UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Aerospace Structures

Beam structure: Wingbox

Sergi Marsol

Nicolás Cortines

Greta- Group 11

PROFESSOR

Juan Carlos Cante

Tuesday 23rd April, 2024

Contents

1 Condiciones iniciales de análisis

La geometría en cuestión presentada en la práctica es:

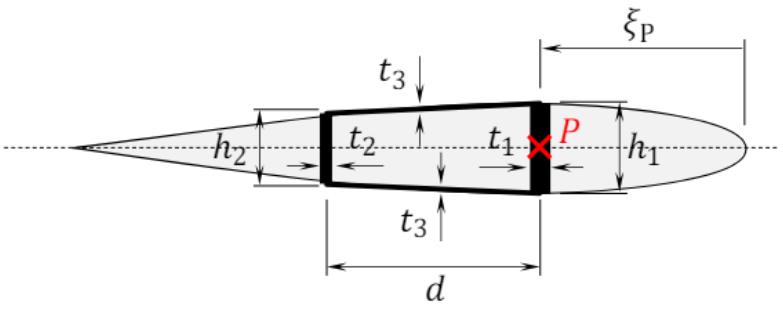

Figure 1: Wing Cross-section

Donde el módulo de elasticidad del material es $E = 210$ GPa y el módulo de cizalladura es $G = 80$ GPa. Los parámetros geométricos en la Figura 1 son: $\zeta_P = 0.3c$, $d = 0.3c$, $h_1 = 0.25c$, $h_2 = 0.15c$, $t_1 = 22$ mm, $t_2 = 15$ mm y $t_3 = 3.5$ mm.

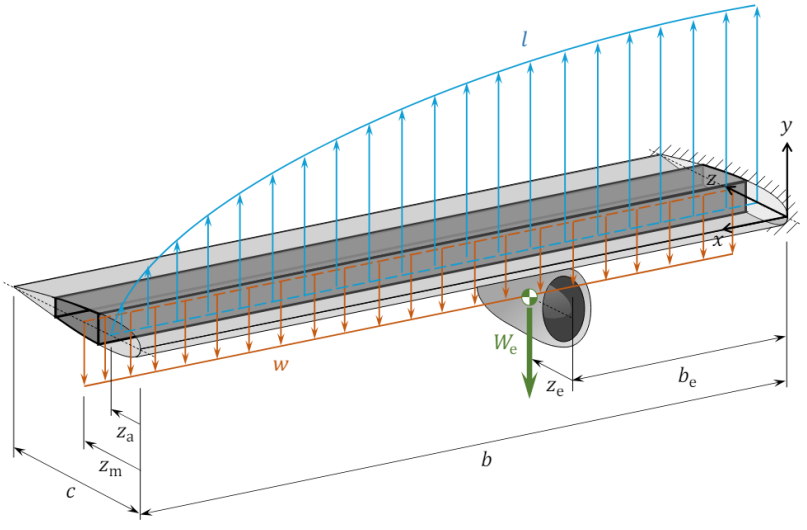

Figure 2: Wing loads distribution

El ala está sometida a una carga distribuida l que actúa en la cuerda aerodinámica, situada a una distancia $z_a = 0.25c$ desde el borde de ataque. También soporta su propio peso, w, con una masa uniformemente distribuida $\lambda = 140$ kg/m a lo largo de la línea de centro de masa, situada a una distancia $z_m = 0.48c$. Además, en el punto representado en la Figura 2, con $b_e = 0.25b$ y $z_e = 0.3c$, el ala soporta el peso del motor de masa $M_e = 2100$ kg. Para la carga aerodinámica, se asume la siguiente distribución:

$$l(x) = \frac{1}{2}\rho_\infty V_\infty^2 \frac{c_l}{c}\left(1 - \left(\frac{x}{b}\right)^2\right)$$

donde $\rho_\infty = 1.225$ kg/m^3, $V_\infty = 750$ km/h, $c_l = 0.1$, $c = 2$ m, $b = 16$ m.

2 Cross-section analysis

2.1 Centroid, bending inertia and normal stress distribution

En esta primera sección, se ha calculado el centroide de la pieza, su inercia y la distribución de tensiones normales σ, como resultado de un momento unitario en la dirección del eje z. La geometría en cuestión presentada es:

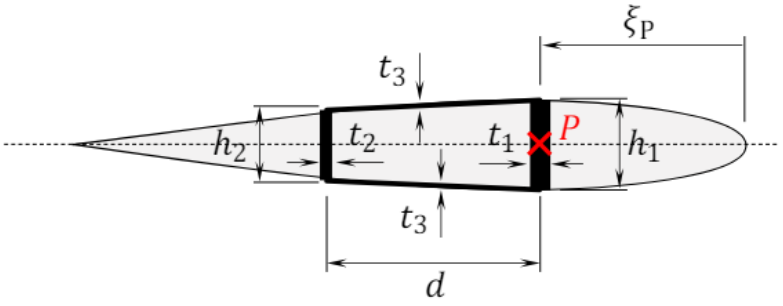

Figure 3: Wing Cross-section

Sin embargo, cabe mencionar que, en primer lugar, se ha debido de discretizar la geometría usando un número adecuado de nodos. Como consecuencia, se ha escogido implementar 50 nodos a la *cross-section*. Este número de nodos proporcionará la precisión necesaria para llevar a cabo las cuestiones planteadas.

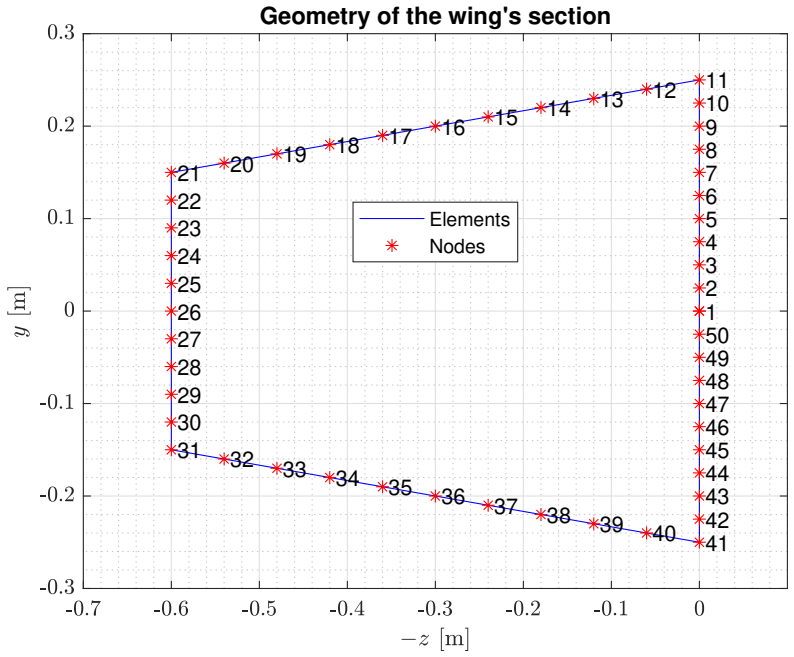

Figure 4: Wing Cross-section

Una vez establecida la geometría del problema, se han procedido a los consiguientes cálculos. En primer lugar, la posición del **centroide** de la sección (x_0, y_0) con respecto el punto 'P' es:

Position (m)	x'_0	y'_0
Value (m)	-0.2013	0

Asimismo, a continuación se presenta el valor de las **inercias** calculadas. Como consecuencia de la simetría de la pieza, debe de verificarse qué $I_{xy} = 0$. Los resultados de los pertinentes cálculos se detallan a continuación:

	Inertia Value (m^4)
I_{xx}	$4.3678 \cdot 10^{-4}$
I_{yy}	0.0013
I_{xy}	0

De forma matricial resulta:

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} = \begin{pmatrix} 4,3678 \cdot 10^{-4} & 0 & 0 \\ 0 & 0,0013 & 0 \\ 0 & 0 & 0 \end{pmatrix};$$

Por último, se ha aplicado un momento flector unitario alrededor del eje z de la estructura ($Mx = -1$ y $My = 0$ en el caso del eje de la sección) y se ha obtenido la **distribución de tensiones normales**, ***Normal stress*** σ. Para ello, se ha ejecutado la función *getNormalStress* codificada en *Matlab*. La distribución resultante se observa a continuación.

