

ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS LILLE

Aile de planeur

PJT

Élèves :

KYLIAN BREUVART
LÉO POUILLY

Enseignant :

AURÉLIEN GROLET



Table des matières

I	Introduction	3
II	Objectifs	3
III	Détermination de l'épaisseur e	4
1	Modèle de poutre	4
2	Équations d'équilibre	4
3	Effort interne	4
4	Équation d'équilibre	4
5	Calcul de l'épaisseur e	5
6	Moment quadratique	5
7	Calculs des contraintes au sol et en vol	5
8	Analyse asymptotique	5
9	Comparaison des matériaux candidats	9
IV	Modélisation par éléments finis	10
1	Modèle 1 : Poutre à section elliptique	10
1.1	Résultats visuels des simulations Abaqus	10
2	Modèle 2 : Ellipse surfacique extrudée	11
2.1	Résultats visuels des simulations Abaqus	11
3	Modèle 3 : Profil réel en surfacique de l'aile	13
3.1	Limites de la modélisation : absence de simulation du profil réel	13
4	Comparaison des modèles établis	14
5	Proposition de solution technologique	14
V	Dimensionnement du longeron	15
1	Étude paramétrique des sections de longeron	15
2	Résultats pour deux configurations géométriques du longeron	20
VI	Vérification dans Abaqus : Modèle de Poutre	23
1	Longeron à section rectangulaire	23
2	Longeron à section circulaire	24
3	Tableau récapitulatif	25
VII	Conclusion	26

Table des figures

III.1	Évolution de les différents critères en fonction de l'épaisseur	9
IV.1	Modèle surfacique du profil d'aile	13
V.1	Zone pour un profil rectangulaire	21
V.2	Zone pour un profil circulaire	22

Liste des tableaux

1	Comparaison des matériaux en termes de masse et de coût	9
2	Illustrations des simulations Abaqus avec flèche et contrainte maximale . .	10
3	Résumé des résultats pour le modèle 1	11
4	Illustrations des simulations Abaqus avec flèche et contrainte maximale . .	11
5	Résumé des résultats pour le modèle 2	12
6	Comparaison des résultats théoriques et numériques (Abaqus) en condi- tions sol et vol	14
7	Comparaison des performances entre deux sections de longeron	20
8	Illustrations des simulations Abaqus avec flèche et contrainte maximale . .	23
9	Illustrations des simulations Abaqus avec flèche et contrainte maximale . .	24
10	Comparaison des résultats théoriques et numériques (Abaqus) en condi- tions sol et vol	25

Listings

1	code Python pour l'évaluation de l'épaisseur	6
2	code Python pour le couple (a	15
3	code Python pour le couple (R	17

I Introduction

Le dimensionnement mécanique des ailes d'un planeur est une étape cruciale dans la conception aéronautique. Il permet de garantir la sécurité, la performance et l'efficacité du planeur tout en respectant les contraintes géométriques, mécaniques, de masse et financières. Ce rapport vise à détailler les différentes étapes du dimensionnement mécanique d'une aile de planeur, en se concentrant particulièrement sur le choix de l'épaisseur et du matériau de l'aile.

II Objectifs

Les objectifs de ce rapport sont les suivants :

1. **Analyser les contraintes géométriques et mécaniques** : Comprendre les limitations imposées par l'envergure, la flèche au sol, la contrainte maximale et la masse du planeur.
2. **Déterminer l'épaisseur optimale de l'aile** : Utiliser les modèles de poutre et les équations d'équilibre pour calculer l'épaisseur e qui respecte les contraintes du cahier des charges.
3. **Évaluer les performances mécaniques** : Calculer la flèche, la contrainte maximale et la masse de l'aile en fonction de l'épaisseur e .
4. **Choisir le matériau approprié** : Comparer différents matériaux en fonction de leur coût, de leur densité et de leur résistance mécanique.
5. **Dimensionner le longeron en évaluant les performances mécaniques** : On fait l'hypothèse qu'il ne dépend que de deux paramètres :
 - (a) circulaire creuse (rayon R et épaisseur e)
 - (b) rectangulaire (hauteur b et largeur a)

III Détermination de l'épaisseur e

1 Modèle de poutre

Pour modéliser l'aile du planeur, nous utilisons le modèle de poutre d'Euler-Bernoulli. Ce modèle repose sur les hypothèses suivantes :

- Une des dimensions (la longueur L) est grandement supérieure aux deux autres (l'épaisseur e et la hauteur h).
- Les sections droites sont supposées rigides et restent perpendiculaires à la ligne moyenne.
- Les hypothèses des petites perturbations sont appliquées, ce qui permet de linéariser les équations.

2 Équations d'équilibre

Nous travaillons sur une seule aile par symétrie. Les équations d'équilibre global sont les suivantes :

- Équilibre des forces verticales :

$$Y_O + \int q \, dx = 0 \Rightarrow Y_0 = -qL$$

- Équilibre des moments :

$$M_0 + \int xq \, dx = 0 \Rightarrow M_0 = -\frac{qL^2}{2}$$

3 Effort interne

Les efforts internes sont calculés comme suit :

- Effort tranchant $T(x)$:

$$T(x) = \int_x^L q \, dx = q(L - x)$$

- Moment fléchissant $M(x)$:

$$M(x) = \int_x^L (\xi - x)q \, d\xi = \frac{q}{2}(x - L)^2$$

4 Équation d'équilibre

L'équation d'équilibre pour la flèche $v(x)$ est donnée par :

$$EIv'' = M(x) = \frac{q}{2}(x - L)^2$$

Avec les conditions aux limites $v(0) = v'(0) = 0$, nous obtenons :

$$v(x) = \frac{qx^2}{24EI} (x^2 - 4Lx + 6L^2)$$

La flèche maximale v_{\max} est donnée par :

$$v_{\max} = v(L) = \frac{qL^4}{8EI}$$

5 Calcul de l'épaisseur e

Pour déterminer l'épaisseur e , nous devons respecter les contraintes suivantes :

— **Flèche maximale :**

$$v(L) = \frac{qL^4}{8EI} < 30 \text{ cm}$$

— **Contrainte maximale :**

$$\sigma_{\max} = \frac{qL^2}{2I} y_{\max} < \sigma_e$$

— **Masse de l'aile :**

$$M_a = \rho SL < 150 \text{ kg}$$

6 Moment quadratique

Pour une section elliptique creuse, le moment quadratique I est donné par :

$$I_{\text{creux}} \approx \frac{\pi}{4} (3eb^2 + eb^3)$$

Pour une poutre à section elliptique le moment quadratique I est donné par :

$$I_{\text{plein}} = \frac{\pi}{4} \cdot a \cdot b^3$$

7 Calculs des contraintes au sol et en vol

D'une part, les calculs ont été réalisées en faisant bien la différence lors d'une ellipse creuse ou pleine que ce soit pour les calculs de sections ou de moment :

$$S_{\text{ellipse pleine}} = \pi ab$$

$$S_{\text{ellipse creuse}} = \pi ab - \pi(a-e)(b-e) \approx \pi(a+b)e$$

De plus, les notre aile n'est pas soumis au même effort linéaire au sol ou en vol :

au sol : $q = -\rho g S$

en vol : $q = p - \rho g S = \frac{M_{ng}}{2L}$

8 Analyse asymptotique

Pour des valeurs petites de e , nous pouvons effectuer une analyse asymptotique :

— **Contrainte :**

$$\begin{aligned} \sigma_{\max} &= \\ \sigma_{\max} &= 2\rho g \frac{(a+b)L^2}{b(3a+b)} = \text{cste, lorsque } e \rightarrow 0 \end{aligned}$$

$$\sigma_{\max} \rightarrow 2\rho g \frac{(a+b)L^2}{b(3a+b)} = \text{cste, lorsque } e \rightarrow 0$$

— **Masse :**

$$M_a \approx \rho L \pi (a+b)e = \alpha e$$

— Flèche :

$$v(L) \rightarrow 4\rho g(a+b) \frac{L^4}{8E(b^2(3a+b))} = \text{cste}, \text{ lorsque } e \rightarrow 0$$

En vol, la contrainte et la flèche évoluent comme $\frac{K}{e}$ lorsque e tend vers zéro.

