

Prédimensionnement Simplifié d'un Avion : Implémentation d'un Code Python

Dylan PERINETTI

28 novembre 2024

Table des matières

1	Introduction	2
2	Implémentation du Code Python	2
2.1	Structure du Code	2
2.2	Explication du Code	3
2.2.1	Chargement des données et extraction des variables	3
2.2.2	Calculs principaux	3
3	Résultats Obtenus	4
4	Conclusion	5

Résumé

Ce document présente l'implémentation d'un script Python pour automatiser les calculs de prédimensionnement simplifié d'un avion. Le code permet de charger des données spécifiques à un avion, d'effectuer les calculs nécessaires et de générer un rapport des résultats. Cette approche facilite l'adaptation à différentes configurations d'avions et améliore l'efficacité du processus de conception. Le code et la documentation sont disponibles sur le dépôt GitHub : https://github.com/dylanPerinetti/airplane_calculations

1 Introduction

Le prédimensionnement d'un avion est une étape cruciale dans le processus de conception aéronautique. Il implique de nombreux calculs qui peuvent être sujets à des erreurs lorsqu'ils sont effectués manuellement. L'automatisation de ces calculs à l'aide d'un script Python offre une solution efficace pour améliorer la précision et la rapidité du processus.

2 Implémentation du Code Python

Pour automatiser les calculs de prédimensionnement simplifié et permettre une adaptation aisée à d'autres configurations d'avions, un script Python a été développé. Ce script lit les données spécifiques à l'avion ainsi que les paramètres environnementaux à partir de fichiers JSON, effectue les calculs nécessaires et génère un rapport des résultats.

2.1 Structure du Code

Le code est structuré de manière modulaire, facilitant la lecture et la maintenance. Les principales étapes du script sont :

1. **Chargement des données et extraction des variables** : Les données spécifiques à l'avion et les paramètres environnementaux sont chargés depuis des fichiers JSON (`avion_data.json` et `parametres.json`). Les variables nécessaires aux calculs sont extraites des dictionnaires Python obtenus.
2. **Calculs principaux et affichage des résultats** : Le script réalise les calculs en suivant l'ordre des sections précédentes et affiche les résultats. Il calcule notamment :
 - La puissance mécanique nécessaire en croisière (P_m) et la puissance chimique nécessaire (P_c).
 - Le débit massique de carburant (\dot{m}_c) et la masse totale de carburant nécessaire, avec une marge.
 - La masse totale de l'avion.
 - Le nombre de Reynolds (Re).
 - Le coefficient de portance (C_z) calculé pour la croisière, permettant de vérifier la cohérence avec le C_z fourni par *PredimRC*.
 - La surface alaire requise (S), l'envergure (L) et la corde moyenne (l).
 - Les dimensions de l'empennage et des ailerons.

2.2 Explication du Code

Le code commence par importer les modules nécessaires :

```
import json
import math
```

2.2.1 Chargement des données et extraction des variables

Les données sont chargées à partir des fichiers JSON et les variables sont extraites :

```
with open('avion_data.json', 'r') as fichier_avion:
    avion_data = json.load(fichier_avion)

with open('parametres.json', 'r') as fichier_parametres:
    parametres = json.load(fichier_parametres)

# Extraction des variables
nom_avion = avion_data['nom_de_l_avion']
masse_a_vide = avion_data['masse_a_vide']
# ... autres variables ...
```

2.2.2 Calculs principaux

Les calculs suivent l'ordre des sections précédentes :

1. Puissance mécanique en croisière (P_m) :

$$P_m = \text{pourcentage}_{P_m} \times P_{m0} \quad (1)$$

2. Puissance chimique nécessaire (P_c) :

$$P_c = \frac{P_m}{\nu_m} \quad (2)$$

3. Débit massique de carburant (\dot{m}_c) :

$$\dot{m}_c = \frac{P_m}{\nu_h \times \nu_m \times \text{PCI}} \quad (3)$$

4. Masse totale de carburant (M_c) :

$$M_c = \dot{m}_c \times t_{\text{vol}} \times (1 + \text{marge}) \quad (4)$$

5. Masse totale de l'avion (m) :

$$m = m_{\text{vide}} + M_c + m_{\text{pilote}} + m_{\text{charge utile}} \quad (5)$$

6. Nombre de Reynolds (Re) :

$$Re = \frac{V \times l_{\text{corde}}}{\nu_{\text{air}}} \quad (6)$$

7. Coefficient de portance (C_z) :

$$C_z = \frac{2 \times m \times g}{\rho \times S \times V^2} \quad (7)$$

8. Surface alaire requise (S) :

$$S = \frac{2 \times m \times g}{\rho \times C_z \times V^2} \quad (8)$$

9. Envergure (L) et corde moyenne (l) :

$$L = \sqrt{\lambda \times S} \quad (9)$$

$$l = \frac{S}{L} \quad (10)$$

10. Géométrie de l'empennage et des ailerons :

$$S_{\text{stab}} = 0,15 \times S \quad (11)$$

$$S_{\text{dériv}} = 0,6 \times S_{\text{stab}} \quad (12)$$

$$S_{\text{ail}} = 0,1 \times S \quad (13)$$

$$l_{\text{bras}} = 2,5 \times l \quad (14)$$

3 Résultats Obtenus

En exécutant le script avec les données du Hughes H-1 Racer, les résultats suivants sont obtenus :

- **Puissance mécanique en croisière** : $P_m = 360 \text{ kW}$.
- **Masse totale de carburant avec marge** : $M_c = 553 \text{ kg}$.
- **Masse totale de l'avion** : $m = 2\,453 \text{ kg}$.
- **Nombre de Reynolds** : $Re = 12,8 \times 10^6$.
- **Coefficient de portance calculé** : $C_z = 0,20$.
- **Surface alaire requise** : $S = 17,00 \text{ m}^2$.
- **Envergure et corde moyenne** : $L = 10,91 \text{ m}$ et $l = 1,56 \text{ m}$.
- **Dimensions de l'empennage et des ailerons** : Calculées selon les proportions standard.

Ces résultats confirment la cohérence des hypothèses de conception et valident les choix effectués pour les dimensions et les performances de l'avion.

4 Conclusion

Le développement de ce script Python a permis de consolider les calculs effectués lors du prédimensionnement simplifié de l'avion, en garantissant une cohérence entre les différentes étapes. Les résultats obtenus sont en accord avec les performances historiques du Hughes H-1 Racer, confirmant ainsi la validité des méthodes employées.

Références

- *PredimRC* - Logiciel de prédimensionnement aéronautique d'aéromodélisme fourni par le CNAM.
- Documentation Python - <https://docs.python.org/fr/3/>
- Dépôt GitHub du projet - https://github.com/dylanPerinetti/airplane_calculations