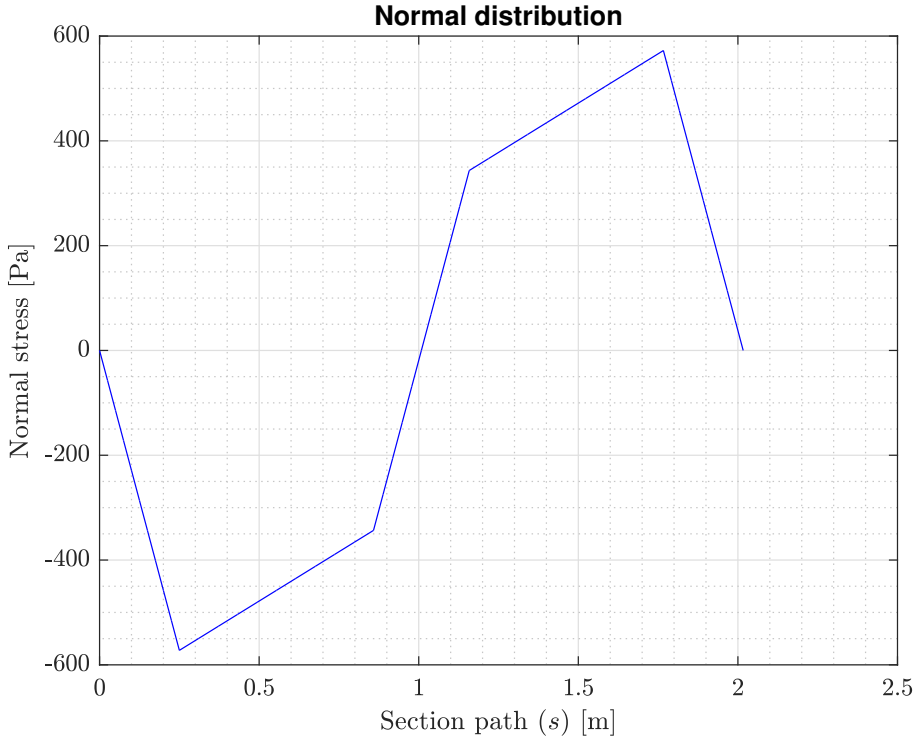

Figure 5: Normal stress distribution

De forma esperada, se puede observar como la tensión normal tanto al inicio como al final de la sección es similar, ya que dicha distribución debe de ser continua a lo largo del dominio. Por otro lado, el punto

de máxima tensión normal se encuentra en $s = 1.75$ m, donde dicho valor del *normal stress* alcanza el $\sigma = 572.4$ Pa.

2.2 Shear center, torsional inertia and tangential stress distribution assuming open section

A continuación, se ha calculado la inercia a torsión, centro de cortante y distribución de esfuerzos tangenciales a lo largo de la sección, asumiendo que la pieza se encuentra abierta. Dicha distribución de esfuerzos tangenciales (τ_o^s y τ_o^t) son causados por un cortante y momento torsor unitarios, aplicados en el centro de cortante.

En primer lugar, el valor de la inercia de torsión, ***torsional inertia for open section,*** es $J_{open} = 2.08562 \cdot 10^{-6}\ m^4$.

$$J_{open} = 2.08562 \cdot 10^{-6}\ m^4$$

Una vez se ha obtenido la inercia necesaria, es posible calcular la posición del **centro de cortante** (x_s, y_s), ***shear center***. Para ello, primero se debe determinar la distribución de esfuerzos cortantes para luego aplicar el equilibrio de momentos en cualquier punto de la sección y así calcular las coordenadas del centro de cortante.

Position (m)	x_s	y_s
Value (m)	-1.1358	0

Una vez más, se verifica que, de forma esperada, el centro de cortante se sitúa sobre el eje de simetría.

Después de realizar los siguientes cálculos, se ha computado la distribución de esfuerzos tangenciales, ***tangential stress distribution*** causados por un momento torsor en dirección z ($M_z = 1$) y un cortante en dirección y ($S_y = 1$), cuyo punto de aplicación se encuentra en el centro de cortante.

En primer lugar, para obtener la distribución de esfuerzos tangenciales como resultado de un cortante unitario, es decir, ***tangential stress distribution due to shear*** se ha empleado la función *getTangentialStressDistribution*.

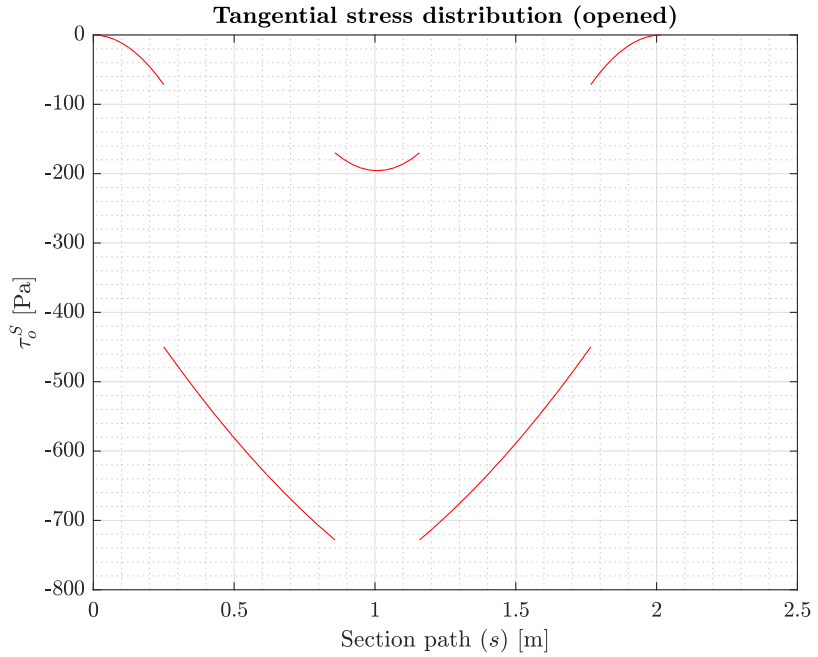

Figure 6: Tangential stress distribution, opened

Por otro lado, para el caso de esfuerzo tangencial como resultado de un momento torsor unitario, es decir, ***tangential stress distribution due to torsion*** se ha empleado la función *getTangentialStressDistributionTorsion*.

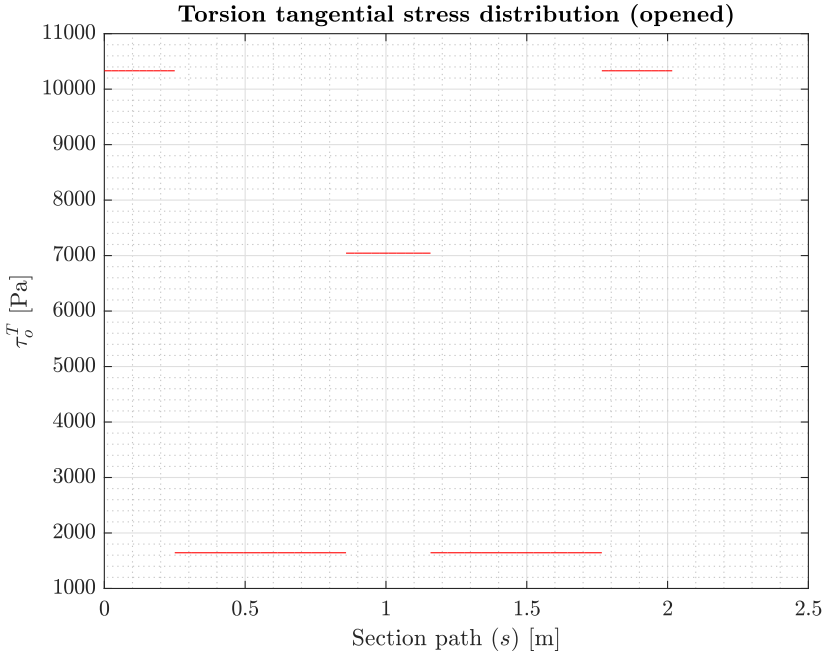

Figure 7: Tangential stress distribution torsion, opened

Se puede observar, que ambas distribuciones no son continuas en cada vórtice de cada sección. Esta

discontinuidad es resultado de las variaciones de grosor de la sección. Asimismo, es interesante visualizar que aunque la distribución de tensiones tangenciales es discontinua, el flujo de cortadura es una función continua a lo largo de la *cross-section* ($q = \tau \cdot t$).

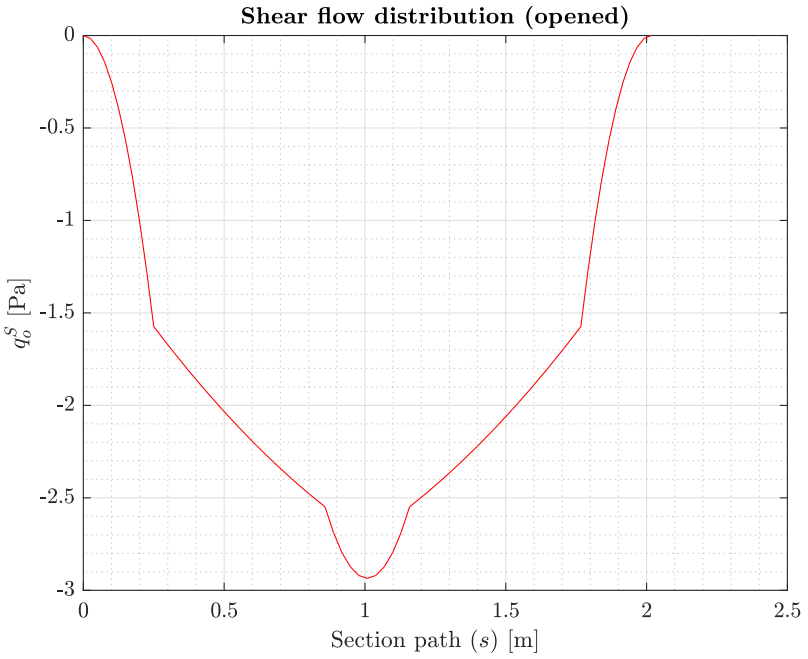

Figure 8: Shear flow opened

2.3 Torsional inertia and tangential stress distribution assuming closed section

En este último apartado se ha calculado la inercia polar y la distribución de esfuerzos tangenciales a lo largo de la sección, asumiendo que la pieza se encuentra cerrada. Dicha distribución de esfuerzos tangenciales (τ_c^s y τ_c^t) son causadas por un cortante y momento torsor unitarios aplicados en el centro de cortante de la pieza. Para este caso, se asumirá que el centro de torsión de la sección coincide con el centroide; por lo que, $X_s = X_0$.

