

ANÁLISIS Y COMPUTACIÓN DE LOS PARÁMETROS AERODINÁMICOS DE PERFILES 4-DIGIT NACA CON FLAPS USANDO EL MÉTODO DE VÓRTICE DISCRETO (DVM)

Yi Qiang Ji, Iván Sermanoukian, Santiago Villarroya,
Cristian Asensio, Álvaro Sánchez

Ingeniería en Tecnologías Aeroespaciales
ESEIAAT

Departamento de Ingeniería Aeronáutica y Aerodinámica
Universitat Politècnica de Catalunya

9 de febrero de 2020

1 Verificación

Se presenta el proyecto para el cálculo y computación de los parámetros aerodinámicos de perfiles 4-digit NACA usando el método DVM.

En primer lugar, la primera tarea requerida se trata de la verificación de la solución del Método de Vórtice Discreto (DVM). Para ello, se considera el perfil NACA 2408. En su estudio, se considera el perfil sin deflexión de flap y por ende, la línea media del perfil no se ve alterada. Esta verificación se ha realizado con un ángulo de ataque $\alpha = 4^\circ$ y se ha comparado con los valores exactos extraídos de la Teoría de perfiles Delgados (TAT).

En consonancia con la explicación anterior, se analiza el coeficiente de sustentación del perfil para un número de paneles N_{panels} en sendas distribuciones *uniforme* y *full coseno* a lo largo de la cuerda. En su estudio, se analiza el C_l obtenido para distintos números de paneles desde los 0 – 200 paneles.

Paralelamente, se compara el valor obtenido en comparación con el valor teórico ofrecido por el TAT y a su vez, se extrae el error relativo de ambos C_l para cada número de paneles [Véase en la Figura 1].

El Discrete Vortex Method proporciona una aproximación a la solución de la teoría de perfiles delgados (Thin Airfoil Theory, TAT). El resultado que se tiende a obtener es, por tanto, la solución de éste. Observando los resultados que se adquieren mediante esta teoría, se

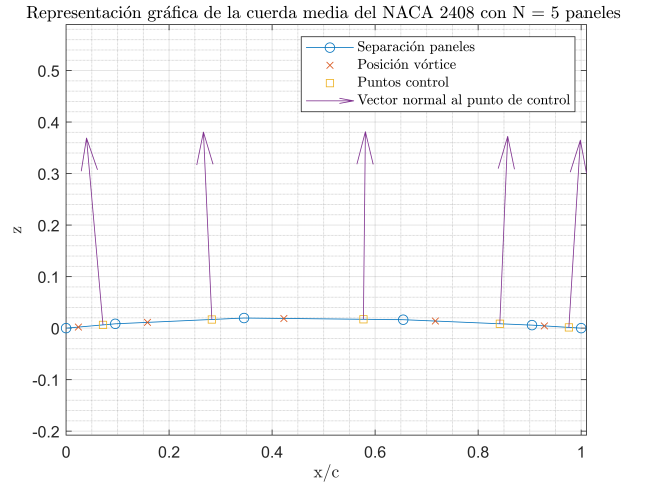


Figura 1 Representación gráfica de la cuerda media del NACA 2408 con $N = 5$ paneles con distribución *Full Coseno*

obtiene $C_l = 0,6664$ y $C_{m_{LE}} = -0,2197$. Se evalúa el error respecto estos valores:

Se observa que, con pocos paneles, como era predecible, se obtiene un error significativo y, a partir de unos $N = 100$ paneles el porcentaje de error ya es menor del 0,5%. De tal modo que este valor puede ser una buena primera aproximación.

En el caso del momento respecto al borde de ataque, se puede ver que la relación entre el error y número de

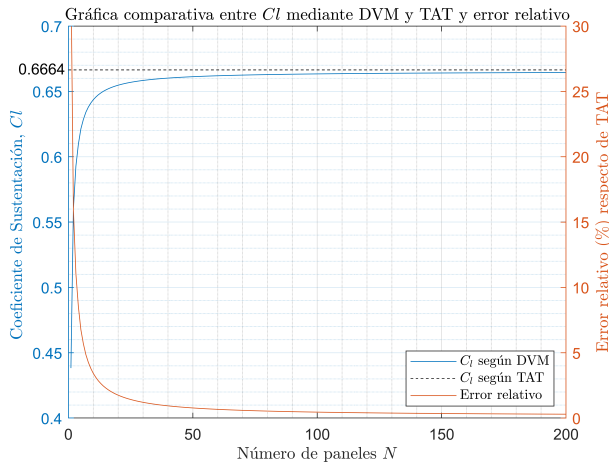


Figura 2 Representación gráfica del cálculo del C_l usando DVM y TAT y error relativo

paneles presenta una tendencia análoga. La precisión del resultado aumenta con el número de paneles hasta los 64 paneles aproximadamente, la cual este valor empieza a estabilizarse y varía muy ligeramente con el aumento del número de paneles.