Afin de déterminer les grandeurs qui nous intéressent nous avons mis en place un programme Python.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import pi
4
5 # Cahier des charges
6 a = 0.625 # demi-grand axe (m)
7 b = 0.2 # demi-petit axe (m)
8 L = 11.0 # longueur aile (m)
9 g = 9.81
10
11 M_tot = 500
12 M_n = 200 #nacelle +pilote(kg)
13 v_max_seuil = 0.3 # m
14 masse_max = 150 # kg
15
16 # Liste des pâisseurs
17 e_list = np.logspace(-6, -2, 500)
18
19 # -----
20 # ===== 1. ALUMINIUM 7075-T6 =====
21 E_alu = 71.7e9
22 rho_alu = 2810
23 sigma_e_alu = 505e6
24
25 masse_alu = []
26 sigma_max_alu = []
27 fleche_alu = []
28
29 for e in e_list:
30     S = pi * (a + b) * e
31     I = (pi / 4) * b**2 * e * (3 * a + b)
32     M_a = rho_alu * S * L
33     q_sol = -rho_alu * S * g
34     p = M_tot * g / (2 * L)
35     q_vol = p - rho_alu * S * g
36
37     sigma = max(
38         abs((q_sol * L**2 / (2 * I)) * b),
39         abs((q_vol * L**2 / (2 * I)) * b)
40     )
41     fleche = max(
42         abs((q_sol * L**4) / (8 * E_alu * I)),
43         abs((q_vol * L**4) / (8 * E_alu * I))
44     )
45
46     masse_alu.append(M_a)
47     sigma_max_alu.append(sigma)

```

```

48     fleche_alu.append(fleche)
49
50 # -----
51 # === 2. ACIER 25CD4 ===
52 E_acier = 210e9
53 rho_acier = 7850
54 sigma_e_acier = 700e6
55
56 masse_acier = []
57 sigma_max_acier = []
58 fleche_acier = []
59
60 for e in e_list:
61     S = pi * (a + b) * e
62     I = (pi / 4) * b**2 * e * (3 * a + b)
63     M_a = rho_acier * S * L
64     q_sol = -rho_acier * S * g
65     p = M_tot * g / (2 * L)
66     q_vol = p - rho_acier * S * g
67
68     sigma = max(
69         abs((q_sol * L**2 / (2 * I)) * b),
70         abs((q_vol * L**2 / (2 * I)) * b)
71     )
72     fleche = max(
73         abs((q_sol * L**4) / (8 * E_acier * I)),
74         abs((q_vol * L**4) / (8 * E_acier * I))
75     )
76
77     masse_acier.append(M_a)
78     sigma_max_acier.append(sigma)
79     fleche_acier.append(fleche)
80
81 # -----
82 # === 3. COMPOSITE CARBONE/EPOXY ===
83 E_compo = 135e9
84 rho_compo = 1550
85 sigma_e_compo = 1200e6
86
87 masse_compo = []
88 sigma_max_compo = []
89 fleche_compo = []
90
91 for e in e_list:
92     S = pi * (a + b) * e
93     I = (pi / 4) * b**2 * e * (3 * a + b)
94     M_a = rho_compo * S * L
95     q_sol = -rho_compo * S * g
96     p = M_tot * g / (2 * L)
97     q_vol = p - rho_compo * S * g
98
99     sigma = max(
100        abs((q_sol * L**2 / (2 * I)) * b),
101        abs((q_vol * L**2 / (2 * I)) * b)
102    )
103    fleche = max(
104        abs((q_sol * L**4) / (8 * E_compo * I)),

```

```

105         abs((q_vol * L**4) / (8 * E_compo * I))
106     )
107
108     masse_compo.append(M_a)
109     sigma_max_compo.append(sigma)
110     fleche_compo.append(fleche)
111
112
113
114 #calcul de S et I
115
116
117 e_test = 0.0007 # e = 0.7mm
118 S_test = pi * (a + b) * e_test
119 I_test = (pi / 4) * b**2 * e_test * (3 * a + b)
120 print(f"S (e=0,7mm) = {S_test:.6f} m , I (e=0,7mm) = {I_test:.6e} m")
121
122 # Trac s
123
124 plt.figure(figsize=(10, 12))
125
126 # 1. Contrainte max
127 plt.subplot(3, 1, 1)
128 plt.semilogx(e_list, np.array(sigma_max_alu)/1e6, label="Alu 7075-T6")
129 plt.semilogx(e_list, np.array(sigma_max_acier)/1e6, label="Acier 25CD4")
130 plt.semilogx(e_list, np.array(sigma_max_compo)/1e6, label="Composite
        carbone/ poxy ")
131 plt.axhline(900, color='r', linestyle='--', label="Seuil max = 900 MPa")
132 plt.ylabel("Contrainte max (MPa)")
133 plt.title("Contrainte maximale vs   paissanceur ")
134 plt.grid(True)
135 plt.legend()
136
137 # 2. Fl che
138 plt.subplot(3, 1, 2)
139 plt.semilogx(e_list, np.array(fleche_alu)*100, label="Alu 7075-T6")
140 plt.semilogx(e_list, np.array(fleche_acier)*100, label="Acier 25CD4")
141 plt.semilogx(e_list, np.array(fleche_compo)*100, label="Composite
        carbone/ poxy ")
142 plt.axhline(v_max_seuil*100, color='r', linestyle='--', label="Fl che
        max = 30 cm")
143 plt.ylabel("Fl che (cm)")
144 plt.title("Fl che maximale vs   paissanceur ")
145 plt.grid(True)
146 plt.legend()
147
148 # 3. Masse
149 plt.subplot(3, 1, 3)
150 plt.semilogx(e_list, masse_alu, label="Alu 7075-T6")
151 plt.semilogx(e_list, masse_acier, label="Acier 25CD4")
152 plt.semilogx(e_list, masse_compo, label="Composite carbone/ poxy ")
153 plt.axhline(masse_max, color='r', linestyle='--', label="Masse max = 150
        kg")
154 plt.ylabel("Masse aile (kg)")
155 plt.xlabel(" paissanceur e (m)")
156 plt.title("Masse de l aile vs   paissanceur ")

```

