

Modélisation et construction réduite du PC-7 MkII via optimisation empirique assistée par simulation numérique

Travail de Maturité
au Gymnase cantonal du Bugnon site de Sévelin



présenté par
Arthur Frésard
Lausanne 2025

“The desire to fly is an idea handed down to us by our ancestors who, in their grueling travels across trackless lands in prehistoric times, looked enviously on the birds soaring freely through space, at full speed, above all obstacles, on the infinite highway of the air.” *Wilbur Wright*

Superviseur: Philippe Kobel, PhD

Expert: Dr. Mohamed Farhat (Laboratoire des Machines Hydrauliques, EPFL)

Date de la défense orale: 2025

Table des matières

Résumé	5
Avant-propos	7
1 Introduction	9
1.1 Cadre et problématique	9
1.2 Objectifs de ce travail	9
2 Contextes et bases théoriques	11
2.1 Caractéristiques d'un fluide	11
2.2 Pression	11
2.3 Vitesse d'un fluide	12
2.4 Viscosité	12
2.5 Écoulements	13
2.5.1 Régime compressible	13
2.5.2 Régime incompressible	13
2.6 Couche limite	13
2.7 Bref historique de l'aérodynamique	14
2.8 Forces	15
2.8.1 Portance	16
2.8.2 Traînée	22
2.8.3 Moment de force	23
2.9 Nombre de Reynolds	25
2.10 Présentation du Pilatus PC-7	25
2.11 Profils NACA	25
3 Méthodologie générale	29
3.1 Présentation des logiciels et langage de programmation utilisés	29
3.1.1 OpenVSP	29
3.1.2 Fusion 360	30
3.1.3 Python	32
3.2 Méthode	32
3.2.1 Modélisation du fuselage et des ailes	32
3.2.2 Paramétrage des simulations	36
3.2.3 Calcul des paramètres obligatoires pour les simulations	37
3.2.4 Automatisation des simulations	39

TABLE DES MATIÈRES

4 Résultats et analyse des simulations	41
4.1 Convergence des calculs	41
4.2 Résultats comparatifs	44
4.3 Discussion	49
4.3.1 Comparaison des Reynolds avec le modèle original	49
4.3.2 Analyse des formes des courbes de tendances	50
4.3.3 Limites et fiabilité relative des résultats	52
4.4 Choix final et portée du classement	53
5 Construction et impression 3D du modèle réduit	55
5.1 Préparation du modèle pour l'impression	55
5.2 Découpage et organisation des pièces	56
5.3 Paramètres d'impression et choix du matériau	56
5.4 Assemblage et ajustements mécaniques	57
5.5 Système de guidage radiocommandé	57
5.6 Difficultés rencontrées et solutions	58
6 Conclusion	59
6.1 Bilan global du travail	59
6.2 Apports personnels et perspectives pour la suite	59
7 Annexe	61
7.1 Plans et fiche technique	61
7.2 Script formatage coordonnees profil	64
7.3 Script automatisation des simulations	64
7.4 Extraction des données	67
7.5 Notebook d'analyse des résultats	69
Remerciements	71

Résumé

Introduction - Dès mon enfance, j'ai été fasciné par le vol des avions. Ayant toujours cette passion, j'ai choisi de l'étudier dans ce travail de maturité. Néophyte en matière d'aéronautique, j'ai consulté internet pour connaître quelles sont les connaissances à étudier pour présenter mon étude. Ayant acquis ces dernières années les connaissances d'impression d'objet en 3D, j'ai décidé de modéliser un avion monomoteur, le PILATUS PC-7 MkII. Le choix de cet avion est culturelle et pratique, car il n'existe aucun modèle opensource permettant sa réalisation. La problématique centrale de ce travail consiste à déterminer quels phénomènes physiques sont à l'origine des forces aérodynamiques, et comment peut-on les observer, analyser et comparer à travers des simulations numériques sur un modèle réduit ? Plus précisément : l'optimisation des performances aérodynamiques du modèle réduit PC7 MkII (échelle 1 pour 10) par variation de l'angle des ailes externes (voir Fig. 3.7) et de l'épaisseur du profil des ailes principales.

Méthode - Après quelques tentatives infructueuses avec d'autres logiciels, c'est finalement par le logiciel OpenVSP que j'ai modélisé le PC-7 MkII. J'ai entrepris une modélisation complète à partir d'un plan de référence téléchargé d'un site de modélisme suisse. Les simulations ont testé six angles d'ailes externes et trois épaisseurs de profil, soit 18 modèles à analyser. Optimiser des simulations prend beaucoup de temps. Pour éviter de devoir rester devant l'ordinateur, j'ai écrit un script Python et ainsi l'ordinateur a pu calculer indépendamment de ma présence pendant plus de treize jours. J'ai développé une méthode de correction des valeurs aberrantes générées par VSPAERO afin d'obtenir des courbes graphiques viables permettant le choix du meilleur modèle physiquement cohérent.