Para un análisis más preciso, se presenta la tabla 1. Para cada número de paneles, se calcula el error relativo y absoluto. Analizando los errores de la tabla 1¹, se puede observar que el error para 64 paneles aun se encuentra en el rango del 0,6378 %. Sin embargo, en el caso de los 128 paneles esta cifra es inferior al 0,5 % para el caso del C_l . En el caso del C_m , este valor sigue siendo superior al 0,5 % para $N = 200$ con la cual cosa también se encuentra en un rango aceptable. Por ende, es posible coger $N = 128$ valor como el número de paneles mínimo que debe tener el perfil para ofrecer una buena aproximación a la solución teórica.

2 Validación

Seguidamente, en esta sección se presentan los cálculos y el código de programación necesario para la obtención del valor de los parámetros pendiente de sustentación Cl_α , el ángulo de sustentación nula α_{l0} , así como el coeficiente de momento alrededor del centro aerodinámico del perfil o también conocido como coeficiente de momento libre Cm_0 así como la eficiencia del flap para distintos *flap-chord ratio* mediante DVM.

A primera vista se puede observar como la curva de sustentación, estudiada en el tramo donde su respuesta

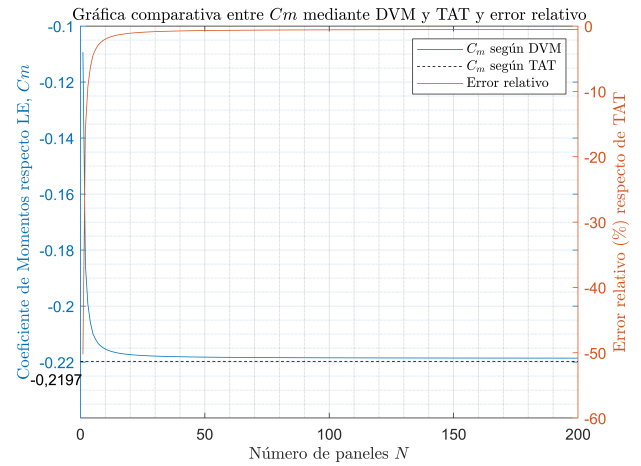


Figura 3 Representación gráfica del cálculo del $C_{m_{LE}}$ usando DVM y TAT y error relativo

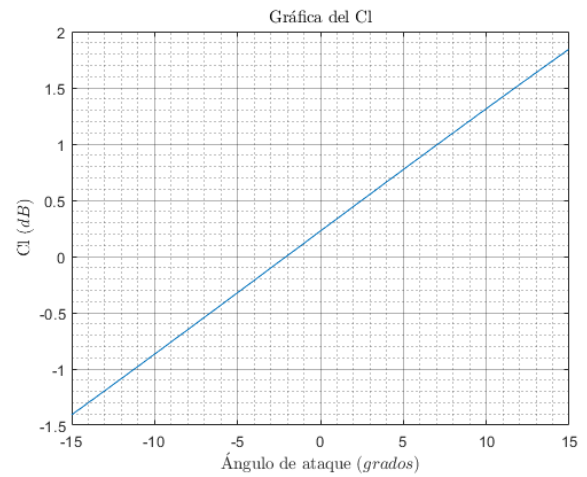


Figura 4 Representación gráfica del C_l del NACA 2408 con $N = 200$ paneles con distribución *Full Coseno*

es lineal, se trata de una recta cuyo pendiente es positivo y el valor del cual lo podemos obtener a partir de la ecuación de la recta que se muestra a continuación. Al mismo tiempo operando con esta misma ecuación también podemos obtener el valor del ángulo de sustentación nula.

Por otro lado, refiriéndonos al coeficiente de momento libre, podemos apreciar como este responde, tal y como hemos estudiado en las clases de teoría, de manera constante para cualquier valor de ángulo de ataque. El valor de este Cm_0 lo podemos obtener de dicha gráfica y también se muestra a continuación.

$$C_l = 6,2413\alpha + 0,2241 \text{ (rad)}$$

¹Ésta se ha hecho en potencias de 2 para facilitar cálculos

Número de paneles	ΔC_l	Error C_l (%)	ΔC_m	Error C_m (%)
1	0,2281	34,2341	0,1104	50,2544
2	0,1062	15,9467	0,0344	15,6727
4	0,0568	8,5271	0,0139	6,3238
8	0,0284	4,2689	0,0056	2,5569
16	0,0136	2,1675	0,0028	1,2954
32	0,0076	1,1412	0,0018	0,8286
64	0,0042	0,6378	0,0014	0,6377
128	0,0025	0,3891	0,0012	0,5533
200	0,002	0,3000	0,0011	0,5247