```

157 plt.grid(True)
158 plt.legend()
159
160 plt.tight_layout()
161 plt.show()

```

Listing 1 – code Python pour l'évaluation de l'épaisseur

Nous avons dû considérer une épaisseur e d'aile qui puisse répondre aux critères de masse, de flèche maximale et de contrainte maximale selon 3 types de matériaux : Aluminium 7075-T6, acier 25D4 et du composite carbone/époxy.

Ces 3 matériaux sont adaptés pour concevoir une aile de planeur mais une autre contrainte à prendre en compte est le prix.

9 Comparaison des matériaux candidats

Matériau	Masse équivalente	Coût estimé
Aluminium 7075-T6	100,0 kg	1500 €
Acier 25CD4	95,4 kg	382 €
Composite carbone/époxy	29,3 kg	2930 €

TABLE 1 – Comparaison des matériaux en termes de masse et de coût

Voici une représentation graphique des valeurs de l'évolution des différents critères du cahier des charges en fonction de l'épaisseur.

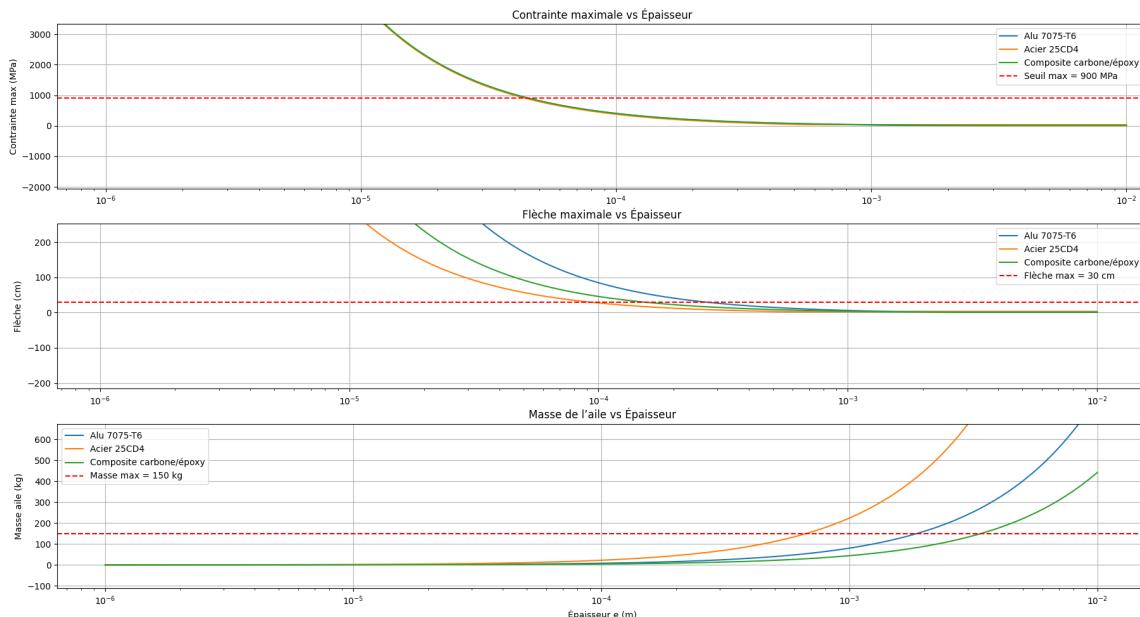


FIGURE III.1 – Évolution de les différents critères en fonction de l'épaisseur

De ce fait, nous avons choisi une épaisseur de 0.7 mm et de l'aluminium 7075-T6 pour concevoir notre aile de planeur.

IV Modélisation par éléments finis

Dans cette partie nous allons comparer 3 modèles : modèle de poutre à section elliptique, modèle surfacique d'une ellipse extrudée et un modèle surfacique du profil réel de notre aile de planeur.

Cahier des charges :

- $M = 500 \text{ kg}$: la masse du planeur
- $V = 80 \text{ km/h}$: la vitesse de vol du planeur
- $L = 22 \text{ m}$: l'envergure totale du planeur
- $c = 1,25 \text{ m}$: la corde

1 Modèle 1 : Poutre à section elliptique

1.1 Résultats visuels des simulations Abaqus

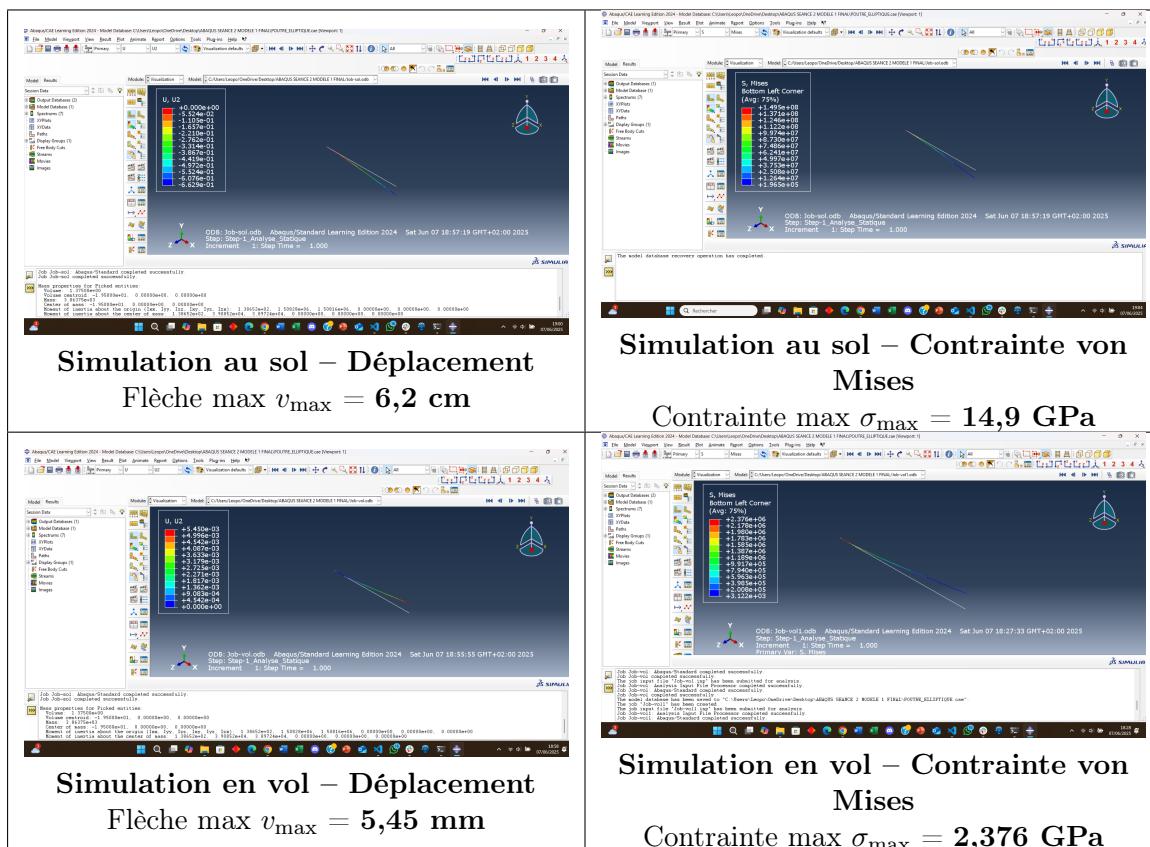


TABLE 2 – Illustrations des simulations Abaqus avec flèche et contrainte maximale

Grandeur	Théorique	Abaqus	Respect du cahier des charges
Au sol			
Flèche max v_{\max}	7 cm	6,2 cm	Oui
Contrainte max σ_{\max}	17 GPa	14,9 GPa	Oui
Masse	12 T	386,375 kg	Non (différence importante)
En vol			
Flèche max v_{\max}	0,6 mm	5,45 mm	Non (flèche trop élevée)
Contrainte max σ_{\max}	0,55 GPa	2,376 GPa	Non (contrainte trop élevée)
Masse	12 T	386,375 kg	Non (différence importante)

TABLE 3 – Résumé des résultats pour le modèle 1

2 Modèle 2 : Ellipse surfacique extrudée

2.1 Résultats visuels des simulations Abaqus

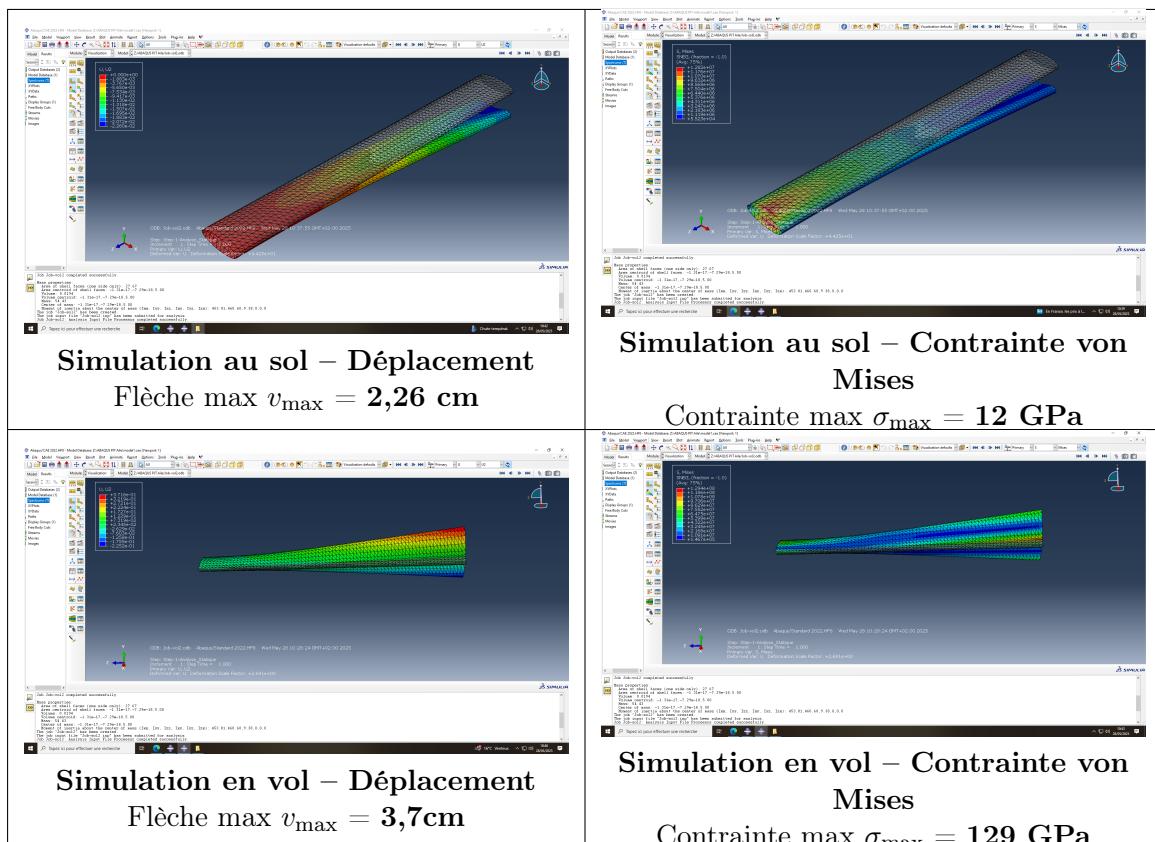


TABLE 4 – Illustrations des simulations Abaqus avec flèche et contrainte maximale

Grandeur	Théorique	Abaqus	Respect du cahier des charges
Au sol			
Flèche max v_{\max}	2,7 cm	2,26 cm	Oui
Contrainte max σ_{\max}	13,3 GPa	12 GPa	Oui
Masse	54,43 kg	54,43 kg	Oui
En vol			
Flèche max v_{\max}	7,08 cm	3,7 cm	Oui
Contrainte max σ_{\max}	2,1 GPa	129 GPa	Non (surcontrainte)
Masse	54,43 kg	54,43 kg	Oui

TABLE 5 – Résumé des résultats pour le modèle 2

Les deux modèles présentés ont pour objectif de simuler le comportement structurel d'une aile de planeur sous différentes conditions (au sol et en vol), tout en respectant les contraintes imposées par le cahier des charges.

Le **Modèle 1**, qui repose sur une poutre à section elliptique, montre une bonne concordance entre les résultats théoriques et ceux issus du modèle numérique Abaqus pour les cas statiques au sol, avec une flèche et une contrainte maximales inférieures aux limites admissibles. Toutefois, sa masse estimée (12 tonnes) est irréaliste et largement supérieure à la masse réelle attendue (environ 386 kg selon Abaqus), ce qui invalide sa pertinence. En vol, le modèle devient également non conforme : les contraintes et la flèche sont trop élevées, ce qui rend ce modèle inadapté pour représenter fidèlement la réalité.

Le **Modèle 2**, basé sur une extrusion surfacique d'une ellipse, se distingue par une meilleure cohérence entre les aspects théoriques et numériques. Il respecte globalement le cahier des charges, tant au niveau de la masse que de la déformation en vol et au sol. Toutefois, une surcontrainte importante est observée en vol dans le modèle Abaqus (12,9 GPa contre 2,1 GPa théorique), ce qui suggère que la distribution réelle des efforts dans l'aile est plus complexe que celle modélisée analytiquement.

En conclusion, le modèle 2 est globalement plus réaliste et fidèle aux conditions physiques attendues. Malgré la surcontrainte observée en vol, il reste plus pertinent que le modèle 1 pour représenter le comportement d'une aile de planeur dans son ensemble.

3 Modèle 3 : Profil réel en surfacique de l'aile

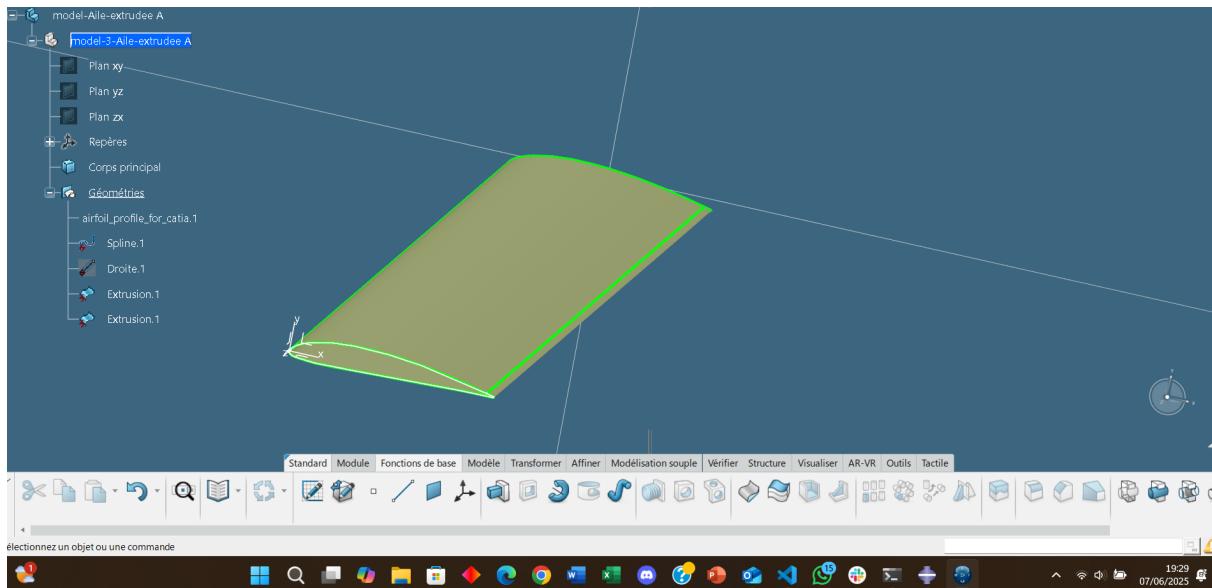


FIGURE IV.1 – Modèle surfacique du profil d'aile

3.1 Limites de la modélisation : absence de simulation du profil réel