En el siguiente gráfico se puede observar la posición del centroide, centro de cortante y la geometría discretizada.

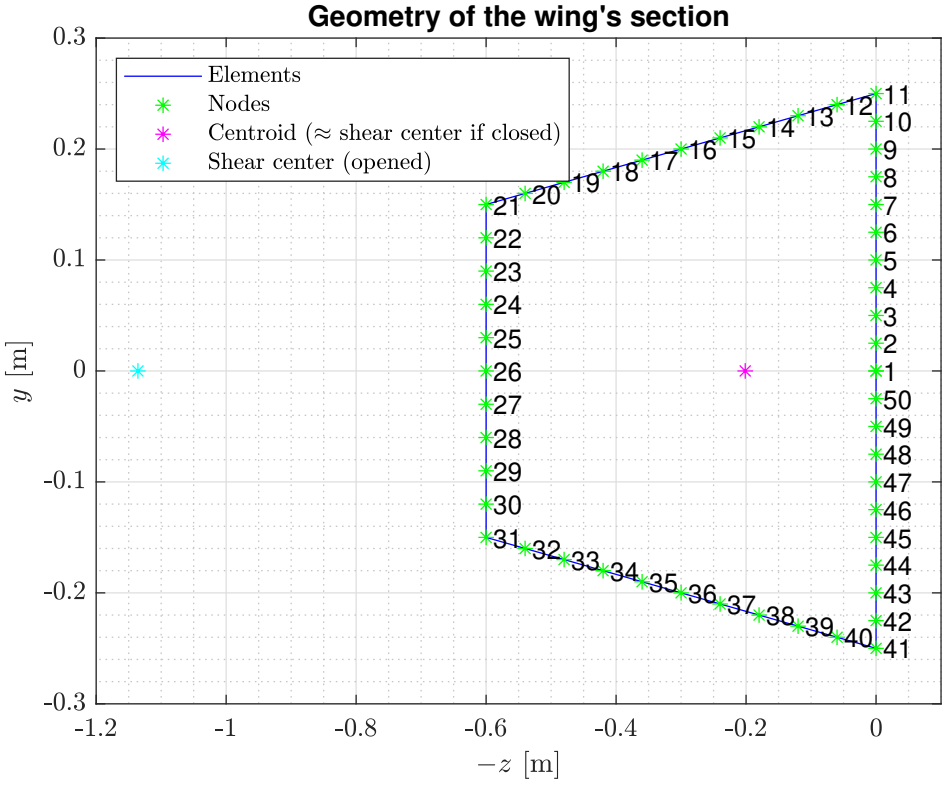

Figure 9: Cross-Section mesh

En primer lugar, el valor del módulo de torsión, ***torsional inertia for closed section,*** es $J_{closed} = 5.9029 \cdot 10^{-5}\ m^4$. Dicho valor se ha obtenido a partir de la función *getSectionPropertiesClosed* de *Matlab*.

$$J_{closed} = 5.9029 \cdot 10^{-5}\ m^4$$

Por ende, la distribución de esfuerzos tangenciales a lo largo de la sección como consecuencia del cortante unitario, ***tangential stress distribution due to shear (closed)*** se detalla en la siguiente figura. Dicha distribución se ha obtenido de forma similar al caso de sección abierta, pero sumándole el flujo de cortadura en la sección inicial q_{so}.

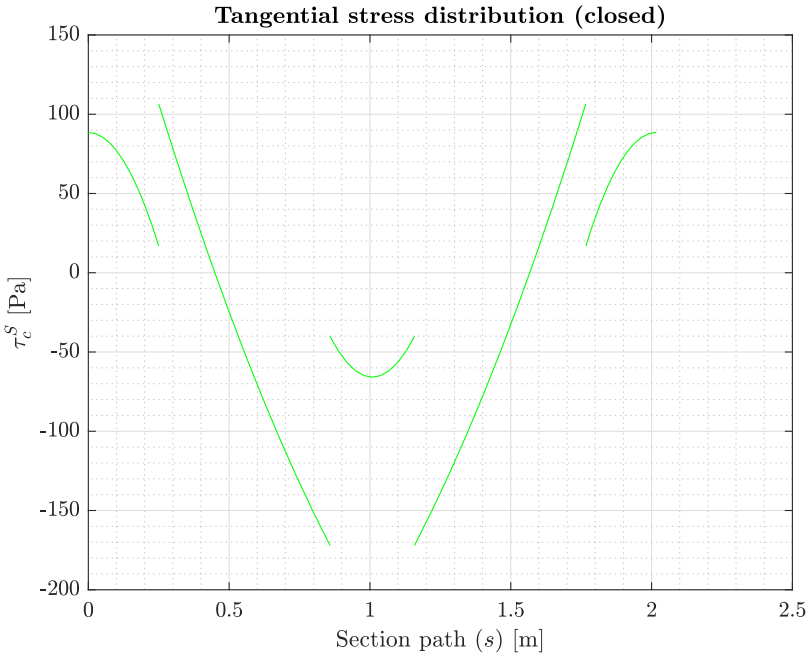

Figure 10: Tangential stress distribution closed

Finalmente, para el esfuerzo tangencial como resultado de un momento torsor unitario, es decir, ***tangential stress distribution due to torsion***:

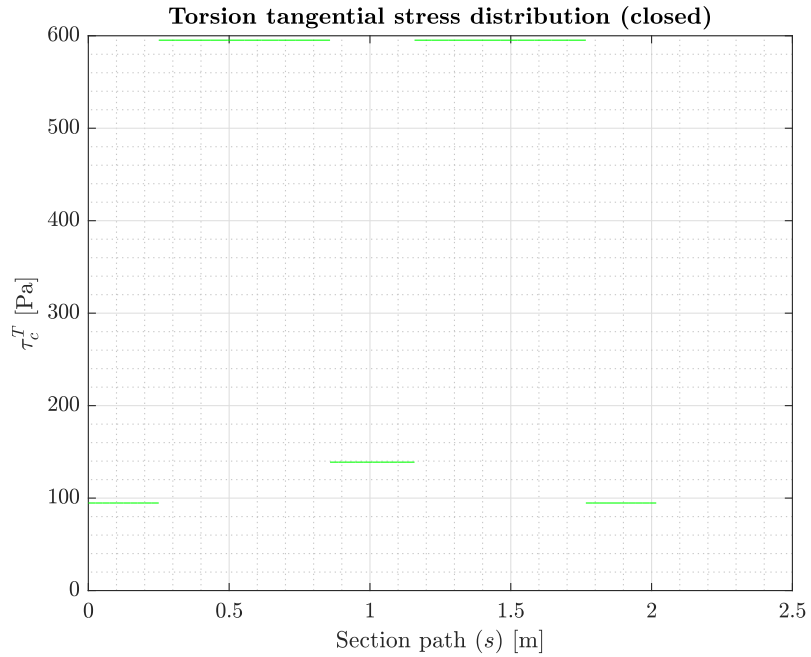

Figure 11: Tangetial stress distribution torsion, closed

De forma similar al apartado anterior, el flujo de cortadura resultante es:

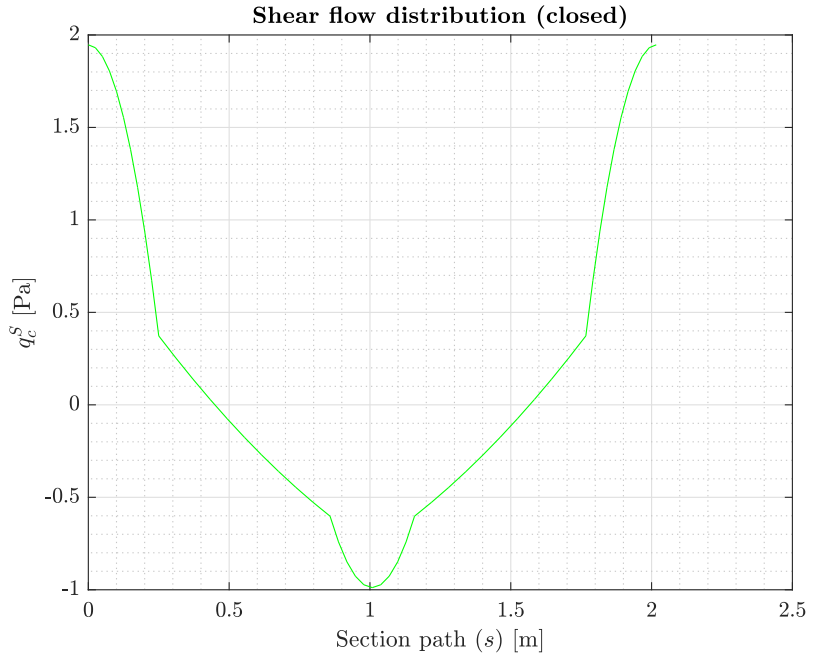

Figure 12: Shear flow closed

La tendencia seguida por el flujo de cortante es idéntica a la mostrada en el caso abierto, con la única diferencia dé q_{so}. Anteriormente, se observaba que en el punto 'P' el flujo de cortante era nulo, mientras que ahora es distinto de dicho valor, consecuencia de que, para pasar del caso abierto al caso cerrado, es necesario agregar el término q_{so}, que corresponde a:

$$q_s^{closed} = q_s^{open} + q_{so}$$

3 Beam analysis for both closed and open section

En esta sección, se ampliará el análisis realizado para los casos de sección cerrada y abierta al conjunto del ala modelada por una viga expuesta a las cargas ya presentadas.

