

Vers une Aviation Durable

Rapport Intermédiaire

Lefebvre Inès, Neveux Baptiste, Larroutuou David, Leene Jules

5 décembre 2025

Table des matières

1 Définition du projet	2
1.1 Objectif Principal	2
1.2 La Mission (Simulation)	2
1.3 Outils et méthodologie	2
1.4 Hypothèses simplificatrices	2
2 Organisation et Planification	2
2.1 Répartition du travail	2
2.2 Déroulé du projet	2
3 Programme Python : Analyse Structurelle	3
3.1 Calcul des propriétés d'une section	3
3.2 Calcul des moments de flexion et déplacements	3
3.3 Force critique de flambage	4
3.4 Calcul de la masse de l'aile	5
4 Analyse Aérodynamique (XFLR5)	5
4.1 Justification des choix	5
4.2 Résultats obtenus	5
4.3 Interprétation	6
4.4 Limites et Conclusion	6

1 Définition du projet

1.1 Objectif Principal

Le projet consiste à simuler une entreprise (équipe de 4) devant concevoir la géométrie d'un avion de ligne capable d'effectuer le trajet **Lille-Copenhague** avec 150 passagers (soit environ 13.5 tonnes de masse utile, sur la base de 90 kg par personne avec bagages). L'objectif central est de consommer le moins de carburant possible.

1.2 La Mission (Simulation)

La performance de l'avion est évaluée sur une simulation numérique de 200 secondes, extrapolée ensuite pour le trajet complet. Le profil de la mission est le suivant :

- **Départ** : Vol droit à altitude de croisière $h = 25\,000$ m et vitesse de croisière $V_c = 870$ km/h, avec un cap initial de -45°(Nord).
- **Manœuvre** : Virage pour atteindre le cap objectif de -60°(direction de Copenhague).
- **Conditions de réussite** :
 - Arriver au cap objectif avec une précision de $|\Psi + 60^\circ| < 1^\circ$.
 - Maintenir l'altitude et l'angle de dérapage stables (pente $< 0.5^\circ$).

1.3 Outils et méthodologie

Le challenge est purement numérique et multidisciplinaire :

- **Aérodynamique & Géométrie** : Utilisation de **XFLR5** pour concevoir l'avion et estimer les coefficients (le fuselage est ignoré pour l'aérodynamique mais compté dans la masse).
- **Structure** : Utilisation et amélioration d'un outil Python fourni pour vérifier la résistance du fuselage et des ailes (MM_RDM).
- **Dynamique du vol** : Utilisation d'un simulateur Python pour tester la trajectoire.
- **Gestion** : Estimation des coûts et de la chaîne de valeur.

1.4 Hypothèses simplificatrices

La masse est considérée constante (malgré la consommation), le centre de gravité est fixe, l'atmosphère standard est constante, et le TSFC (Thrust Specific Fuel Consumption) est fixé à 15 g/k.N.s. Pour que la portance compense le poids, la relation suivante doit être vérifiée :

$$L = \frac{1}{2} \rho S V^2 C_L = mg \quad (1)$$

2 Organisation et Planification

2.1 Répartition du travail

Les responsabilités ont été distribuées comme suit au sein de l'équipe :

2.2 Déroulé du projet

- **1ère phase (jusqu'au 28/11)** : Définition du projet, interprétation des consignes, découverte de XFLR5, définition des premiers jalons et mise en équation.
- **2ème phase (jusqu'au 5/12)** : Finalisation du rapport intermédiaire, développement du programme Python et calculs structurels.

Membre	Tâches principales
Baptiste	Interprétation Python de la partie structure (calculs de moments, déplacements, forces...)
Jules	Rédaction du rapport final
Inès	Interprétation des résultats aérodynamiques obtenus
David	Réalisation de la simulation sur XFLR5 et obtention des résultats

TABLE 1 – Matrice de répartition des tâches

3 Programme Python : Analyse Structurelle

Cette section détaille les scripts développés pour répondre aux exercices 1, 2, 3 et 4 du cours de structure.

