera una mayor diferencia de presiones entre el intradós y el extradós, aumentando la sustentación. Cuando retrasamos la posición de máxima combadura el desplazamiento de la recta es menor que cuando aumentamos el camber máximo.

Podemos observar que el α_{l0} disminuye cuando retrasamos y aumentamos la combadura, mientras que el C_{m0} , aumenta linealmente con signo negativo, por lo tanto, se está desestabilizando el perfil. Cuando aumentamos f , el aumento del momento se debe al aumento del módulo de la sustentación, mientras que cuando aumentamos p , el aumento del momento se debe a un mayor brazo de palanca.

Código (matlab)

```
naca=2212; % Introducción del perfil NACA 4-dígitos
f=floor((naca/1000))/100; % Camber máximo (centésimas)
p=floor(((naca/1000-floor(naca/1000))*10))/10; % Posición de camber máxima (décimas)
t=((naca/1000-floor(naca/1000))*10-floor((naca/1000-floor(naca/1000))*10))*100; % Espesor máximo
alpha_grade=-4.5; % Ángulo de ataque [°]
alpha=((2*pi)/360)*alpha_grade; % Ángulo de ataque [rad]
npanels=99; % Número de paneles
distrib=0; % Distribución de paneles. Valor 0 --> Uniforme Valor 1 --> Full cosine
flap=0; % Activar flap (1= activo)
i=1;
j=1;
N=npanels+1;
x=[]; % Inicialización x puntos de cambio de placa
z=[]; % Inicialización z puntos de cambio de placa
x_v=[]; % Inicialización x torbellinos
x_cp=[]; % Inicialización x collocation point
z_cp=[]; % Inicialización z collocation point
z_v=[]; % Inicialización z torbellinos
panellength=[]; % Inicialización longitud de los paneles cos
nx=[]; % Componente x vector normal panel
nz=[]; % Componente z vector normal panel

if distrib==1
    while i<=N % x cos puntos de cambio de placa
        x(i)=1/2*(1-cos((i-1)/(N-1)*(pi)));
        i=i+1;
    end
    i=1;
    while i<=npanels % x cos torbellinos
        x_v(i)=(x(i+1)-x(i))*0.25+x(i);
        i=i+1;
    end
    i=1;
    j=1;
    while j<=npanels % x cos collocation point
        x_cp(j)=(x(j+1)-x(j))*0.75+x(j);
        j=j+1;
    end
    while i<=npanels && x(i)<=p % z cos cambio de panel
        z(i)=(f/(p^2))*((2*p*x(i))-(x(i)^2));
        i=i+1;
    end
    while i<=N
        z(i)=(f/((1-p)^2))*(1-(2*p)+(2*p*x(i))-(x(i)^2));
        i=i+1;
    end
    i=1;
    while i<=npanels && x_v(i)<=p % z cos torbellinos
        z_v(i)=(f/(p^2))*((2*p*x_v(i))-(x_v(i)^2));
        i=i+1;
    end
    while i<=npanels
        z_v(i)=(f/((1-p)^2))*(1-(2*p)+(2*p*x_v(i))-(x_v(i)^2));
        i=i+1;
    end
    i=1;
    j=1;
    while j<=npanels && x_cp(j)<=p % z cos collocation point
        z_cp(j)=(f/(p^2))*((2*p*x_cp(j))-(x_cp(j)^2));
        j=j+1;
    end
    while j<=npanels
        z_cp(j)=(f/((1-p)^2))*(1-(2*p)+(2*p*x_cp(j))-(x_cp(j)^2));
```

```

        j=j+1;
    end
    while i<=npanels % longitud de los paneles cos
        panellength(i)=sqrt((x(i+1)-x(i))^2+(z(i+1)-z(i))^2);
        i=i+1;
    end
    i=1;
    while i<=npanels % vector normal panel cos
        nx(i)=(-(z(i+1)-z(i)))/panellength(i);
        nz(i)=(x(i+1)-x(i))/panellength(i);
        i=i+1;
    end
else if distrib==0
    x=0:(1/npanels):1; % x uniforme puntos de cambio de placa
    x_v=(0.25*1/npanels):(1/npanels):1; % x uniforme torbellinos
    x_cp=(0.75*1/npanels):(1/npanels):1; % x uniforme collocation point
    while i<=npanels && x(i)<=p % z uniform puntos cambio de panel
x<=p...
        z(i)=(f/(p^2))*((2*p*x(i))-(x(i)^2));
        i=i+1;
    end
    if flap==1 % Parte del flap asegurar p<flap_position
        while i<=N*0.8
            z(i)=(f/((1-p)^2))*(1-(2*p)+(2*p*x(i))-(x(i)^2));
            i=i+1;
        end
        % ...x>p
        for i=N*0.8:N
            z(i)=(f/((1-p)^2))*(1-(2*p)+(2*p*x(i))-(x(i)^2))-tand(5)*(x(i)-0.8); %
z flap
        end % ...x>p
        else
            while i<=N
                z(i)=(f/((1-p)^2))*(1-(2*p)+(2*p*x(i))-(x(i)^2));
                i=i+1;
            end % ...x>p
        end
        i=1;
        while i<=npanels && x_v(i)<=p % z uniform torbellinos...
            z_v(i)=(f/(p^2))*((2*p*x_v(i))-(x_v(i)^2));
            i=i+1;
        end
        while i<=npanels
            z_v(i)=(f/((1-p)^2))*(1-(2*p)+(2*p*x_v(i))-(x_v(i)^2));
            i=i+1;
        end%...x>p
        j=1;
        i=1;
        while j<=npanels && x_cp(j)<=p % z uniform collocation point
            z_cp(j)=(f/(p^2))*((2*p*x_cp(j))-(x_cp(j)^2));
            j=j+1;
        end
        while j<=npanels
            z_cp(j)=(f/((1-p)^2))*(1-(2*p)+(2*p*x_cp(j))-(x_cp(j)^2));
            j=j+1;
        end
        while i<=npanels % longitud de los paneles uniforme
            panellength(i)=sqrt((x(i+1)-x(i))^2+(z(i+1)-z(i))^2);
            i=i+1;
        end
        i=1;
        while i<=npanels % vector normal panel uniforme
            nx(i)=(-(z(i+1)-z(i)))/panellength(i);
            nz(i)=(x(i+1)-x(i))/panellength(i);
            i=i+1;
        end
    end
end
end