3.1 Wingbox magnitudes analysis

En primer lugar, se ha implementado un código con el objetivo de calcular numéricamente ciertas magnitudes de la estructura alar, tanto para el caso cerrado como abierto. Asimismo, se han computado las distribuciones de las siguientes magnitudes.

- Vertical deflection
- Section bending rotation
- Section torsion rotation
- Shear force
- Bending moment
- Torsion moment

3.1.1 Algorithm

A continuación, se presentan las funciones y ciertas líneas de código con el objetivo de completar la discretización de toda la sección alar, al mismo tiempo que los parámetros anteriormente mencionados. Cabe mencionar que la distinción entre el caso abierto y cerrado radica en la inercia polar y posición del centro de cortante.

En primer lugar, se han computado las fuerzas y momentos repartidos que actúan sobre cada elemento. Las fuerzas puntuales, como ahora el peso del motor, y los momentos puntuales, también debido al peso del motor, no se han considerado en este caso, puesto que se han contemplado como input de fuerzas y momentos puntuales a la hora de resolver la estructura, como se verá posteriormente.

```
%%
% Fuerza distribuida equivalente resultante en cada ELEMENTO
distributed_lift_nodes = arrayfun(lift_func,x);
distributed_lift_elements = (distributed_lift_nodes(1:end-1) + distributed_lift_nodes(2:end))/2;
distributed_load = distributed_lift_elements - weight_distribution;

% Momento torsor distribuido equivalente resultante en cada ELEMENTO
distributed_torsion_lift = (chi_s+chi_p-za)*distributed_lift_elements;
distributed_torsion = distributed_torsion_lift - (chi_s+chi_p-zm)*weight_distribution;

% Fuerzas puntuales y momentos/torsores en cada NODO
motor_node_position = find(x==be);
F = [motor_node_position 1 -Me*g
     motor_node_position 3 -Me*g*(chi_s+chi_p-ze)]; % Point loads (only the engine)
```

Por consiguiente, a partir de las funciones presentadas en el *Appendix D. Beam problems*, en concreto, *D.2 Element stiffness function*, *D.3 Element force function* y *D.4 Element internal forces function*, al mismo tiempo que el conjunto de funciones que establecen el *solver* de problemas matriciales en *Matlab* se han podido computar las distribuciones y magnitudes deseadas.

```
    %%
    % SOLVER:
    fel = ForceFunction(Data,X,Tn,fe,me); % Compute element force vectors
    [K,f] = assemblyFunction(Data,Td,Kel,fel); % Assemble global stiffness matrix
    [up,vp] = applyBC(Data,P); % Apply prescribed DOFs
    [u,r] = solveSystem(Data,K,f,up,vp); % Solve system
```

En concreto, para obtener la ***vertical deflection*** y la ***section bending and torsion rotation*** se han utilizado la gran mayoría de dichas funciones. La función *ForceFunction* calcula los vectores de

fuerza de cada elemento, lo que permite el ensamblaje de la matriz de rigidez global mediante la función *assemblyFunction*. Después de prescribir los grados de libertad iniciales mediante *applyBC*, el uso de *solveSystem*, que utiliza el ensamblaje realizado y la prescripción para determinar el desplazamiento de cada elemento discreto, devuelve el vector deseado.

Por otro lado, para obtener la ***shear force, bending moment and torsion moment*** en el centro de cortante, se ha implementado la función *D.4 Element internal forces function*.

3.2 Convergence of the numerical solution

Para estudiar la convergencia de la solución numérica para diferentes tamaños de elementos en la discretización de la viga ($h_{el} = \{\frac{b}{4}, \frac{b}{8}, \frac{b}{16}, \frac{b}{32}, \frac{b}{64}\}$), se ha calculado el error relativo de la deflexión y la torsión en la punta del ala en cada caso en comparación con un caso de referencia con un número grande de elementos.

El error relativo se define como:

$$\epsilon(h_{el}) = \left| \frac{u_{h_{el}} - u_{h_{ref}}}{u_{h_{ref}}} \right|$$

donde:

- h_{el} es el tamaño del elemento considerado.
- $u_{h_{el}}$ es la solución obtenida con el tamaño de elemento h_{el}.
- $u_{h_{ref}}$ es la solución de referencia obtenida con un tamaño de elemento muy pequeño, por ejemplo $h_{ref} = \frac{b_b}{512}$.

Los tamaños de elementos considerados son $h_{el} = \{\frac{b_b}{4}, \frac{b_b}{8}, \frac{b_b}{16}, \frac{b_b}{32}, \frac{b_b}{64}\}$.

Tanto para el caso abierto como cerrado, los gráficos del error relativo de la deflexión y torsión en la punta del ala son:

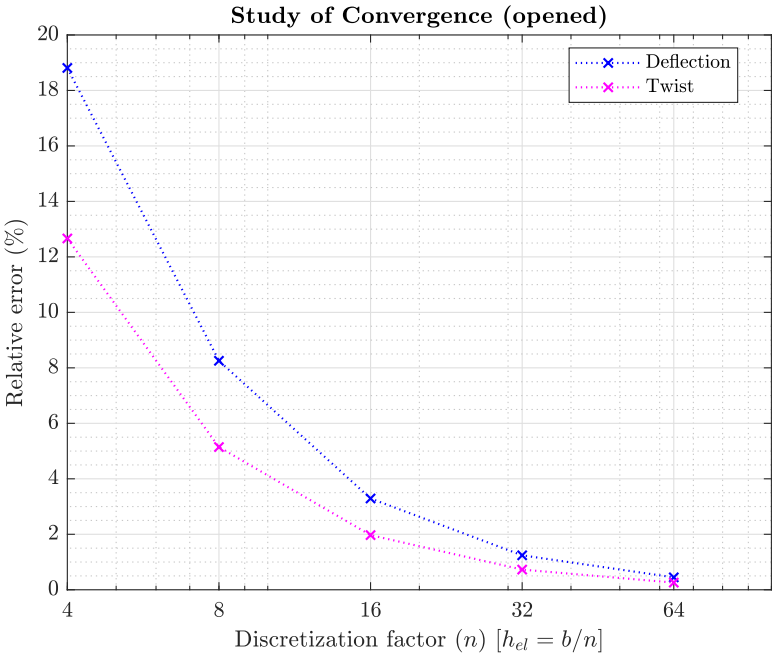

Figure 13: Study of Convergence (opened)

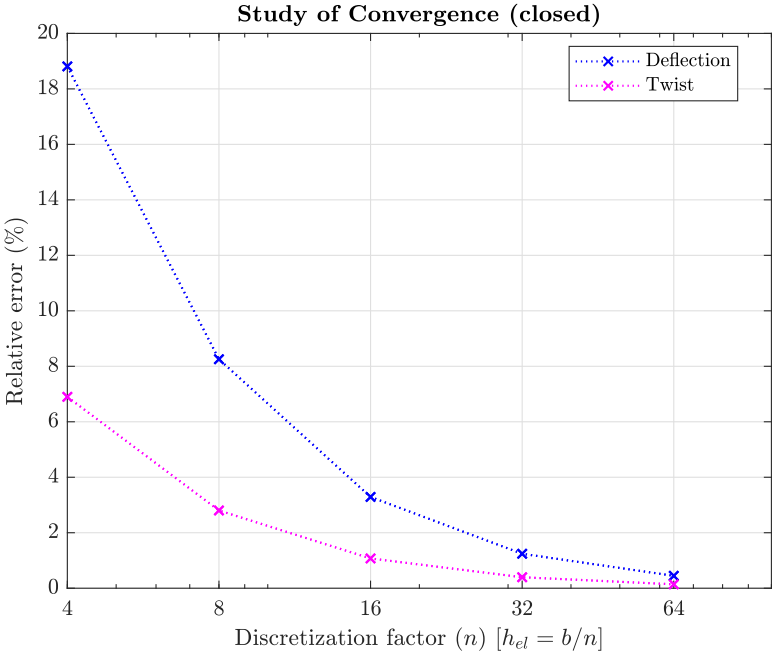

Figure 14: Study of Convergence (closed)

3.3 Wingbox magnitudes analysis, reference case ($h_{el} = \frac{b}{512}$)

Para el caso de referencia, se han graficado las magnitudes descritas en el punto 3.1, tanto para el caso abierto como cerrado.