3.1 Calcul des propriétés d'une section

Le script calcule l'aire (S), le moment quadratique (I_x, I_y) et le rayon de giration pour une section en I.

```

1 def calcul_section(h, b, e):
2     """
3         Calcule S (Aire), Ix (Inertie flexion), et Rho pour une section en I.
4     """
5     # Aire : 2*petit rectangle + grand rectangle
6     S = 2*b*e + h*e
7
8     # Inertie Ix (autour axe neutre horizontal)
9     I_x = e*( h**3/12 + b*e**2/6 + b*(h + e)**2/2)
10
11    I_y = e/6*( h*e**2/2 + b**3)
12
13    return S, I_x, I_y

```

Listing 1 – Fonction calcul_section

Exemple d'application numérique : Pour une section en I de 7×3 cm avec une épaisseur de 4 mm :

- Aire (S) : 5.200000e-04 m²
- Moment Quadratique (I_x) : 4.432133e-07 m⁴
- Moment Quadratique (I_y) : 1.837333e-08 m⁴

3.2 Calcul des moments de flexion et déplacements

Ce module détermine $M(x)$ et $y(x)$ par superposition des cas de charge répartie (q) et de force ponctuelle (F).

```

1 def calcul_poutre_console_superposition(x_array, L, a, F, q, E, I):
2     # Cas 1 : Charge répartie q (C01)
3     # M(x) = -q/2 * (L-x)^2
4     M_q = -(q / 2) * (L - x_array)**2
5     # Formule de la flèche y(x) pour charge répartie
6     y_q = (q / (24 * E * I * L)) * x_array**2 * (x_array**2 - 4*L*x_array + 6*L
7     **2)
8
9     # Cas 2 : Force ponctuelle F en a (C02)
10    # Initialisation des tableaux
11    M_F = np.zeros_like(x_array)
12    y_F = np.zeros_like(x_array)

```

```

12
13 # Formules par morceaux
14 for i, x in enumerate(x_array):
15     if x <= a:
16         M_F[i] = -F * (a - x)
17         y_F[i] = (F / (6 * E * I)) * x**2 * (3*a - x)
18     else:
19         M_F[i] = 0
20         y_F[i] = (F / (6 * E * I)) * a**2 * (3*x - a)
21
22 # Superposition
23 y_tot = y_q + y_F
24 M_tot = M_q + M_F
25
26 return M_tot, y_tot

```

Listing 2 – Calcul par superposition (C01 + C02)

Exemple d’application numérique : Avec $L = 20.0$, $a = 4.0$, $F = 200.0$, $q = -15.0$, $E = 70$ GPa, $I \approx 4.43e-07$ m⁴ :

- Moment Fléchissant Max : 2200.00 N.m (à $x = 0.00$ m)
- Flèche Maximale (bout) : 479.18 mm (à $x = 20.00$ m)

3.3 Force critique de flambage

L’évaluation de la force critique compare le flambage global (Euler) et le voilement local/-plastique (Johnson/Crippling).

```

1 def calcul_force_critique(L, E, I, S, nu, b, h, e):
2     # 1. Flambage Global (Euler) - Hypothèse Bi-encastré K=0.5
3     K = 0.5
4     L_eq = K * L
5     if L_eq == 0: L_eq = 0.001
6     F_euler = (np.pi**2 * E * I) / (L_eq**2)
7
8     # 2. Voilement Local (Crippling)
9     # Récupération de k via interpolation (Data_k_bh.py)
10    # Calcul de Sigma_cc (formule cours)
11
12    # 3. Flambage Réel (Interaction Johnson-Euler)
13    # Comparaison élancement vs élancement critique
14
15    if élancement >= élancement_critique:
16        # Domaine élastique -> Euler
17        F_critique_finale = F_euler
18        mode = "Euler (Global)"
19    else:
20        # Domaine plastique/local -> Johnson
21        # Sigma_johnson = Sigma_cc - ...
22        F_critique_finale = sigma_johnson * S
23        mode = "Johnson (Local/Plastique)"
24
25 return F_critique_finale, mode, élancement