Tabla 1 Errores obtenidos en función del número de paneles

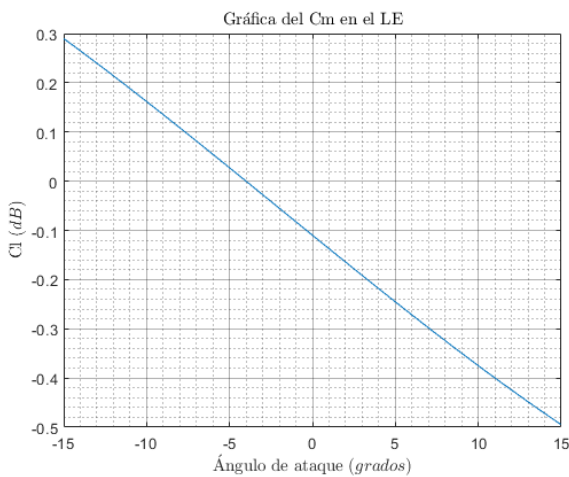


Figura 5 Representación gráfica del C_{m_0} del NACA 2408 con $N = 200$ paneles con distribución *Full Coseno*

$$C_{l\alpha} = 6,2413 \text{ rad} \approx 2\pi \text{ rad}$$

$$\alpha_{l_0} = -0,0359 \text{ rad} = -2,0569^\circ$$

$$C_{m_0} = -0,045$$

Tal y como se puede observar en la tabla anterior, los resultados experimentales obtenidos del libro de referencia ² son realmente similares a los que hemos obtenido mediante el método numérico basado en el DVM. A consecuencia de que el error más grande no supera el 12% podemos dar por correctos y aptos los resultados obtenidos teniendo en cuenta que una pequeña

variación repercute considerablemente. Como se ha comentado anteriormente, los resultados de este apartado se han obtenido a partir de analizar el perfil con un número equivalente a 200 paneles. Es por eso que el error, por muy pequeño que sea, no se puede menospreciar y deberá ser tenido en cuenta a la hora de presentar cualquier resultado obtenido mediante dicho método numérico. Se destaca el hecho de que el error que presenta el coeficiente de momento libre sea el más elevado de los tres.

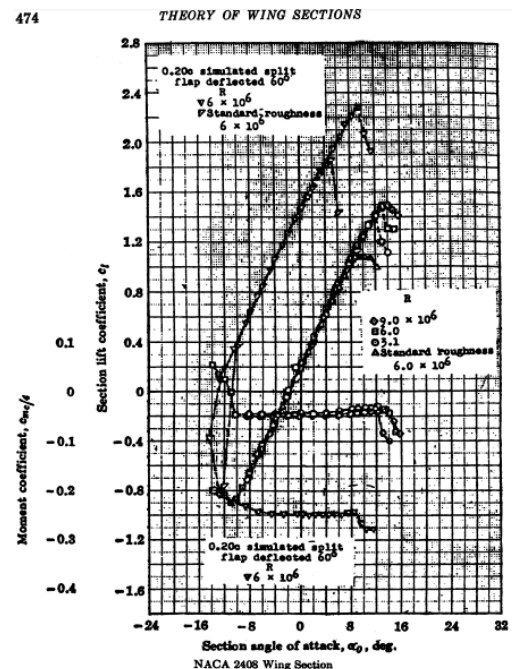


Figura 6 Representación gráfica de la curva de sustentación (C_l) en función del ángulo de ataque (α)

La razón de ese fenómeno puede ser debida a las condiciones en que se han tomado los datos experimentales,

²Abbott, Ira H. y Doenhoff, Albert E. 1959. *Theory of wing sections* New York : Dover Publications, INC., 1959. 468-60586-8.

	Parámetros		
	$C_{l\alpha}$	$\alpha_{l0}(^\circ)$	C_{m0}
Numéricos	6.2413	-2.0569	-0.045
Experimentales	6.01	-2,24	-0.051
Error %	3.61	8.9	11.1

Tabla 2 Comparación resultados teóricos (DVM) vs resultados experimentales.

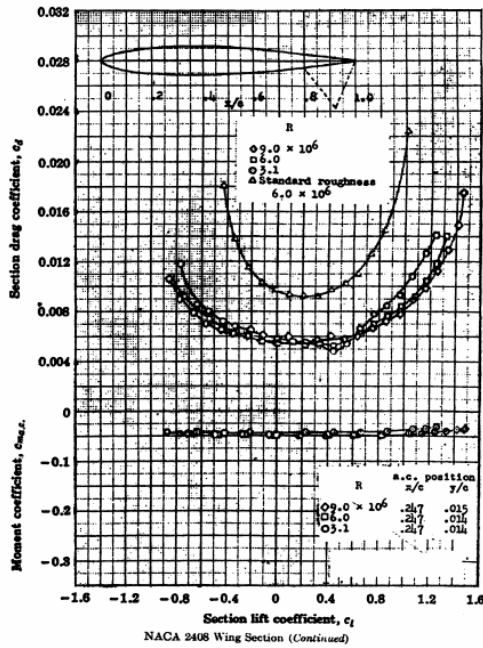


Figura 7 Representación gráfica de la polar C_d y el coeficiente de momento C_{m0} en función de C_l

debido a que toda medida experimental conlleva, ya de por sí, un cierto margen de error. De cualquier forma y analizando los resultados globalmente, está claro que el error es pequeño y existe una gran similitud entre los datos calculados numéricamente y los obtenidos a través de pruebas y mediciones.