Dans le cadre de cette étude, nous n'avons pas été en capacité de réaliser une simulation Abaqus du profil réel de l'aile du planeur. Cette limitation est principalement due à la complexité géométrique du profil, ainsi qu'à la nécessité d'un maillage plus fin et plus précis, qui requiert des ressources de calcul importantes et une préparation numérique rigoureuse que nous n'avons pas réussi à mettre en place.

La réalisation d'une telle simulation aurait permis de confronter directement la théorie aux résultats numériques sur la géométrie réelle, apportant ainsi une validation plus fidèle du comportement mécanique de l'aile. On aurait pu observer, par exemple, une distribution plus réaliste des contraintes dans l'épaisseur du profil, notamment au niveau des zones de transition entre l'extrados et l'intrados. Théoriquement, les résultats attendus auraient été intermédiaires entre les modèles simplifiés (poutre elliptique ou extrusion d'ellipse) et la réalité, avec des flèches et des contraintes mieux localisées, et une répartition des efforts plus homogène.

Cette modélisation aurait également permis d'évaluer plus finement les zones critiques de l'aile, notamment au niveau des points de fixation ou des renforts internes. En l'absence de cette simulation, l'analyse reste limitée aux modèles idéalisés, qui ne capturent qu'une partie des phénomènes mécaniques réels.

4 Comparaison des modèles établis

Modèle	État	Grandeur	Théorique	Abaqus	Écart
Modèle 1	Sol	Flèche max v_{\max}	7 cm	6,2cm	-
		Contrainte max σ_{\max}	17 GPa	14,9 GPa	-
		Masse	12 T	386,375 kg	-
	Vol	Flèche max v_{\max}	0,6 mm	5,45mm	-
		Contrainte max σ_{\max}	0,55 GPa	2,376 GPa	-
		Masse	12T	386,375 kg	-
Modèle 2	Sol	Flèche max v_{\max}	2,7 cm	2,26 cm	-
		Contrainte max σ_{\max}	13,3 GPa	12 GPa	-
		Masse	54,43 kg	54,43 kg	-
	Vol	Flèche max v_{\max}	7,08 cm	3,7 cm	-
		Contrainte max σ_{\max}	2,1 GPa	129 GPa	-
		Masse	54,43 kg	54,43 kg	-

TABLE 6 – Comparaison des résultats théoriques et numériques (Abaqus) en conditions sol et vol

5 Proposition de solution technologique

Pour remédier à cette non-conformité, une solution consiste à renforcer la structure de la section en intégrant des renforts internes (ex : longerons supplémentaires) qui limitent la déformation locale de la section et garantissent le maintien de sa forme géométrique.

Alternativement, on peut adopter une modélisation plus avancée (ex : théorie des poutres de Timoshenko ou modélisation par éléments finis 3D avec couplage flexion-torsion) qui prend en compte la déformation réelle des sections, afin d'optimiser la conception sans négliger les effets complexes.

V Dimensionnement du longeron

1 Étude paramétrique des sections de longeron

On suppose que le longeron possède une section caractérisée par deux paramètres seulement, selon deux géométries possibles :

- Une section **circulaire creuse**, définie par un rayon extérieur R et une épaisseur e ;
- Une section **rectangulaire**, définie par une hauteur b et une largeur a .

Dans cette étude, on fait varier les paramètres R et e selon une échelle logarithmique, pour un matériau donné. L'objectif est d'identifier les couples (R, e) qui satisfont l'ensemble des contraintes imposées par le cahier des charges (flèche maximale, contrainte maximale, masse maximale, etc.).

Exemple pour la contrainte maximale σ_{\max} :

- On trace la surface $\sigma_{\max}(R, e)$ obtenue par le modèle mécanique ;
- On identifie la zone de validité définie par $\sigma_{\max} < \sigma_e$ (résistance maximale admissible du matériau) ;
- On répète cette démarche pour les autres critères du cahier des charges (flèche, masse...).

Chaque contrainte impose une région admissible différente dans l'espace (R, e) . Le croisement de toutes ces régions permet de déduire un domaine de conception valide. Cette synthèse est représentée sur un graphique 2D (voir figure ci-contre).

Un couple (R, e) est ensuite choisi à l'intérieur de cette zone, en fonction d'un compromis entre contraintes mécaniques et objectifs de performance (masse minimale, robustesse, etc.).

Voici les codes python qui nous ont permis de trouver nos couples :

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib.patches import Patch
4 from matplotlib.lines import Line2D
5
6 # === CONSTANTES ===
7 L = 11.0          # Longueur de l'aile (m)
8 g = 9.81          # Gravit
9 M_tot = 500       # Masse totale en vol (kg)
10 v_max_seuil = 0.3 # Flèche max (m)
11 masse_max = 150  # Masse max du longeron (kg)
12
13 # Propriétés aluminium 7075-T6
14 E = 71.7e9         # Module Young [Pa]
15 sigma_e = 503e6    # Contrainte admissible [Pa]
16 rho = 2810         # Masse volumique [kg/m3]
17
18 # === FONCTIONS DE CALCUL ===
19 def calculer_contraintes_fleche_masse_rect(b, h):
20     S = b * h
21     I = (b * h**3) / 12
22     M_a = rho * S * L
23