```

Listing 3 – Calcul de la force critique

Résultats pour la section donnée ($E = 70$ GPa, $I \approx 4.43e-07$) :

- Pour $L = 0.5$ m :
 - Force Critique = **351 323.02 N**
 - Élancement = 8.56
 - Mode : Johnson (Local/Plastique)
- Pour $L = 2.0$ m :

- Force Critique = **253 301.46 N**
- Élancement = 34.25
- Mode : Johnson (Local/Plastique)

3.4 Calcul de la masse de l'aile

Conformément à la consigne d'implémenter une formule pour la masse, voici la fonction ajoutée au projet :

```

1 def calcul_masse_aile(envergure, corde, e_eq, rho):
2     """
3         Calcule la masse estimée d'une aile.
4         e_eq : Épaisseur équivalente (m)
5         rho : Densité du matériau (kg/m^3)
6         """
7     S_ref = envergure * corde
8     vol_matiere = 2 * S_ref * e_eq
9     masse = vol_matiere * rho
10    return masse

```

Listing 4 – Calcul de la masse de l'aile

4 Analyse Aérodynamique (XFLR5)

4.1 Justification des choix

Pour assurer une modélisation cohérente, les choix suivants ont été faits :

- **Profil NACA 4412** : Retenu car modérément cambré, adapté aux conditions ($V_\infty = 10$ m/s, $Re \approx 3 \times 10^6$).
- **Méthode** : Analyse 2D via Xfoil pour générer les polaires (Re 1M à 4M), puis méthode des panneaux 3D (VLM) pour l'aile complète.
- **Géométrie** : Envergure 36 m, corde constante 4 m, pas de twist ni de dièdre. Cette simplification est volontaire pour isoler les effets fondamentaux sans complexifier l'interprétation géométrique.
- **Conditions** : Analyse à $V_\infty = 10$ m/s pour des incidences de -2°, 0° et 4°.

4.2 Résultats obtenus

Le tableau ci-dessous résume les coefficients obtenus via l'analyse VLM :

Alpha	CL	CD	Cm	XCP (m)	ZCP (m)
-2°	0.18848	0.01147	-0.18882	5.165	1.288
0°	0.37910	0.01521	-0.27108	3.911	1.222
4°	0.75736	0.03012	-0.44468	3.294	1.190

TABLE 2 – Résultats de l'analyse VLM ($V_\infty = 10$ m/s)