2.1. Cálculo de la eficiencia del flap, E

Para completar la validación procederemos a comparar la eficiencia del flap obtenida mediante el método de vórtice discreto (DVM) con la teórica obtenida en el libro de referencia: *Theory of wings sections* de Abbot y Doenhoff.³

A continuación, se estudiará el factor de eficiencia del flap en función del *flap-chord ratio*. Éste se define como: $E = 1 - x_h$. La eficiencia del flap es $\frac{\Delta\alpha_{l0}}{\Delta\eta}$, es decir,

³Abbott, Ira H. y Doenhoff, Albert E. 1959. *Theory of wing sections* New York : Dover Publications, INC., 1959. 468-60586-8.

la variación del ángulo de sustentación nula en respecto al ángulo de deflexión del flap. Cabe recordar, que x_h (*hinge*) corresponde a la posición del eje de charnela respecto de la cuerda. Por ende, esta es la posición donde empieza la deflexión del flap.

Por teoría de perfiles delgados, se cumple que $\Delta\alpha_{l0}$ es proporcional linealmente al ángulo de deflexión η .

Para calcular la eficiencia se calculó la siguiente relación de incrementos de forma iterativa para un rango de η entre 0 i 10° :

$$\frac{\Delta\alpha_{l0}}{\Delta\eta} \quad (1)$$

Esta ecuación (1) nos proporciona un solo valor, así pues se calculó para cuatro posiciones posibles de la posición de comienzo del flap: 0,15, 0,2, 0,25, 0,3. Por lo tanto, la representación gráfica consiste en estos cuatro puntos.

Además se incluye otra serie de 4 puntos obtenidos mediante la multiplicación por un *factor de corrección* de 0,88. Con el fin de presentar la comparación en un solo gráfico se calculó una regresión de los valores teóricos obtenidos en la referencia anterior.

La gráfica obtenida para un numero de paneles $N = 200$ y una geometría de discretización de *Full cosine* es la siguiente:

En primer lugar, la Figura 8 se comparan los valores de DVM sin factor de corrección, con factor frente a los valores experimentales.

Se puede observar la gran similitud entre los valores del DVM corregido y los obtenidos experimentalmente. Cuanto mayor es la E , menor es la diferencia entre ambas curvas, pudiendo suponer que a partir de $E = 0,3$ las dos curvas sean coincidentes.

Este hecho es debido a la teoría de que, para E pequeñas, la relación entre el espesor de la capa límite y el tamaño de flap incrementa, provocando así que la eficiencia obtenida en la práctica disminuya.

De este modo se explica que para E mayores, los

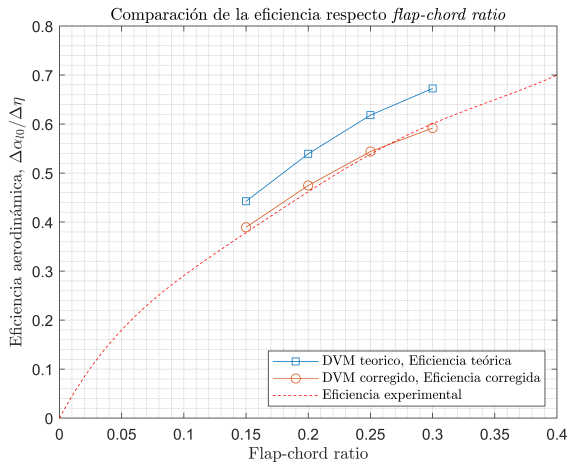


Figura 8 Eficiencia del flap

resultados se asemejen más a los teóricos. El factor de corrección es lo que hace que se elimine este error para E pequeñas. Como se observa en la Figura 8, para estos valores hay una gran semejanza entre los valores del DVM y los de TAT corregidos. Sin embargo, a partir de $E = 0,2$ el factor deja de tener efecto y se aleja más de los valores experimentales.

Podemos observar que los puntos obtenidos mediante el método de vórtice discreto (DVM) siguen la misma tendencia que la curva teórica, hecho que se puede comprobar al introducir el anterior factor de corrección donde la diferencia es como máximo un 1 %, pudiéndose decir por lo tanto que se trata de una buena aproximación.