```

```

24     q_sol = -rho * g * S
25     p = M_tot * g / (2 * L)
26     q_vol = p - rho * g * S
27
28     sigma_sol = abs((q_sol * L**2 / (2 * I)) * (h / 2))
29     sigma_vol = abs((q_vol * L**2 / (2 * I)) * (h / 2))
30     sigma_max = max(sigma_sol, sigma_vol)
31
32     fleche_sol = abs((q_sol * L**4) / (8 * E * I))
33     fleche_vol = abs((q_vol * L**4) / (8 * E * I))
34     fleche_max = max(fleche_sol, fleche_vol)
35
36     contrainte_ok = sigma_max < sigma_e
37     fleche_ok = fleche_max < v_max_seuil
38     masse_ok = M_a < masse_max
39
40     return sigma_max, fleche_max, M_a, contrainte_ok, fleche_ok,
41     masse_ok
42
43 # === GRILLE DE PARAM TRES ===
44 b_list = np.linspace(0.005, 0.05, 200) # largeur b (m)
45 h_list = np.linspace(0.01, 0.3, 200) # hauteur h (m)
46
47 b_grid, h_grid = np.meshgrid(b_list, h_list)
48
49 # === MATRICES DE CONTRAINTES ===
50 contrainte_ok = np.zeros_like(b_grid, dtype=bool)
51 fleche_ok = np.zeros_like(b_grid, dtype=bool)
52 masse_ok = np.zeros_like(b_grid, dtype=bool)
53
54 sigma_vals = np.zeros_like(b_grid)
55 fleche_vals = np.zeros_like(b_grid)
56 masse_vals = np.zeros_like(b_grid)
57
58 for i in range(len(h_list)):
59     for j in range(len(b_list)):
60         b = b_grid[i, j]
61         h = h_grid[i, j]
62         sigma, fleche, masse, c_ok, f_ok, m_ok =
63         calculer_contraintes_fleche_masse_rect(b, h)
64
65         contrainte_ok[i, j] = c_ok
66         fleche_ok[i, j] = f_ok
67         masse_ok[i, j] = m_ok
68
69         sigma_vals[i, j] = sigma
70         fleche_vals[i, j] = fleche
71         masse_vals[i, j] = masse
72
73 # === ZONES DE DIMENSIONNEMENT ===
74 zones = np.zeros_like(b_grid, dtype=int)
75 zones[masse_ok] = 1
76 zones[masse_ok & contrainte_ok & ~fleche_ok] = 2
77 zones[masse_ok & fleche_ok & ~contrainte_ok] = 3
78 zones[masse_ok & contrainte_ok & fleche_ok] = 4
79
80 # === AFFICHAGE ===

```

```

79 plt.figure(figsize=(10, 8))
80 colors = ['black', 'red', 'yellow', 'orange', 'green']
81 labels = [
82     'Aucune contrainte respectée',
83     'Seulement masse OK',
84     'Masse + contrainte mécanique',
85     'Masse + flèche OK',
86     'Toutes contraintes OK'
87 ]
88
89 zones_masked = np.ma.masked_where(b_grid >= h_grid, zones)
90
91 im = plt.contourf(b_grid, h_grid, zones, levels=np.arange(-0.5, 5.5, 1),
92                     colors=colors, extend='neither')
93
94 plt.xlabel('Largeur a (m)', fontsize=12)
95 plt.ylabel('Hauteur b (m)', fontsize=12)
96 plt.title('Dimensionnement du longeron rectangulaire - Aluminium 7075-T6',
97             ,
98             fontsize=14, fontweight='bold')
99 plt.grid(True, alpha=0.3)
100
101
102 # === LEGENDE ===
103 legend_elements = [Patch(facecolor=colors[i], label=labels[i]) for i in
104 range(len(colors))]
105 legend_elements.append(Line2D([0], [0], marker='o', color='blue', label=
106 'Point optimal', markersize=8, linestyle='None'))
107
108 plt.legend(handles=legend_elements, loc='upper right', fontsize=10,
109             frameon=True,
110             fancybox=True, shadow=True, title="Zones et point optimal",
111             title_fontsize=11)
112
113 plt.tight_layout()
114 plt.show()
115
116 # === AFFICHAGE DES RÉSULTATS POUR LE COUPLE CHOISI ===
117 a_choisi = 0.02 # largeur (m)
118 b_choisi = 0.20 # hauteur (m)
119
120 sigma_c, fleche_c, masse_c, c_ok, f_ok, m_ok =
121     calculer_contraintes_fleche_masse_rect(a_choisi, b_choisi)
122
123 check = lambda cond: " " if cond else " "
124
125 print(f"\nPour le couple (a = {a_choisi:.2f} m, b = {b_choisi:.3f} m)
126 :")
127 print(f" - Masse de l'aile : {masse_c:.2f} kg {check(m_ok)}")
128 print(f" - Contrainte maximale : {sigma_c/1e6:.2f} MPa {check(c_ok)}")
129 print(f" - Flèche maximale : {fleche_c*100:.2f} cm {check(f_ok)}
130 ")

```

Listing 2 – code Python pour le couple (a)

```

1 import numpy as np
2 import matplotlib.pyplot as plt