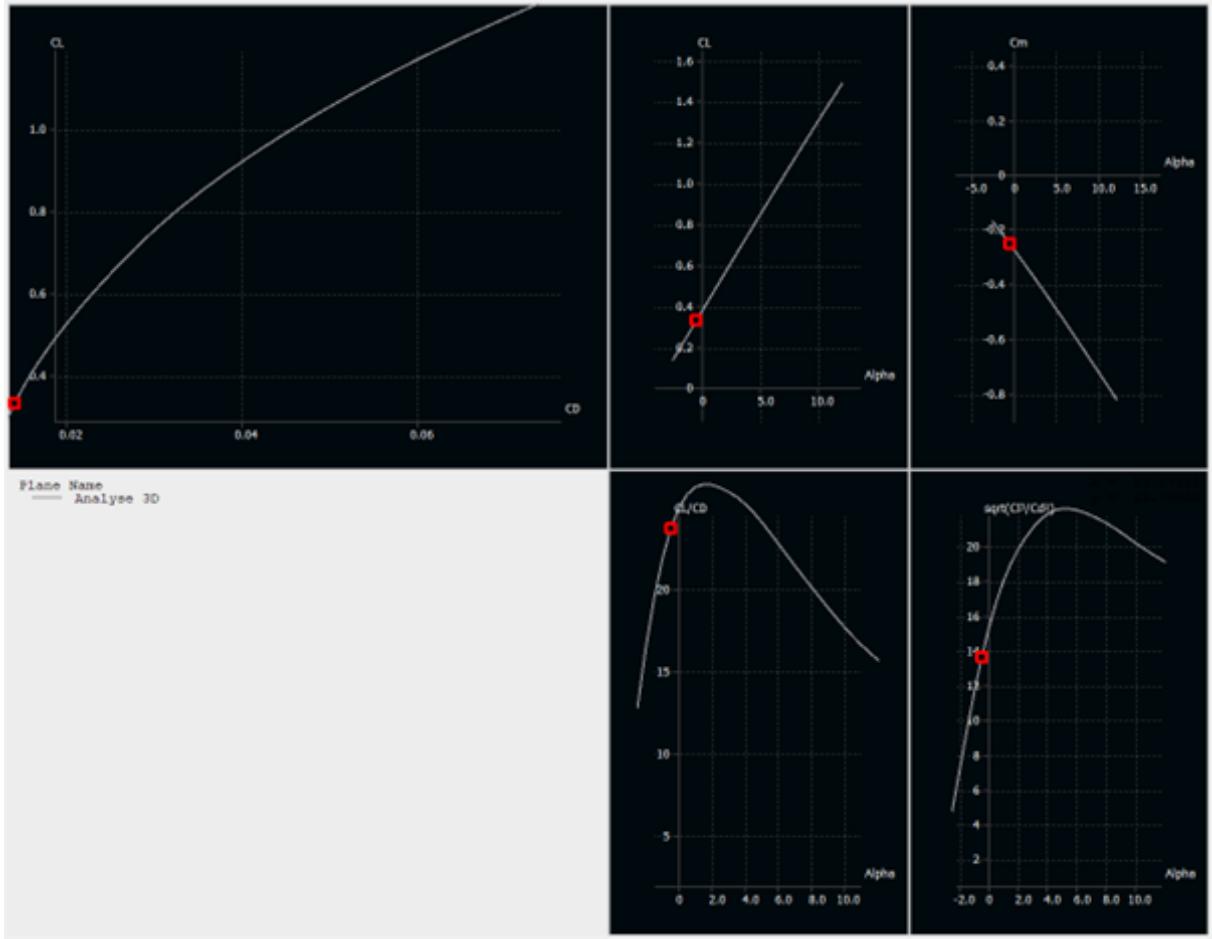


FIGURE 1 – Évolution des coefficients aérodynamiques (Polaire, C_L , C_m et Finesse)

4.3 Interprétation

- Portance (C_L) :**
 - À -2° , la portance est faible mais positive grâce à la cambrure.
 - À 0° , elle double presque.
 - À 4° , $C_L \approx 0.75$, indiquant un régime stable loin du décrochage.
- Traînée (C_D) :** Elle augmente logiquement avec l'incidence. Elle double à 4° (environ 0.03) par rapport au régime faible incidence, cohérent avec l'augmentation de la traînée induite ($\propto C_L^2$). L'optimum C_L/C_D se situe vers $2-3^\circ$.
- Moment de tangage (C_m) :** Toujours négatif (piqueur), comportement typique d'un profil cambré qui nécessitera un empennage horizontal compensateur.
- Centre de poussée (X_{CP}) :** Il avance vers le bord d'attaque lorsque l'incidence augmente (de 5.16 m à 3.29 m).

4.4 Limites et Conclusion

La modélisation présente des limites connues : absence de fuselage (sous-estimation de la traînée), absence de twist (optimisation de charge alaire non réalisée), et limites de la méthode VLM sur le décrochage.

Néanmoins, l'analyse montre que l'aile équipée du profil NACA 4412 possède un comportement sain, performant et une marge de sécurité confortable. Les outils structurels et aérodynamiques sont désormais opérationnels pour la suite du projet.