3 Discusión

Finalmente en este tercer último apartado se procederá a comparar los efectos de variar la máxima combadura de la línea media (f) y la posición relativa de dicha combadura máxima en cada perfil (p). Para ello se estudiarán y analizarán los valores de ángulo de sustentación nula y de coeficiente de momento libre para distintos valores razonables de f y p .

En ambas gráficas podemos observar como el comportamiento tanto del C_m como del α_{l0} , a medida que aumentamos los valores de f tanto como los de p tienden a aumentar en módulo. No obstante, solo lo hacen en módulo, por lo que nos lleva a la conclusión de que a medida que un perfil va aumentando su máxima combadura así como va retrasando la posición de esta, los valores de su coeficiente de momentos libre y ángulo de

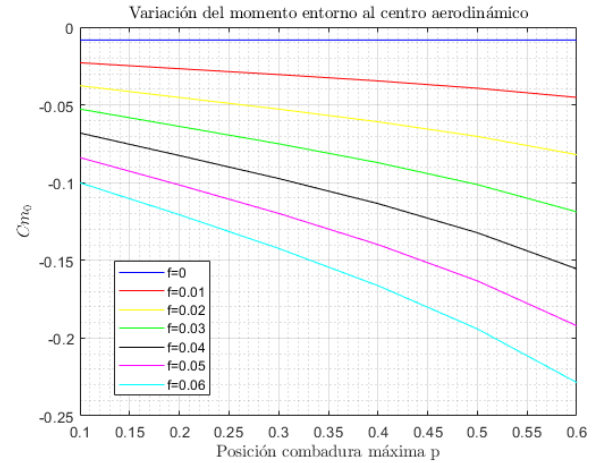


Figura 9 Coeficiente de momento libre C_{m0} en función de p

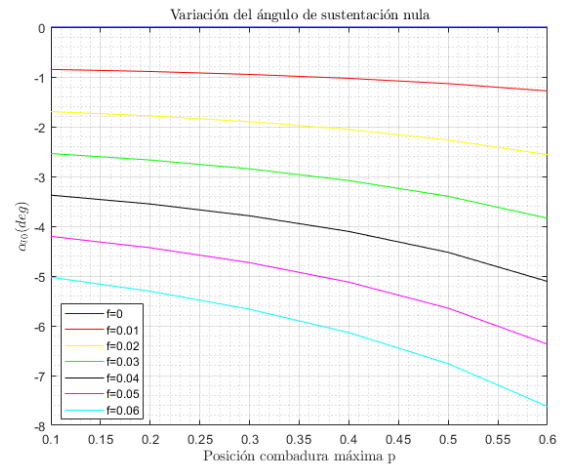


Figura 10 Valor de α_{l0} en función de p

sustentación nula tienen a ser cada vez más negativos.

En la primera gráfica [Figura 10, se puede observar cómo el ángulo de sustentación nula α_{l0} se hace cada vez más grande en módulo. Este hecho tiene sentido físico, ya que se está aumentando la máxima combadura y, por ende, se aumenta la circulación, que a su vez supone un incremento del coeficiente de sustentación Cl (según el teorema de *Kutta-Joukowski*). Esto provoca que, para un mismo ángulo de ataque α el perfil genere más sustentación cuanto mayor sea su combadura, por lo que el α_{l0} aumentará en valor absoluto [Véase en la Figura 11].

El comportamiento sobre la variación del ángulo de sustentación nula se parece en gran medida a uno de los efectos que se obtienen al realizar una deflexión ocasionada por la introducción de flaps en el perfil. De hecho,

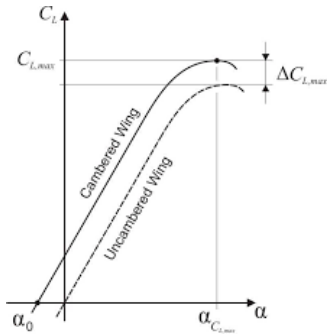


Figura 11 Curva del coeficiente de sustentación en función de ángulo de ataque α

al modificar f y p se está cambiando la geometría del ala y, por lo tanto, modificando la función de su línea media, retrasando el punto de máxima combadura y aumentando la misma, tal y como ocurre cuando aplicamos una deflexión de flaps sobre la línea media. Un hecho interesante que se observa de estas gráficas es que para valores fijos de máxima combadura relativa, el ángulo de sustentación nula se hace cada vez más negativo a medida que se va retrasando la posición de ésta sobre el perfil.

Refiriéndonos al coeficiente de momento libre, Cm_0 observamos como para una placa plana, cuyo perfil tendría valor 0 máxima combadura, este coeficiente es de valor constante nulo por lo que tal debido a que no se provoca ninguna perturbación lo demasiado grande sobre el flujo como para que el perfil induzca un momento libre por sí mismo. Además, se debe destacar cómo todas las rectas parten de un $Cm_0 = 0$ para una combadura $f = 0$ (caso de placa plana, la cual no genera momento libre como se ha comentado anteriormente), y todas ellas van aumentando, en valor absoluto, a medida que aumenta la combadura f .