```

```

3 from math import pi
4
5 # Cahier des charges
6 L = 11.0 # longueur aile (m)
7 g = 9.81
8 M_tot = 500 # masse du planeur complet inférieur 500 kg
9 M_n = 200 # masse nacelle + pilote (kg)
10 v_max_seuil = 0.3
11 masse_max = 150 # masse d'une aile qu'on ne doit pas dépasser (kg)
12
13 # Propriétés Aluminium 7075-T6
14 E_alu = 71.7e9
15 rho_alu = 2810
16 sigma_e_alu = 505e6
17
18 def calculer_contraintes_fleche_masse(R, e):
19     R_int = R - e
20     if R_int <= 0:
21         return float('inf'), float('inf'), float('inf'), False, False, False
22
23     S = pi * (R**2 - R_int**2)
24     I = (pi / 4) * (R**4 - R_int**4)
25     M_a = rho_alu * S * L
26
27     q_sol = -rho_alu * S * g
28     p = M_tot * g / (2 * L)
29     q_vol = p - rho_alu * S * g
30
31     sigma_sol = abs((q_sol * L**2 / (2 * I)) * R)
32     sigma_vol = abs((q_vol * L**2 / (2 * I)) * R)
33     sigma_max = max(sigma_sol, sigma_vol)
34
35     fleche_sol = abs((q_sol * L**4) / (8 * E_alu * I))
36     fleche_vol = abs((q_vol * L**4) / (8 * E_alu * I))
37     fleche_max = max(fleche_sol, fleche_vol)
38
39     contrainte_ok = sigma_max < sigma_e_alu
40     fleche_ok = fleche_max < v_max_seuil
41     masse_ok = M_a < masse_max
42
43     return sigma_max, fleche_max, M_a, contrainte_ok, fleche_ok, masse_ok
44
45 # Grille de paramètres
46 e_list = np.logspace(-4, -1, 200)
47 R_list = np.logspace(-2, 0, 200)
48 e_grid, R_grid = np.meshgrid(e_list, R_list)
49
50 contrainte_ok = np.zeros_like(e_grid, dtype=bool)
51 fleche_ok = np.zeros_like(e_grid, dtype=bool)
52 masse_ok = np.zeros_like(e_grid, dtype=bool)
53
54 for i in range(len(R_list)):
55     for j in range(len(e_list)):
56         R = R_grid[i, j]
57         e = e_grid[i, j]

```

```

58     if e >= R:
59         continue
60     sigma_max, fleche_max, M_a, c_ok, f_ok, m_ok =
61     calculer_contraintes_fleche_masse(R, e)
62     contrainte_ok[i, j] = c_ok
63     fleche_ok[i, j] = f_ok
64     masse_ok[i, j] = m_ok
65
66 zones = np.zeros_like(e_grid, dtype=int)
67 zones[masse_ok] = 1
68 zones[masse_ok & contrainte_ok & ~fleche_ok] = 2
69 zones[masse_ok & fleche_ok & ~contrainte_ok] = 3
70 zones[masse_ok & contrainte_ok & fleche_ok] = 4
71
72 zones_masked = np.ma.masked_where(e_grid >= R_grid, zones)
73
74 plt.figure(figsize=(10, 8))
75
76 colors = ['black', 'red', 'yellow', 'orange', 'green']
77 labels = [
78     'Aucune contrainte respect e',
79     'Seulement masse OK',
80     'Masse + contrainte m canique',
81     'Masse + fl che OK',
82     'Toutes contraintes OK'
83 ]
84
85 im = plt.contourf(e_grid, R_grid, zones_masked, levels=np.arange(-0.5,
86                     5.5, 1),
87                     colors=colors, extend='neither')
88
89 plt.xlabel(' pisseur e (m)', fontsize=12)
90 plt.ylabel('Rayon R (m)', fontsize=12)
91 plt.title('Dimensionnement du longeron - Aluminium 7075-T6\nnZones
92 autoris es (e,R)', fontweight='bold')
93 plt.xscale('log')
94 plt.yscale('log')
95 plt.grid(True, alpha=0.3)
96
97 # Ligne limite physique e = R
98 e_line = np.logspace(-4, -1, 100)
99 plt.plot(e_line, e_line, 'k--', linewidth=2, alpha=0.8, label='Limite
100 physique (e = R)')
101
102 # Ajout de lignes d'aide
103 plt.axvline(x=0.005, color='blue', linestyle='--', linewidth=1.5, label=
104     'e = 5 mm')
105 plt.axhline(y=0.10, color='blue', linestyle='--', linewidth=1.5, label=
106     'R = 10 cm')
107
108 from matplotlib.patches import Patch
109 legend_elements = [Patch(facecolor=colors[i], label=labels[i]) for i in
110     range(len(colors))]
111 zone_legend = plt.legend(handles=legend_elements, loc='upper left',
112     fontsize=10, frameon=True,
113     fancybox=True, shadow=True, title="Zones de

```

```

        dimensionnement", title_fontsize=11)
107
108 lines_legend = plt.legend(loc='lower right', fontsize=9, frameon=True,
109                         fancybox=True, shadow=True, title="Rep res",
110                         title_fontsize=10)
111 plt.gca().add_artist(zone_legend)
112 plt.tight_layout()
113 plt.show()
114
115 # === ANALYSE POUR LE COUPLE (R = 10 cm, e = 5 mm) ===
116 R_choisi = 0.10 # 10 cm
117 e_choisi = 0.005 # 5 mm
118
119 sigma_max, fleche_max, M_a, contrainte_ok, fleche_ok, masse_ok =
120     calculer_constraintes_fleche_masse(R_choisi, e_choisi)
121
122 print("== ANALYSE POUR L'ALUMINIUM 7075-T6 ==")
123 print(f"\nPour le couple (R = {R_choisi:.2f} m, e = {e_choisi:.3f} m
124 :")
125 print(f" - Masse de l'aile : {M_a:.2f} kg {'if' masse_ok
126     else ' ', }")
127 print(f" - Contrainte maximale : {sigma_max/1e6:.2f} MPa {'if'
128     contrainte_ok else ' ', }")
129 print(f" - Flèche maximale : {fleche_max*100:.2f} cm {'if'
130     fleche_ok else ' ', }")

```

Listing 3 – code Python pour le couple (R

On peut voir graphiquement une zone adéquate pour laquelle nos couples respectent tous les critères du cahier des charges. *

- pour un profil circulaire : $(R, e) = (0.1, 0.005)$ pour un profil rectangulaire : $(a, b) = (0.02, 0.2)$

2 Résultats pour deux configurations géométriques du longeron

Paramètres	Section circulaire creuse	Section rectangulaire
Dimensions	$R = 0,10 \text{ m}$, $e = 0,005 \text{ m}$	$a = 0,02 \text{ m}$, $b = 0,200 \text{ m}$
Masse de l'aile	94,68 kg	123,64 kg
Contrainte maximale	57,52 MPa	51,13 MPa
Flèche maximale	24,27 cm	21,57 cm