Sección abierta

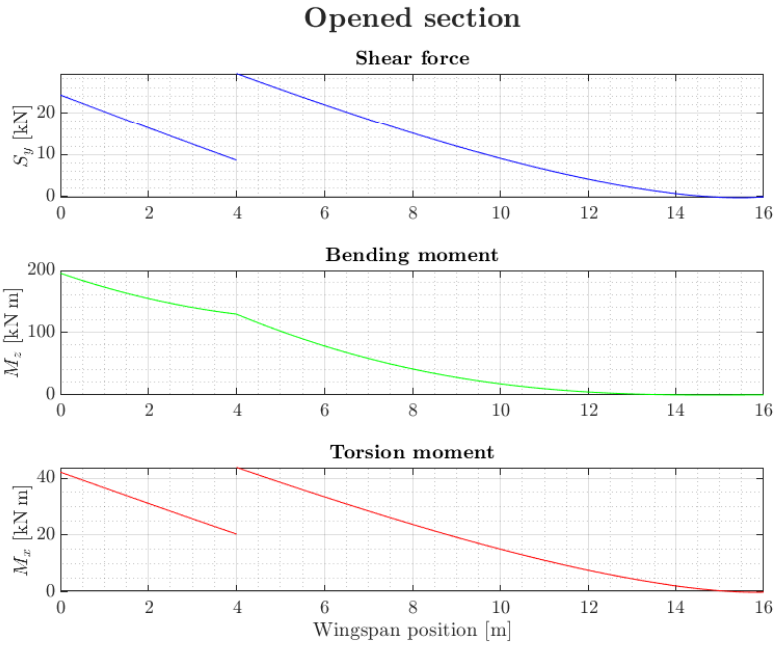

Figure 15: Shear force, bending moment and Torsion moment, opened

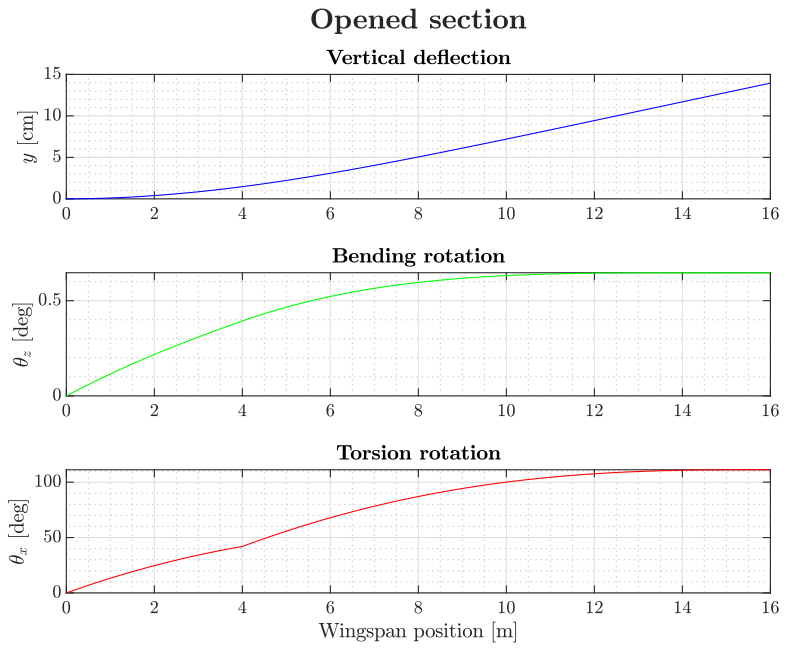

Figure 16: Vertical deflection, bending rotation and torsion rotation, opened

Sección cerrada

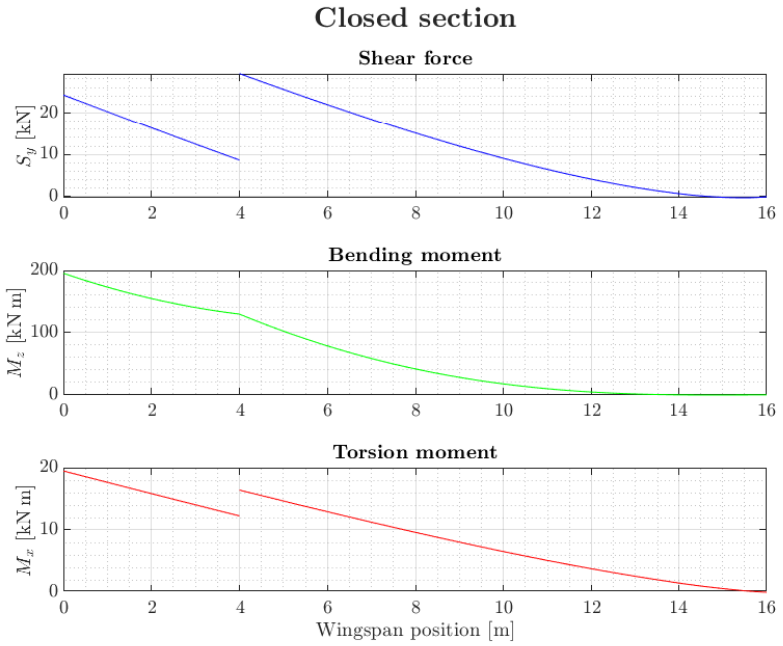

Figure 17: Shear force, bending moment and Torsion moment, closed

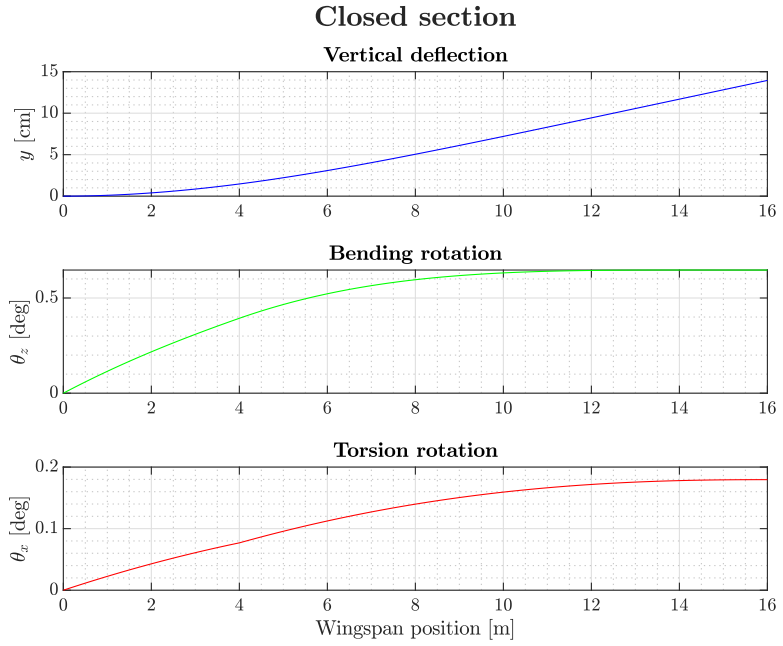

Figure 18: Vertical deflection, bending rotation and torsion rotation, closed

4 Von Mises criterion

Por último, se ha aplicado el criterio de Von Mises para localizar el punto más crítico en el interior de la estructura del ala, tanto para los casos de sección cerrada como abierta. Para ello, se tiene en cuenta que dicho criterio constata que la carga última sigue la forma:

$$\sigma_{\text{VM}} = \sqrt{\sigma^2 + 3\tau^2}$$

Por ende, se obtienen los resultados:

	Open Section	**Closed Section**
Beam Position (m)	4	0
Section Position (m)	$(0, -0.25)$	$(0, 0.25)$
Stress Value (MPa)	788.90	118.44

Los puntos donde se encuentran las máximas tensiones admisibles, para cada caso, se ubican en la sección donde se encuentra el turbofan y en él encastre del ala respectivamente. En concreto, dichos puntos en cada sección se encuentran:

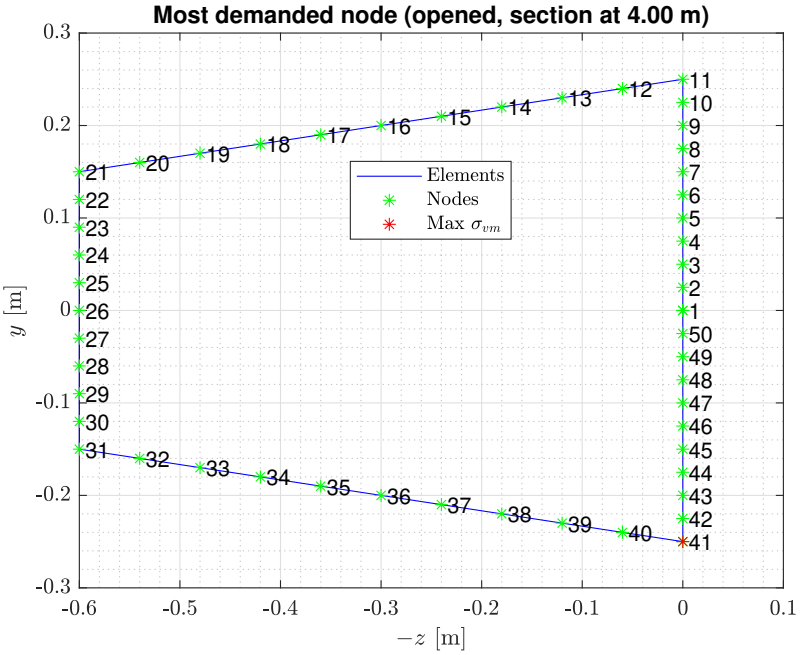

Figure 19: Von Mises criterion node, opened

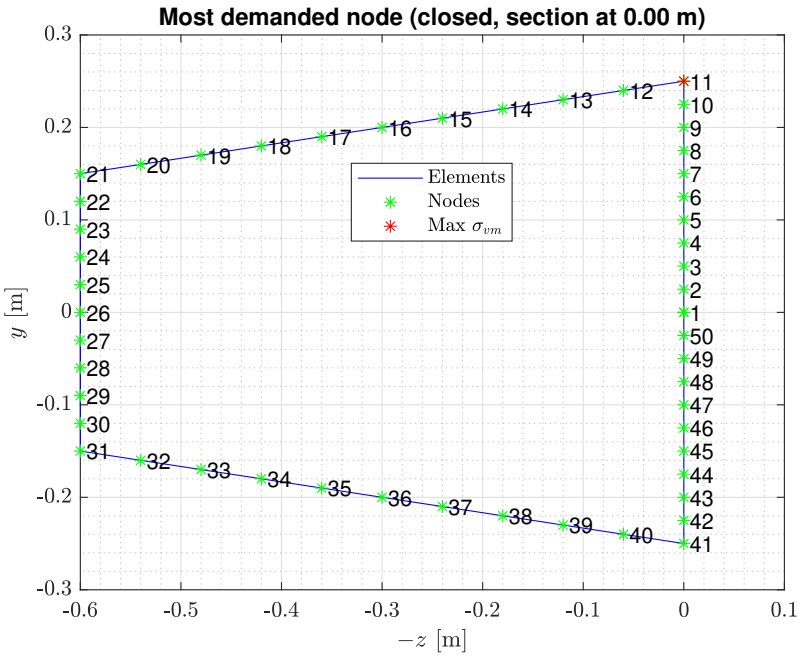

Figure 20: Von Mises criterion node, closed

5 Conclusiones

Los resultados geométricos obtenidos en referencia a la sección transversal del ala son los esperados, debido a la simetría de la estructura. En relación con las fuerzas y deformaciones obtenidas, cabe destacar el ángulo de giro de torsión del ala en el caso abierto frente al cerrado. Para el abierto se ha obtenido un ángulo máximo de giro (en el *wingtip*) 626 veces mayor que para el caso de sección transversal cerrada. Esto es debido al módulo de torsión para el caso cerrado, pues es mucho mayor en comparación con el abierto. También destaca el hecho que tanto las deformaciones verticales y la rotación por flector obtenidas para ambos casos (abierto y cerrado) como el cortante y flector, coinciden. La explicación de dicho suceso se atribuye a la aproximación o asunción de que el centro de cortante coincide con el centro de torsión para el caso cerrado, de no haber sido así, los resultados diferirían.

En relación con la convergencia en la deflexión y torsión en el *wingtip*, podemos afirmar que a partir de utilizar como longitud de discretización del ala la trigesimosegunda parte de la longitud de la misma, obtenemos un error menor al 2%, tanto en el caso abierto como en el cerrado. Destacar que la convergencia en la deflexión es la misma para ambos casos, puesto que las deflexiones son las mismas por lo mencionado anteriormente. Para la sección cerrada, el error relativo obtenido para la torsión es menor, de todas formas, en ambos casos se aprecia una rápida convergencia.

Finalmente, se ha estudiado la tensión máxima de Von Mises a la cual se somete la estructura. Por un lado, para el caso abierto, la sección más solicitada se encuentra en la misma sección del ala donde se sitúa el motor. Por otro lado, dicha sección, para el caso cerrado, se ubica en el encastre del ala. Cabe mencionar que la tensión máxima de Von Mises, **a igual solicitación de la estructura**, es mucho mayor para el caso abierto que para el cerrado, como consecuencia la estructura con sección transversal abierta fallaría primero.

La práctica realizada, pues, ha sido un buen ejercicio para trabajar la discretización y la implementación de la teoría trabajada en clase respecto al cálculo de esfuerzos y requerimientos estructurales, así como para asentar y repasar conceptos.

A Código empleado

A.1 Sección A

En primer lugar, el main de la sección A:

```
%% CODE, MAIN A:
clear; clc; close all;
addpath('functions\')

% Datos geométricos
c = 2; % [m]
d = .3*c;
h1 = .25*c;
h2 = .15*c;
t1 = 22e-3;
t2 = 15e-3;
t3 = 3.5e-3;
n_subBars = 10;

% Datos del material
E = 210e9;
G = 80e9;

% Geometría y propiedades
x = [0 0
     0 h1/2
     -d h2/2
     -d -h2/2
     0 -h1/2
     0 0];

[x_disc, Tn] = discretize(x,n_subBars);

nel = length(Tn);

m = [t1 E G
     t2 E G
     t3 E G];

Tm = [];
for e = 1:length(x)-1
    if e==1 || e==5
        Tm = [Tm;ones(n_subBars,1)*1];
    end
    if e==2 || e==4
        Tm = [Tm;ones(n_subBars,1)*3];
    end
    if e==3
        Tm = [Tm;ones(n_subBars,1)*2];
    end
end

%% A) Cross-section analysis

% a.1)
[Xc,Xs,Atot,Ixx,Iyy,Ixy,J] = getSectionProperties(x_disc,Tn,m,Tm);
[~,~,~,~,~,~,J_closed,Ain] = getSectionPropertiesClosed(x_disc,Tn,m,Tm);

% a.2) i a.3)
[sigma, s_n] = getNormalStressDistribution(x_disc,Tn,Xc,Ixx,Iyy,Ixy,-1,0);

[q, tau, s] = getTangentialStressDistribution(x_disc,Tn,m,Tm,Xc,Ixx,Iyy,Ixy,0,1);
[q_closed, tau_closed, s_closed] = getTangentialStressDistributionClosed(x_disc,Tn,m,Tm,Xc,Ixx,Iyy,Ixy,0,1,Xs,Ain); % assuming Xs=Xc

[tau_t, s_t] = getTangentialStressDistributionTorsion(x_disc,Tn,m,Tm,J,1);
[tau_t_closed, s_t_closed] = getTangentialStressDistributionTorsionClosed(x_disc,Tn,m,Tm,1,Ain);

%
section_data.x_disc = x_disc;
section_data.Tn = Tn;
section_data.Tm = Tm;
section_data.m = m;
section_data.Xc = Xc;
section_data.Xs = Xs;
section_data.Ixx = Ixx;
section_data.Iyy = Iyy;
section_data.Ixy = Ixy;
section_data.Ain = Ain;
section_data.J = J;
section_data.J_closed = J_closed;

save('section_data.mat','section_data')

%% Gràfics

% a.0) Discretization
hold on;
box on;
grid on; grid minor;
plot(x_disc(:,1),x_disc(:,2),'b-')
plot(x_disc(:,1),x_disc(:,2),'*','color','r')
legend('Elements','Nodes','location','best')
title("Geometry of the wing's section");
xlabel('$-z$ [m]','interpreter','latex');
ylabel("$y$ [m]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1)
xlim([min(x(:,1))-0.1,max(x(:,1))+0.1]);
text(x_disc(1:end-1,1),x_disc(1:end-1,2),sprintfc(' %d',1:numel(x_disc(:,1))-1));
hold off;

%%
% a.1)
% Normal stress (opened section)
```

```
figure('Name','Normal stress distribution');
plot(s_n, sigma,'b-')
title('Normal distribution');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("Normal stress [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;


% a.2)
% Shear flow (opened section)
figure('Name','Shear flow distribution (opened)');
plot(s, q,'r-')
title('\textbf{Shear flow distribution (opened)}','Interpreter','latex');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("$q^S_o$ [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Tangential stress (opened section)
figure('Name','Tangential stress distribution (opened)');
plot(s, tau,'r-')
title('\textbf{Tangential stress distribution (opened)}','Interpreter','latex');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("$\tau^S_o$ [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Torsion tangential stress (opened section)
figure('Name','Torsion tangential stress distribution (opened)');
plot(s, tau_t,'r-')
title('\textbf{Torsion tangential stress distribution (opened)}','Interpreter','latex');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("$\tau^T_o$ [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;



% a.3)
% Shear flow (closed section)
figure('Name','Shear flow distribution (closed)');
plot(s_closed, q_closed,'g-')
title('\textbf{Shear flow distribution (closed)}','Interpreter','latex');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("$q^S_c$ [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Tangential stress (closed section)
figure('Name','Tangential stress distribution (closed)');
plot(s_closed, tau_closed,'g-')
title('\textbf{Tangential stress distribution (closed)}','Interpreter','latex');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("$\tau^S_c$ [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Torsion tangential stress (closed section)
figure('Name','Tangential stress distribution (closed)');
plot(s_closed, tau_t_closed,'g-')
title('\textbf{Torsion tangential stress distribution (closed)}','Interpreter','latex');
xlabel('Section path ($s$) [m]','interpreter','latex');
ylabel("$\tau^T_c$ [Pa]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;
```