Los perfiles que presentan más momento libre son aquellos que tienen la máxima combadura más alejada del borde de ataque, por tanto una p mayor.

De hecho, en efecto, al aumentar ambos valores p y f , tanto α_{i0} como Cm_0 aumentan, de manera que existe un conflicto a la hora de diseñar el perfil deseado. Aumentar el ángulo de sustentación nula resulta beneficioso en términos de sustentación, pues proporcionará más sustentación para un ángulo de ataque menor. Sin embargo, a su vez se le induce un momento indeseado en el perfil, que afectará a la estabilidad del avión. De este modo, hay que encontrar una solución óptima entre α_{i0} y Cm_0 que no comprometa a los parámetros

aerodinámicos del avión ni las cargas estructurales.

En definitiva, des de nuestro punto de vista y considerando todos nuestros conocimientos sobre la teoría de perfiles y su comportamiento podemos afirmar que estos resultados se asemejan bastante a los que esperábamos cuando se procedió a realizar este último apartado de discusión. No obstante, es necesario destacar la idea de que estos resultados son el fruto de un análisis basado en ciertas hipótesis que facilitan los cálculos, de tal modo que la hora de comprobarlos frente a los resultados experimentales, siempre va a existir un ligero error fruto de estas aproximaciones.

4 Anexo

4.1. Anexo 1: Código de Matlab: Main Code

```

1 naca_vector = [2 4 0 8]; % Asking the NACA Airfoil Parameters - Airfoil Designation
2
3 aattack_deg = 5; % Asking for Angle Attack of Analysis
4 aattack_rad = deg2rad(aattack_deg);
5
6 N_panels = 5; % Asking for Number of Panels
7
8 flap_existence = false; % Additional Flap % True or False
9 xh = 1;
10 eta = 0;
11 eta_rad = deg2rad(eta);
12
13 distribution = 'a'; % Geometry discretization % (a) Uniform and (b) Full cosine
14
15 [nu_matrix,v_matrix,points_cp_matrix] = GEOMD_function (naca_vector,N_panels,
    flap_existence,xh,eta_rad,distribution);
16
17 [circulation,coeficients] = CIRCULATION_function(N_panels,aattack_rad,nu_matrix,v_matrix,
    points_cp_matrix);
18
19 % Loads Calculations
20 Cl = 2*sum(circulation); % Cl calculation
21 angle = cos(aattack_rad) % Cm_LE Calculations
22 Cm_LE = 0;
23
24 for i=1:N_panels
25 Cm_LE = Cm_LE -2*circulation(i)*v_matrix(1,i)*angle;
26 end

```

4.2. Anexo 2: Código de Matlab: Geometry Discretization

```

1
2 function [nu_matrix,v_matrix,points_cp_matrix] = GEOMD_function (naca_vector,N_panels,
    flap_existence,xh,eta_rad,distribution)
3
4 f = naca_vector(1)/100; % Obtention of the maximum camber from profile designation
5 p = naca_vector(2)/10; % Obtention of the maximum camber position from profile
    designation
6 t = (naca_vector(3)*10+naca_vector(4))/100; % Obtention of the maximum thickness from
    profile designation
7 N=N_panels+1; % Data points
8 points_matrix = zeros (2,N); % Matrix for saving data points (1,N es x / 2,N es z)
9 points_cp_matrix = zeros (2,N_panels); % Matrix for saving data of Control Points
10
11 if flap_existence == false
12 xh=2;
13 end
14 zh=0;
15 for i=1:N
16     if (distribution == 'a' || distribution == 'A') % Uniform distribution
17         xi = (i-1)/(N_panels);
18         points_matrix(1,i) = xi;
19     else % Full cosine function distribution
20         xi = (1/2)*(1 - cos((i - 1)/(N - 1)*pi));
21         points_matrix(1,i) = xi;
22     end
23