TABLE 7 – Comparaison des performances entre deux sections de longeron

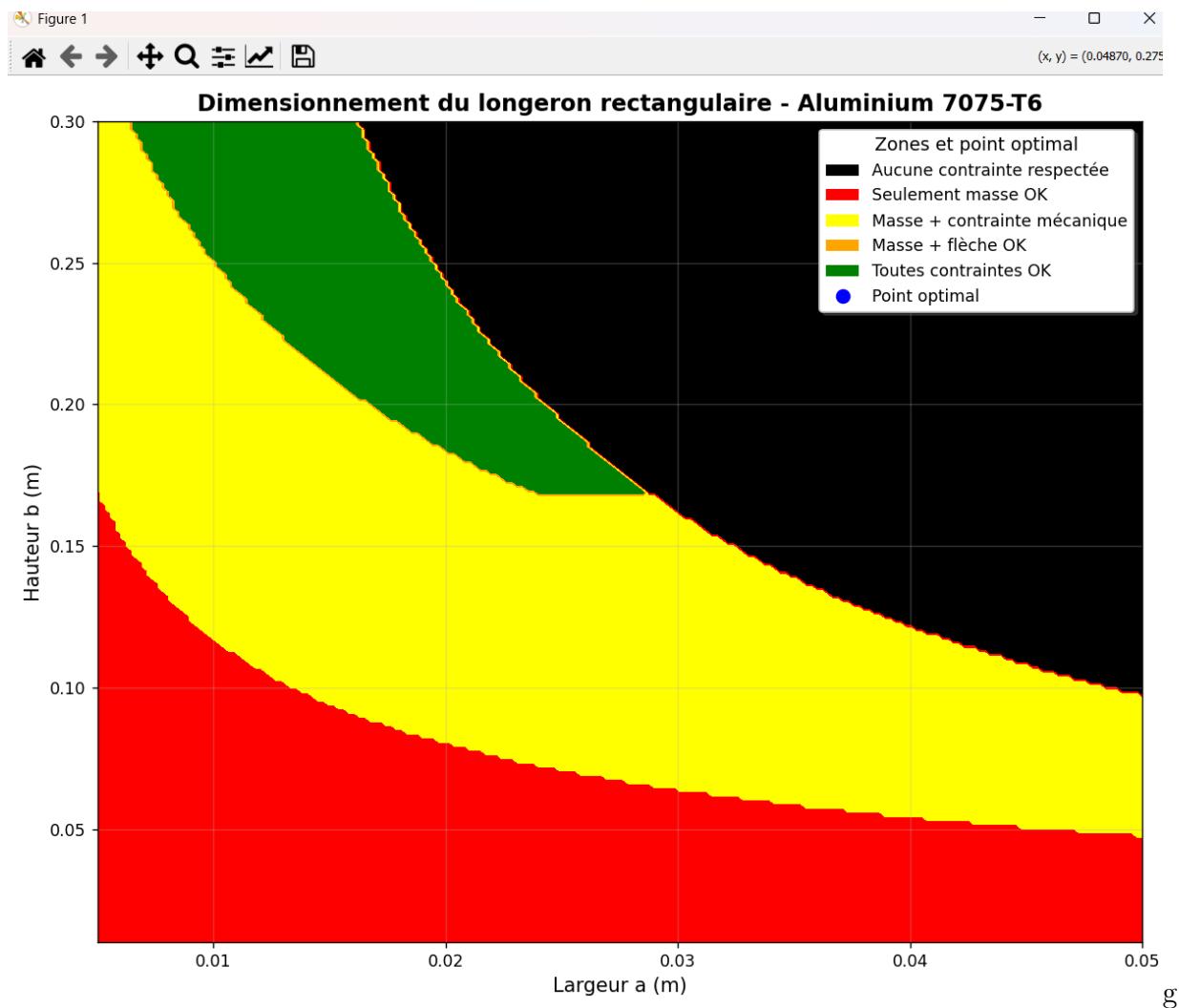


FIGURE V.1 – Zone pour un profil rectangulaire

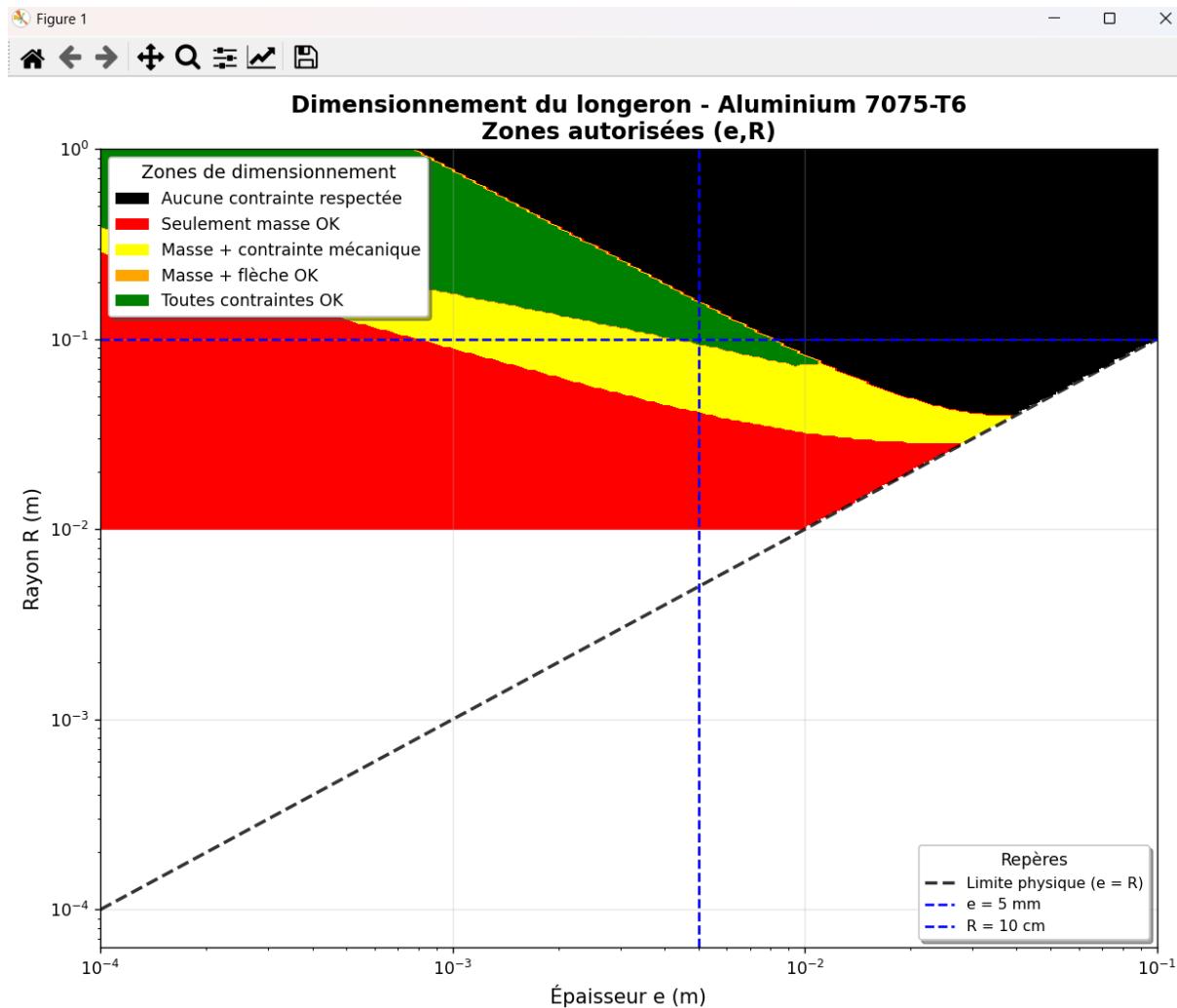


FIGURE V.2 – Zone pour un profil circulaire

VI Vérification dans Abaqus : Modèle de Poutre

Nous allons vérifier si nos calculs sont cohérents avec le modèle de poutre.