A.2 Sección B

A.2.1 Caso abierto

El main de la sección B incluyendo el criterio de Von Mises para el caso abierto:

```
clear; clc; close all;

addpath('functions\')
load('section_data.mat')

g = 9.81; % [m/s^2]
% Datos geométricos
chi_s = -section_data.Xs(1); % [m] shear center caso abierto
b = 16; % [m]
be = 0.25*b; % [m]
c = 2; % [m]
ze = 0.3*c; % [m]
za = 0.25*c; % [m]
zm = 0.48*c; % [m]
chi_p = 0.3*c; % [m]
Me = 2100; % [kg]
weight_distribution = 140*g; % [N/m]

vinf = 750/3.6; % [m/s]
rho = 1.225; % [kg/m^3]
Cl = 0.1;
lift_func = @(x)0.5*rho*vinf^2*c*Cl*sqrt(1-(x/b)^2);

% Geometría y propiedades
dx = b/512;
x = 0:dx:b;
```

```
data.x = x;
data.nnod = numel(x);
data.nel = numel(x)-1;
data.nne = 2;
data.ni = 3;
data.ndof = data.nnod*data.ni;  % Total number of degrees of freedom
Tn = zeros(data.nel,2);
for e=1:data.nel
    Tn(e,:) = [e, e+1];
end
Td = connectDOF(data,Tn);

p = [1 1 0 % no desplazamiento vertical en el empotramiento
     1 2 0 % no giro en el empotramiento
     1 3 0]; % no giro de torsion en el empotramiento

% Datos del material
E = 210e9;
G = 80e9;
I = section_data.Ixx; % [m^4] Inercia en el eje x local de la sección transversal (el eje z global del dibujo)
J = section_data.J; % [m^4] Módulo de torsión (polar inertia) (caso abierto)
m = [E G I J];
Tm = ones(data.nel,1); % Toda el ala presenta la misma sección con mismo material


% Fuerza distribuida equivalente resultante en cada ELEMENTO
distributed_lift_nodes = arrayfun(lift_func,x);
distributed_lift_elements = (distributed_lift_nodes(1:end-1) + distributed_lift_nodes(2:end))/2;
distributed_load = distributed_lift_elements - weight_distribution;

% Momento torsor distribuido equivalente resultante en cada ELEMENTO
distributed_torsion_lift = (chi_s+chi_p-za)*distributed_lift_elements;
distributed_torsion = distributed_torsion_lift - (chi_s+chi_p-zm)*weight_distribution;

% Fuerzas puntuales y momentos/torsores en cada NODO
motor_node_position = find(x==be);
F = [motor_node_position 1 -Me*g
     motor_node_position 3 -Me*g*(chi_s+chi_p-ze)]; % Point loads (only the engine)


%% B) Beam analysis (opened)
[u,r, x_el,S_el,Mb_el,Mt_el] = solveStructure(data,Tn,Td,m,Tm,p,distributed_load,distributed_torsion,F);

u_3rows = reshape(u,3,[]);


%% Gràfics

% b.1)
figure('Name','Opened section');
t = tiledlayout(3,1);
title(t,'\textbf{Opened section}','interpreter','latex')

% Vertical deflection (opened section)
ax1 = nexttile;
plot(ax1,x,u_3rows(1,:)*100,'b-')
title(ax1,'\textbf{Vertical deflection}','interpreter','latex');
ylabel("$y$ [cm]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;
%daspect([1 4 .5])

% Bending rotation (opened section)
ax2 = nexttile;
plot(ax2,x,u_3rows(2,:)*180/pi,'g-')
title(ax2,'\textbf{Bending rotation}','interpreter','latex');
ylabel("$\theta_z$ [deg]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Torsion rotation (opened section)
ax3 = nexttile;
plot(ax3,x,u_3rows(3,:)*180/pi,'r-')
title(ax3,'\textbf{Torsion rotation}','interpreter','latex');
ylabel("$\theta_x$ [deg]",'Interpreter','latex');
xlabel('Wingspan position [m]','interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

%


figure('Name','Opened section');
t = tiledlayout(3,1);
title(t,'\textbf{Opened section}','interpreter','latex')

% Shear force (opened section)
ax1 = nexttile;
plot(ax1,x_el,S_el/1000,'b-')
title(ax1,'\textbf{Shear force}','interpreter','latex');
ylabel("$S_y$ [kN]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;
%daspect([1 4 .5])

% Bending moment (opened section)
ax2 = nexttile;
plot(ax2,x_el,Mb_el/1000,'g-')
title(ax2,'\textbf{Bending moment}','interpreter','latex');
ylabel("$M_z$ [kN\,m]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Torsion moment (opened section)
ax3 = nexttile;
plot(ax3,x_el,Mt_el/1000,'r-')
title(ax3,'\textbf{Torsion moment}','interpreter','latex');
ylabel("$M_x$ [kN\,m]",'Interpreter','latex');
xlabel('Wingspan position [m]','interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
```

```
grid minor, grid on;


%% B2 Study of convergence

[discretization_factor, tip_deflection_error, tip_torsion_error] = getTipDeflectoinAndTorsionError(512);

% Gráfico de la convergencia
hold on;
box on;
grid on; grid minor;
plot(discretization_factor,tip_deflection_error*100,'bx:','LineWidth',1)
plot(discretization_factor,tip_torsion_error*100,'mx:','LineWidth',1)
set(gca,'xscale','log')

ticks = [1, 2, 4, 8, 16, 32, 64, 128];
tickLabels = cellstr(num2str(ticks'));
set(gca, 'XTick', ticks, 'XTickLabel', tickLabels);

legend('Deflection','Twist','location','best','interpreter','latex')
title("\textbf{Study of Convergence (opened)}",'interpreter','latex');
xlabel('Discretization factor ($n$) [$h_{el} = b/n$]','interpreter','latex');
ylabel("Relative error (\%)",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1)
hold off;



%% C) Most critical point (Von Misses)

aplana = @(vec)[vec(1,1) vec(1,2:end) vec(2,end)];

cross_section_max_sigma_vm_vec = zeros(1,data.nel);
node_cross_section_max_sigma_vm_vec = zeros(1,data.nel);
for i=1:data.nel
    [~,tau,~] = getTangentialStressDistribution(section_data.x_disc,section_data.Tn,section_data.m,section_data.Tm,section_data.Xc,
      section_data.Ixx,section_data.Iyy,section_data.Ixy,0,S_el(1,i));
    [tau_t,~] = getTangentialStressDistributionTorsion(section_data.x_disc,section_data.Tn,section_data.m,section_data.Tm,section_data.J,-
      Mt_el(1,i));
    [sigma,~] = getNormalStressDistribution(section_data.x_disc,section_data.Tn,section_data.Xc,section_data.Ixx,section_data.Iyy,section_data
      .Ixy,Mb_el(1,i),0);
    tau = aplana(tau); % tau de cada nodo (cada posición es el nodo correspondiente de la discretización de la sección transversal)
    tau_t = aplana(tau_t); % tau_t de cada nodo (cada posición es el nodo correspondiente de la discretización de la sección transversal)
    sigma = aplana(sigma); % sigma de cada nodo (cada posición es el nodo correspondiente de la discretización de la sección transversal)

    sigmas_vm = sqrt(sigma.^2+3*(tau+tau_t).^2);
    [cross_section_max_sigma_vm_vec(i), node_cross_section_max_sigma_vm_vec(i)] = max(sigmas_vm);
end

[wingspan_max_sigma_vm, node_wingspan_max_sigma_vm] = max(cross_section_max_sigma_vm_vec)
node_cross_section_max_sigma_vm = node_cross_section_max_sigma_vm_vec(node_wingspan_max_sigma_vm)

position_wing_max_sigma_vm = (node_wingspan_max_sigma_vm-1)*dx
position_cross_section_max_sigma_vm = (node_cross_section_max_sigma_vm-1)*norm(diff(section_data.x_disc(1:2,:)))

% Cross section maximum sigma Von Misses
hold on;
box on;
grid on; grid minor;
plot(section_data.x_disc(:,1),section_data.x_disc(:,2),'b-')
plot(section_data.x_disc(:,1),section_data.x_disc(:,2),'*','color','g')
plot(section_data.x_disc(node_cross_section_max_sigma_vm,1),section_data.x_disc(node_cross_section_max_sigma_vm,2),'*','color','r')
legend('Elements','Nodes','Max $\sigma_{vm}$','location','best','interpreter','latex')
title(sprintf("Most demanded node (opened, section at %.2f m)",position_wing_max_sigma_vm));
xlabel('$-z$ [m]','interpreter','latex');
ylabel("$y$ [m]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1)
%xlim([min(x(:,1))-0.1,max(x(:,1))+0.1]);
text(section_data.x_disc(1:end-1,1),section_data.x_disc(1:end-1,2),sprintfc(' %d',1:numel(section_data.x_disc(:,1))-1));
hold off;
```