```

```

24         if(0<xi) && (xi<=p)
25             zc1 = (f/(p^2))*(2*p*xi - xi^2);
26             points_matrix(2,i) = zc1;
27         elseif xi>p && xi<=1 && xi<=xh
28             zc2 = (f/((1-p)^2))*(1 - 2*p + 2*p*xi - xi^2);
29             points_matrix(2,i) = zc2;
30             zh=zc2;
31         elseif xi>xh && (i<=N)
32             zc2 = (f/((1-p)^2))*(1 - 2*p + 2*p*xi - xi^2);
33             points_matrix(2,i) = zc2;
34             xi=points_matrix(1,i);
35             zi=points_matrix(2,i);
36             points_matrix(1,i)=xh+(xi-xh)*cos(eta_rad)+(zi-zh)*sin(eta_rad);
37             points_matrix(2,i)=zh-(xi-xh)*sin(eta_rad)+(zi-zh)*cos(eta_rad);
38         end
39     end
40     points_matrix;
41     plot(points_matrix(1,:),points_matrix(2,:), '-o');
42
43 %% Calculation of the numerical data for each panel
44 % Vectors for saving x,z from each panel vector
45 px_vector = zeros(1,N_panels);
46 pz_vector = zeros(1,N_panels);
47 % Vector for saving data points of the lenght of each panel
48 panel_lenght = zeros(1,N_panels);
49 % Vectors for saving x,y of the unitary vector of each panel
50 ux_vector = zeros(1,N_panels);
51 uz_vector = zeros(1,N_panels);
52 % Vectors for saving x,y of the unitary normal vector to each panel
53 nux_vector = zeros(1,N_panels);
54 nuz_vector = zeros(1,N_panels);
55 % Vectors for saving data points x,y of the position of the vortex
56 vx = zeros(1,N_panels);
57 vz = zeros(1,N_panels);
58 %v_matrix = [vx; vz];
59
60 for a = 1:N_panels
61     % Calculating the panel vectors, its modulus and unitary vectors
62     px_vector(1,a) = points_matrix(1,a+1) - points_matrix(1,a);
63     pz_vector(1,a) = points_matrix(2,a+1) - points_matrix(2,a);
64     panel_lenght(1,a) = sqrt((px_vector(1,a))^2 + (pz_vector(1,a))^2);
65     ux_vector(1,a) = px_vector(1,a) / panel_lenght(1,a);
66     uz_vector(1,a) = pz_vector(1,a) / panel_lenght(1,a);
67     % Calculating the unitary normal panel vectors
68     nux_vector(1,a) = - uz_vector(1,a);
69     nuz_vector(1,a) = ux_vector(1,a);
70     % Calculating the vortex positions
71     vx(1,a) = points_matrix(1,a) + 0.25 * panel_lenght(1,a) * ux_vector(1,a);
72     vz(1,a) = points_matrix(2,a) + 0.25 * panel_lenght(1,a) * uz_vector(1,a);
73     % Calculating the control points positions
74     points_cp_matrix(1,a) = points_matrix(1,a) + 0.75 * panel_lenght(1,a) * ux_vector
75         (1,a);
76     points_cp_matrix(2,a) = points_matrix(2,a) + 0.75 * panel_lenght(1,a) * uz_vector
77         (1,a);
78 end
79 v_matrix = [vx; vz];
80 nu_matrix = [nux_vector; nuz_vector];
81 hold on
82 plot(vx,vz,'x'); % For verification purposes of the vortex positions
83 plot(points_cp_matrix(1,:),points_cp_matrix(2,:), 's')
84 quiver(points_cp_matrix(1,:),points_cp_matrix(2,:),nu_matrix(1,:),nu_matrix(2,:))
85 xlabel('x/c', 'interpreter', 'latex');

```



```

84     ylabel('z','interpreter','latex');
85     title('Representaci\''on gr\''afica de la cuerda media del NACA 2408 con N = 5 paneles
        ','interpreter','latex');
86 end

```

4.3. Anexo 3: Código de Matlab: Circulation

```

1
2 function [circulation,coeficients] = CIRCULATION_function(N_panels,aattack_rad,nu_matrix,
    v_matrix,points_cp_matrix)
3
4 vx = v_matrix(1,:); % Vortex
5 vz = v_matrix(2,:);
6 circulation = zeros(N_panels,1); % Circulation of each panel
7 coeficients = zeros(N_panels,N_panels); % Coeficients Matrix
8 RHS = zeros(N_panels,1);
9
10 for i=1:N_panels
11     xi = points_cp_matrix(:,i);
12     ni = nu_matrix(:,i);
13
14     for j=1:N_panels
15         xj = [vx(j); vz(j)];
16         delta = xi - xj ;
17         r_square = dot(delta,delta) ;
18         factor = 0.5 / pi / r_square ;
19         u_ind = factor * delta(2);
20         w_ind = - factor * delta(1);
21         coeficients(i,j) = u_ind * ni(1) + w_ind * ni(2);
22     end
23     RHS(i,1) = - 1 * (cos(aattack_rad) * ni(1) + sin(aattack_rad) * ni(2));
24 end
25 circulation = inv(coeficients) * RHS;
26 end