1 Longeron à section rectangulaire

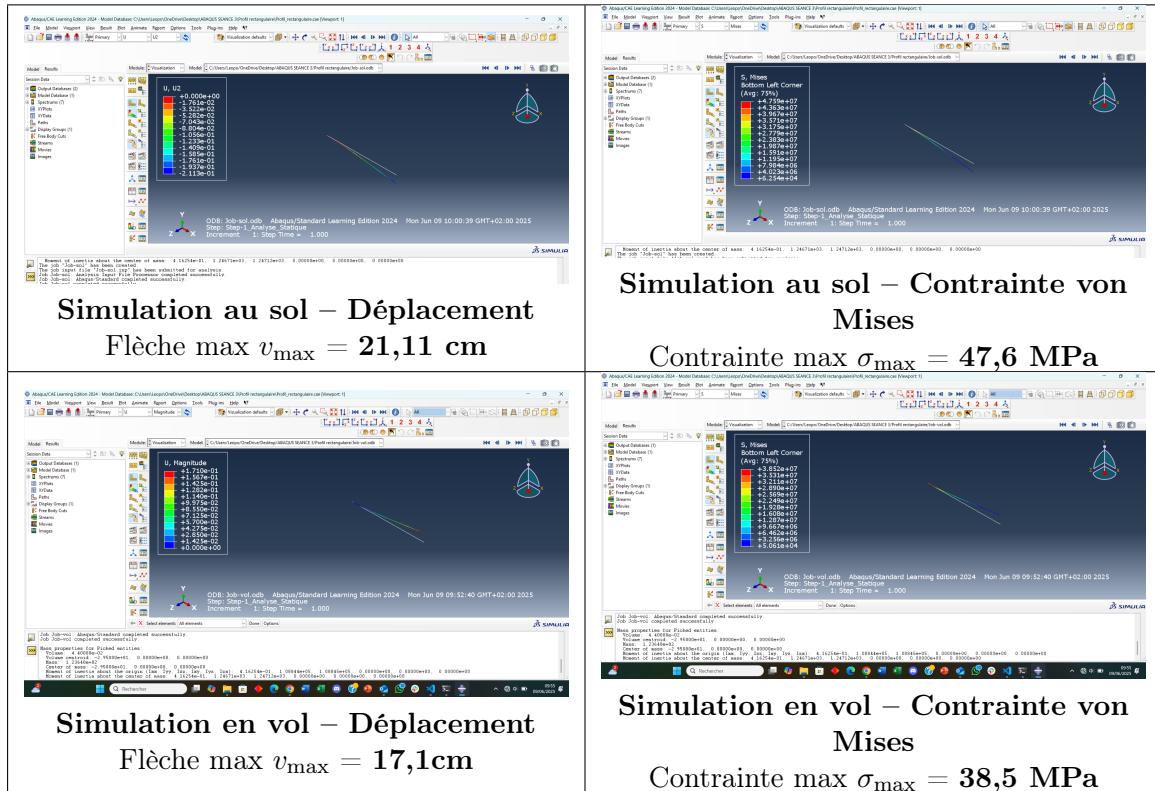


TABLE 8 – Illustrations des simulations Abaqus avec flèche et contrainte maximale

2 Longeron à section circulaire

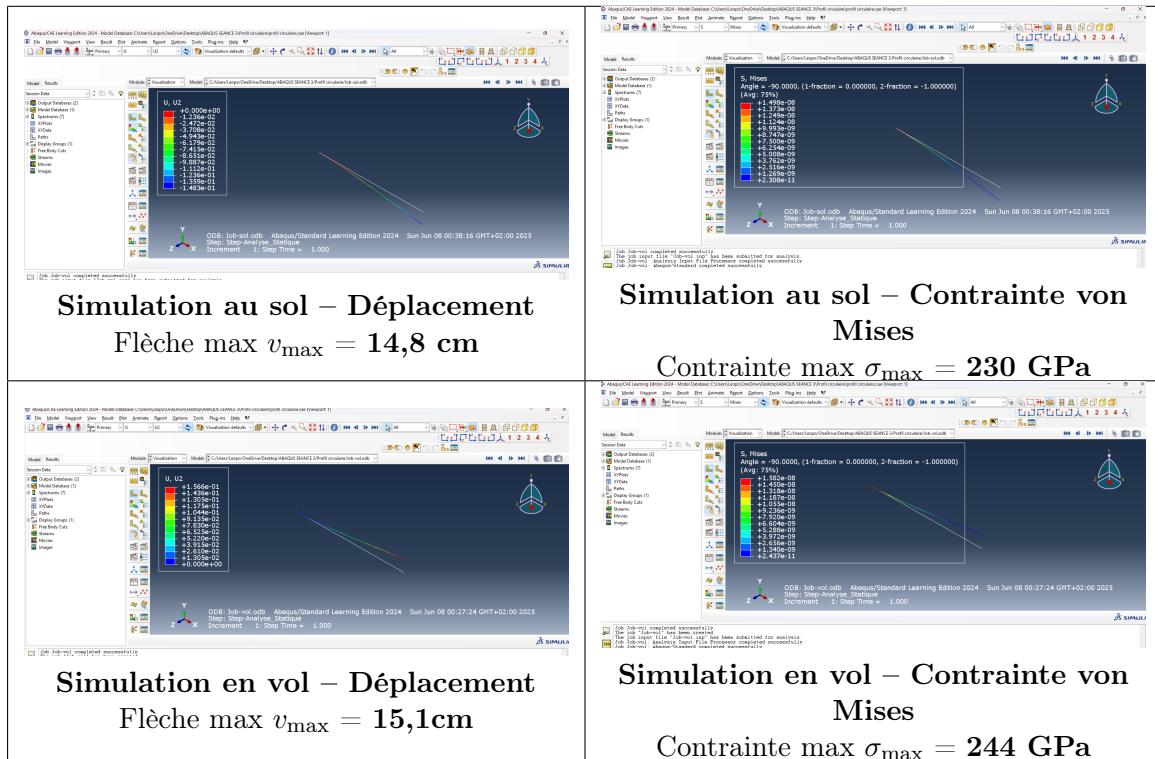


TABLE 9 – Illustrations des simulations Abaqus avec flèche et contrainte maximale

3 Tableau récapitulatif

Modèle	État	Grandeur	Théorique	Abaqus	Écart
Section rectangulaire	Sol	Flèche max v_{\max}	21,1 cm	21,11 cm	-
		Contrainte max σ_{\max}	50 MPa	47,6 MPa	-
		Masse	123,64 kg	123,64 kg	-
	Vol	Flèche max v_{\max}	21,6mm	17,1cm	-
		Contrainte max σ_{\max}	51,1 MPa	38,5 MPa	-
		Masse	123,64 kg	123,64 kg	-
Section circulaire	Sol	Flèche max v_{\max}	14,79 cm	14,8 cm	-
		Contrainte max σ_{\max}	35 MPa	230 MPa	-
		Masse	94,68 kg	94,68 kg	-
	Vol	Flèche max v_{\max}	24,27cm	15,1 cm	-
		Contrainte max σ_{\max}	57,52 MPa	244 GPa	-
		Masse	94,68 kg	94,68 kg	-

TABLE 10 – Comparaison des résultats théoriques et numériques (Abaqus) en conditions sol et vol

Analyse des résultats : En analysant les résultats des simulations numériques réalisées sous Abaqus et en les comparant aux valeurs théoriques, nous pouvons les confronter aux exigences du cahier des charges. Une contrainte importante est que la **flèche maximale admissible est de 50 cm**, soit $v_{\max} \leq 0,5 \text{ m}$. De plus, les contraintes mécaniques ne doivent pas dépasser la limite élastique du matériau utilisé (en général, l'aluminium 7075 avec une limite d'environ 250 MPa), et la masse du longeron doit rester raisonnable.

- **Le longeron à section rectangulaire** satisfait l'ensemble des contraintes :
 - Flèches maximales : 21,11 cm au sol et 17,1 cm en vol, toutes deux bien inférieures à la limite de 50 cm.
 - Contraintes maximales : 47,6 MPa au sol et 38,5 MPa en vol, bien en dessous de la limite admissible.
 - Masse : 123,64 kg, conforme aux attentes.
- **Le longeron à section circulaire** présente des résultats problématiques :
 - Flèches maximales : 14,8 cm au sol et 15,1 cm en vol, elles respectent la contrainte de flèche.
 - Contraintes maximales : 230 MPa au sol et 244 GPa en vol. Si la première valeur est acceptable, la seconde est irréaliste (probablement une erreur d'unité ou de simulation).
 - Masse : 94,68 kg, avantageuse mais insuffisante pour compenser la contrainte excessive en vol.

Finalement, Le modèle à **section rectangulaire est le seul qui respecte entièrement le cahier des charges**. Il combine rigidité, résistance mécanique et masse

maîtrisée. Le modèle circulaire, bien que plus léger et suffisamment rigide, présente une contrainte en vol manifestement excessive, ce qui le rend inadapté en l'état.

VII Conclusion

Ce rapport a détaillé les différentes étapes du dimensionnement mécanique d'une aile de planeur, en se concentrant sur le choix de l'épaisseur et du matériau de l'aile, ainsi que sur le dimensionnement du longeron. Les objectifs principaux étaient d'analyser les contraintes géométriques et mécaniques, de déterminer l'épaisseur optimale de l'aile, d'évaluer les performances mécaniques, de choisir le matériau approprié, et de dimensionner le longeron.

Les modèles théoriques et les simulations numériques doivent être utilisés de manière complémentaire pour garantir la sécurité, la performance et l'efficacité du planeur. Les résultats obtenus montrent que l'Aluminium 7075-T6 est un matériau approprié pour la conception de l'aile, et que les modèles simplifiés peuvent fournir des indications utiles, bien que des simulations plus avancées soient nécessaires pour une validation complète.