A.2.2 Caso cerrado

Para el caso cerrado:

```
clear; clc; close all;

addpath('functions\')
load('section_data.mat')

g = 9.81; % [m/s^2]
% Datos geométricos
chi_s = -section_data.Xc(1); % [m] shear center caso cerrado (approx= centroide)
b = 16; % [m]
be = 0.25*b; % [m]
c = 2; % [m]
ze = 0.3*c; % [m]
za = 0.25*c; % [m]
zm = 0.48*c; % [m]
chi_p = 0.3*c; % [m]
Me = 2100; % [kg]
weight_distribution = 140*g; % [N/m]

vinf = 750/3.6; % [m/s]
rho = 1.225; % [kg/m^3]
Cl = 0.1;
lift_func = @(x)0.5*rho*vinf^2*c*Cl*sqrt(1-(x/b)^2);

% Geometría y propiedades
dx = b/512;
x = 0:dx:b;
data.x = x;
data.nnod = numel(x);
```

```
data.nel = numel(x)-1;
data.nne = 2;
data.ni = 3;
data.ndof = data.nnod*data.ni;  % Total number of degrees of freedom
Tn = zeros(data.nel,2);
for e=1:data.nel
    Tn(e,:) = [e, e+1];
end
Td = connectDOF(data,Tn);

p = [1 1 0 % no desplazamiento vertical en el empotramiento
     1 2 0 % no giro en el empotramiento
     1 3 0]; % no giro de torsion en el empotramiento

% Datos del material
E = 210e9;
G = 80e9;
I = section_data.Ixx; % [m^4] Inercia en el eje x local de la sección transversal (el eje z global del dibujo)
J = section_data.J_closed; % [m^4] Módulo de torsión (polar inertia) (caso abierto)
m = [E G I J];
Tm = ones(data.nel,1); % Toda el ala presenta la misma sección con mismo material


% Fuerza distribuida equivalente resultante en cada ELEMENTO
distributed_lift_nodes = arrayfun(lift_func,x);
distributed_lift_elements = (distributed_lift_nodes(1:end-1) + distributed_lift_nodes(2:end))/2;
distributed_load = distributed_lift_elements - weight_distribution;

% Momento torsor distribuido equivalente resultante en cada ELEMENTO
distributed_torsion_lift = (chi_s+chi_p-za)*distributed_lift_elements;
distributed_torsion = distributed_torsion_lift - (chi_s+chi_p-zm)*weight_distribution;

% Fuerzas puntuales y momentos/torsores en cada NODO
motor_node_position = find(x==be);
F = [motor_node_position 1 -Me*g
     motor_node_position 3 -Me*g*(chi_s+chi_p-ze)]; % Point loads (only the engine)


%% B) Beam analysis (closed)
[u,r, x_el,S_el,Mb_el,Mt_el] = solveStructure(data,Tn,Td,m,Tm,p,distributed_load,distributed_torsion,F);

u_3rows = reshape(u,3,[]);


%% Gràfics

% b.1)
figure('Name','Closed section');
t = tiledlayout(3,1);
title(t,'\textbf{Closed section}','interpreter','latex')

% Vertical deflection (closed section)
ax1 = nexttile;
plot(ax1,x,u_3rows(1,:)*100,'b-')
title(ax1,'\textbf{Vertical deflection}','interpreter','latex');
ylabel("$y$ [cm]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;
%daspect([1 4 .5])

% Bending rotation (closed section)
ax2 = nexttile;
plot(ax2,x,u_3rows(2,:)*180/pi,'g-')
title(ax2,'\textbf{Bending rotation}','interpreter','latex');
ylabel("$\theta_z$ [deg]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Torsion rotation (closed section)
ax3 = nexttile;
plot(ax3,x,u_3rows(3,:)*180/pi,'r-')
title(ax3,'\textbf{Torsion rotation}','interpreter','latex');
ylabel("$\theta_x$ [deg]",'Interpreter','latex');
xlabel('Wingspan position [m]','interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

%

figure('Name','Closed section');
t = tiledlayout(3,1);
title(t,'\textbf{Closed section}','interpreter','latex')

% Shear force (closed section)
ax1 = nexttile;
plot(ax1,x_el,S_el/1000,'b-')
title(ax1,'\textbf{Shear force}','interpreter','latex');
ylabel("$S_y$ [kN]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;
%daspect([1 4 .5])

% Bending moment (closed section)
ax2 = nexttile;
plot(ax2,x_el,Mb_el/1000,'g-')
title(ax2,'\textbf{Bending moment}','interpreter','latex');
ylabel("$M_z$ [kN\,m]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;

% Torsion moment (closed section)
ax3 = nexttile;
plot(ax3,x_el,Mt_el/1000,'r-')
title(ax3,'\textbf{Torsion moment}','interpreter','latex');
ylabel("$M_x$ [kN\,m]",'Interpreter','latex');
xlabel('Wingspan position [m]','interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1);
grid minor, grid on;


```

```
%% B2 Study of convergence

[discretization_factor, tip_deflection_error, tip_torsion_error] = getTipDeflectoinAndTorsionClosedError(512);

% Gráfico de la convergencia
hold on;
box on;
grid on; grid minor;
plot(discretization_factor,tip_deflection_error*100,'bx:','LineWidth',1)
plot(discretization_factor,tip_torsion_error*100,'mx:','LineWidth',1)
set(gca,'xscale','log')

ticks = [1, 2, 4, 8, 16, 32, 64, 128];
tickLabels = cellstr(num2str(ticks'));
set(gca, 'XTick', ticks, 'XTickLabel', tickLabels);

legend('Deflection','Twist','location','best','interpreter','latex')
title("\textbf{Study of Convergence (closed)}",'interpreter','latex');
xlabel('Discretization factor ($n$) [$h_{el} = b/n$]','interpreter','latex');
ylabel("Relative error (\%)",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1)
hold off;



%% C) Most critical point (Von Misses)

aplana = @(vec)[vec(1,1) vec(1,2:end) vec(2,end)];

cross_section_max_sigma_vm_vec = zeros(1,data.nel);
node_cross_section_max_sigma_vm_vec = zeros(1,data.nel);
for i=1:data.nel
    [~,tau,~] = getTangentialStressDistributionClosed(section_data.x_disc,section_data.Tn,section_data.m,section_data.Tm,section_data.Xc,
      section_data.Ixx,section_data.Iyy,section_data.Ixy,0,S_el(1,i),section_data.Xc,section_data.Ain);
    [tau_t,~] = getTangentialStressDistributionTorsionClosed(section_data.x_disc,section_data.Tn,section_data.m,section_data.Tm,-Mt_el(1,i),
      section_data.Ain);
    [sigma,~] = getNormalStressDistribution(section_data.x_disc,section_data.Tn,section_data.Xc,section_data.Ixx,section_data.Iyy,section_data
      .Ixy,Mb_el(1,i),0);
    tau = aplana(tau); % tau de cada nodo (cada posición es el nodo correspondiente de la discretización de la sección transversal)
    tau_t = aplana(tau_t); % tau_t de cada nodo (cada posición es el nodo correspondiente de la discretización de la sección transversal)
    sigma = aplana(sigma); % sigma de cada nodo (cada posición es el nodo correspondiente de la discretización de la sección transversal)

    sigmas_vm = sqrt(sigma.^2+3*(tau+tau_t).^2);
    [cross_section_max_sigma_vm_vec(i), node_cross_section_max_sigma_vm_vec(i)] = max(sigmas_vm);
end

[wingspan_max_sigma_vm, node_wingspan_max_sigma_vm] = max(cross_section_max_sigma_vm_vec)
node_cross_section_max_sigma_vm = node_cross_section_max_sigma_vm_vec(node_wingspan_max_sigma_vm)

position_wing_max_sigma_vm = (node_wingspan_max_sigma_vm-1)*dx
position_cross_section_max_sigma_vm = (node_cross_section_max_sigma_vm-1)*norm(diff(section_data.x_disc(1:2,:)))

% Cross section maximum sigma Von Misses
hold on;
box on;
grid on; grid minor;
plot(section_data.x_disc(:,1),section_data.x_disc(:,2),'b-')
plot(section_data.x_disc(:,1),section_data.x_disc(:,2),'*','color','g')
plot(section_data.x_disc(node_cross_section_max_sigma_vm,1),section_data.x_disc(node_cross_section_max_sigma_vm,2),'*','color','r')
legend('Elements','Nodes','Max $\sigma_{vm}$','location','best','interpreter','latex')
title(sprintf("Most demanded node (closed, section at %.2f m)",position_wing_max_sigma_vm));
xlabel('$-z$ [m]','interpreter','latex');
ylabel("$y$ [m]",'Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','XLim',xlim*1,'YLim',ylim*1)
%xlim([min(x(:,1))-0.1,max(x(:,1))+0.1]);
text(section_data.x_disc(1:end-1,1),section_data.x_disc(1:end-1,2),sprintfc(' %d',1:numel(section_data.x_disc(:,1))-1));
hold off;
```

Véanse las funciones y todo el código en el siguiente enlace.