```

4.4. Anexo 4: Código de Matlab : Thin Airfoil Theory (TAT)

```

1 t=0.08; p=0.4;f=0.02;
2 coef1=f/(p^2);
3 coef2=f/(1-p)^2;
4 fun11= @(x) coef1.*(2*p-1+cos(x));
5 fun12= @(x) coef2.*(2*p-1+cos(x));
6 fun21= @(x) coef1.*(2*p-1+cos(x)).*cos(x);
7 fun22= @(x) coef2.*(2*p-1+cos(x)).*cos(x);
8 fun31= @(x) coef1.*(2*p-1+cos(x)).*cos(2*x);
9 fun32= @(x) coef2.*(2*p-1+cos(x)).*cos(2*x);
10 A0=4*pi/180-1/pi*integral(fun11,0,acos(1-2*p))-1/pi*integral(fun12,acos(1-2*p),pi);
11 A1=2/pi*(integral(fun21,0,acos(1-2*p))+integral(fun22,acos(1-2*p),pi));
12 A2=2/pi*(integral(fun31,0,acos(1-2*p))+integral(fun32,acos(1-2*p),pi));
13 TAT_C1=2*pi*(A0+A1/2)
14 TAT_Cmle=-TAT_C1/4+pi/4*(A2-A1)
15 Cmo=pi/4*(A2-A1)

```

4.5. Anexo 5: Código de Matlab : α_{l0} and C_{m0}

```

1 N_panels = 200; flap_existence = false; xh =0; eta=0; eta_rad=0; j=1;
2 for f = 0:0.01:0.06
3     for p = 0.1:0.1:0.6
4         naca_vector = [f p 0 0]
5         aattack_rad = -pi/12;
6         for i=1:524;

```

```

7      [dist,nu_matrix,v_matrix,points_cp_matrix] = GEOMD_function (naca_vector, N_panels
      ,flap_existence,xh,eta_rad,f,p);
8      [circulation,coeficients] = CIRCULATION_function(N_panels,aattack_rad,nu_matrix,
      v_matrix,points_cp_matrix);
9      [Cl,Cm_LE] = LOADS_function(circulation,v_matrix,aattack_rad,N_panels);
10     Cl_vector(i,j) = Cl;
11     Cm_LE_vector(i,j) = Cm_LE;
12     alpha(i) = aattack_rad;
13     aattack_rad = aattack_rad + 0.001;
14     end
15     total_circulation(:,j) = circulation(:,1);
16     j = j+1;
17 end
18 end
19 for i=1:42
20 pend_Cl = polyfit(alpha',Cl_vector(:,i),1);
21 pend_Cm_LE = polyfit(alpha',Cm_LE_vector(:,i),1);
22 pend_Cl_Cm = - polyfit(Cl_vector(:,i),Cm_LE_vector(:,i),1); %xac
23 Cm_0(i) = -2*sum(total_circulation(:,i)'.*(v_matrix(1,:)-pend_Cl_Cm(1)));
24 alpha_l0(i) = -pend_Cl(2)/pend_Cl(1);
25 end

```

4.6. Anexo 6: Código de Matlab : Flap efficiency

```

1  naca_vector=[2,4,0,8]; xh_vec=[0.15,0.2,0.25,0.3];
2  alpha_lo_r=zeros(10,1); eta_rad_vec=zeros(10,1); eff=zeros(4,2);
3  for i=1:1:4
4      xh=xh_vec(i);
5      for a=0:1:9
6          eta_rad=deg2rad(a);
7          eta_rad_vec(a+1)=deg2rad(a);
8          for b=1:1:35
9              aattack_rad_vec(b)=deg2rad(b-25);
10             aattack_rad=deg2rad(b-25);
11             [nu_matrix,v_matrix,points_cp_matrix] = GEOMD_function (naca_vector, N_panels
              ,0,1-xh,eta_rad,0);
12             [circulation,coeficients] = CIRCULATION_function(N_panels,aattack_rad,
              nu_matrix,v_matrix,points_cp_matrix);
13             Cl(b) = 2*sum(circulation);
14             [nu_matrix,v_matrix,points_cp_matrix] = GEOMD_function (naca_vector, N_panels
              ,1,1-xh,eta_rad,0);
15             [circulation,coeficients] = CIRCULATION_function(N_panels,aattack_rad,
              nu_matrix,v_matrix,points_cp_matrix);
16             Cl_f(b) = 2*sum(circulation);
17         end
18         func=polyfit(aattack_rad_vec,Cl,1);
19         alpha_l0=-func(2)/func(1);
20         func_f=polyfit(aattack_rad_vec,Cl_f,1);
21         alpha_l0_f=-func_f(2)/func_f(1);
22         alpha_lo_r(a+1)=alpha_l0_f-alpha_l0;
23     end
24     eff(i,:)=polyfit(eta_rad_vec,alpha_lo_r,1);
25 end
26
27 %Experimental plot
28 eff_t=[0,0.18,0.29,0.38,0.46,0.54,0.6,0.65,0.7]; xh_t_vec=0:0.05:0.4;
29 %Theoretical plot
30 eff_t_r=polyfit(xh_t_vec,eff_t,7); x1 = linspace(0,0.4); y1 = polyval(eff_t_r,x1);
31 %Corrected plot
32 eff_corregido=-eff(:,1)*0.88;
33 %Overlay plots
34 plot(xh_vec,-eff(:,1),'s',xh_vec,eff_corregido,'o',x1,y1,'r--');

```