```

```

u=[];
w=[];
i=1;
while i<=npanels % Componentes de la velocidad u,w
    j=1;
    while j<=npanels
        u(i,j)=(1/(2*pi))*(z_cp(i)-z_v(j))/(((x_cp(i)-x_v(j))^2)+((z_cp(i)-
z_v(j))^2));
        w(i,j)=-(1/(2*pi))*(x_cp(i)-x_v(j))/(((x_cp(i)-x_v(j))^2)+((z_cp(i)-
z_v(j))^2));
        j=j+1;
    end
    i=i+1;
end
rhs=[];
i=1;
while i<=npanels
    rhs(i)=-(cos(alpha)*nx(i)+sin(alpha)*nz(i)); %Sustitución cos=1
sin=alpha
    i=i+1;
end % rhs
A=[];
i=1;
while i<=npanels
    j=1;
    while j<=npanels
        A(i,j)=u(i,j)*nx(j)+w(i,j)*nz(j);
        j=j+1;
    end
    i=i+1;
end % Aij
circulation=[];
circulation=inv(A)*transpose(rhs);
cl=0;
i=1;

while i<=npanels % circulation
    cl=cl+2*circulation(i);
    i=i+1;
end
cm_le=0;
i=1;
while i<=npanels % cm_le
    cm_le=cm_le+(-2*circulation(i)*x_v(i)*cos(alpha));
    i=i+1;
end
dCp=[];
i=1;
while i<=npanels % delta cp
    dCp(i)=2*circulation(i)*(1/panellength(i));
    i=i+1;
end

```