Résultats et Discussion - Les simulations ont validé l'influence des variables choisies sur la portance et la traînée, avec des courbes de rapport portance-traînée (L_D) suivant les tendances classiques décrites dans la littérature : augmentation progressive avec l'angle d'incidence, puis chute correspondant au décrochage. Le classement final, basé sur l'intégration du rapport L/D et la dérivée du coefficient de moment de force C_M , a identifié le modèle optimal combinant angle d'ailes de 10° et épaisseur de 15%. Cependant, les résultats restent propres au modèle réduit et ne sont pas représentatifs du comportement de l'avion grandeur nature, car conserver le même nombre de Reynolds nécessiterait une envergure de 12,5 m ou une vitesse de vol quasi Mach 13. La construction du modèle réduit fonctionnel sera finalisé pour l'oral en LW-PLA-HT (matériau d'impression 3D résistant à des chaleurs de bons jours d'été) avec découpage optimisé pour l'impression 3D et intégration de l'électronique de radiocommande.

Avant-propos

Depuis tout petit, voler m'a toujours fasciné. Être assis dans un oiseau de métal tout figé et n'avoir rien de visible à l'œil nu en dessous de l'appareil a été une grande source d'inspiration et d'épicurisme envers la science. Je compris plus tard qu'il y avait en réalité une atmosphère tout autour de nous, composée de différents gaz. Ainsi, si l'on déplace à une vitesse appropriée un objet dans cette masse de molécules, des forces apparaissent. L'étude de ces forces et de leurs corrélations avec la forme des objets avec lesquels le flux d'air interagit, s'appelle l'aérodynamisme.

Ce rapport s'inscrit dans le cadre d'un travail de maturité en physique appliquée, avec des connaissances technologiques et scientifiques. Il portera sur la compréhension des phénomènes aérodynamiques, en particulier la portance, appliqués à l'aviation, puis sur l'optimisation d'un modèle d'avion selon différents paramètres aérodynamiques (C_L , C_D , C_M , L/D). Plus récemment, j'ai développé une passion pour l'aviation et le pilotage sur simulateur. Le travail d'un ingénieur concevant et construisant un avion, de l'empennage aux trains d'atterrissement, m'intéresse particulièrement.

Je n'ai cependant ni bagage universitaire, ni le temps nécessaire au design et à la création complète d'un avion en partant de zéro. C'est pourquoi j'ai choisi un modèle existant. Pour des raisons culturelles, j'ai choisi un Pilatus PC-7 Mk II, car Pilatus est un constructeur d'avions suisse. Le PC-7 fait partie de la Patrouille suisse et a longtemps été utilisé comme avion d'entraînement militaire national. Il est également monomoteur, ce qui est plus pratique pour la conception d'un modèle réduit télécommandé.

Il faut souligner que j'ai rencontré quelques difficultés. J'ai d'abord eu beaucoup de peine avec les simulations. J'ai commencé par prendre un modèle open source du PC-7 MkII sur internet pour ensuite le découper et ne garder que le fuselage. Mais cela présentait plusieurs problèmes. Premièrement, comme j'avais à ma disposition les plans, les modèles utilisés se sont avérés approximatifs après vérification. De plus, ma première idée de logiciels permettant les calculs des différentes valeurs aérodynamiques ne permettait pas l'importation d'un fichier 3D STL — *Standard Tessellation Language*.

J'ai donc directement modélisé mon avion sur le logiciel de modélisation (OpenVSP), pour ensuite exporter ce modèle vers le logiciel de calcul (Ansys Student). Mais cela ne fonctionnait toujours pas. Lors de la création du fichier — cette fois au format Trimmed STEP (*Standard for the Exchange of Product model data*) — OpenVSP créait des zones dont deux surfaces se

chevauchaient ou s'écartaient. J'ai donc finalement utilisé uniquement OpenVSP pour tout, car cela me permettait de modéliser l'avion selon les bons profils, les bonnes formes, et de pouvoir ensuite tout simuler directement dans la même application.

Une autre difficulté majeure a été que j'ai dû effectuer beaucoup de simulations afin d'optimiser le PC-7 sur les variables. Depuis le tout début de mes recherches et tests, je cherchais à pouvoir automatiser complètement le processus de simulation et de stockage des résultats, afin d'éviter de devoir rester tout le temps devant l'ordinateur et ainsi de faire durer inutilement le processus de calcul. Ce n'est que tardivement que j'ai pu écrire un script Python qui m'a permis de laisser tourner les calculs pendant un peu plus de treize jours.

Concernant la modélisation du modèle réduit fonctionnel, son impression 3D et sa mise en marche réelle, de par mon expérience j'ai pu éviter beaucoup de problèmes potentiels et obtenir ce que je voulais.

Je précise que l'utilisation d'outils d'intelligence artificielle dans ce travail s'est limitée à la recherche de littérature pertinente via Google Scholar, ainsi qu'à la reformulation ponctuelle de certaines phrases dont les mots n'étaient pas employés de manière appropriée. Pour être précis, j'ai d'abord effectué mes recherches moi-même, puis j'ai utilisé Perplexity [2] afin de comparer ses résultats aux miens et m'assurer de n'avoir omis aucun document pertinent. Parmi les résultats fournis par l'IA, je n'ai conservé qu'un seul document traitant de la convergence des courbes de tendances, les autres étant déjà mieux traités par les sources trouvées précédemment.

Sinon, Perplexity a été utilisé uniquement comme un dictionnaire intelligent, permettant de trouver des termes mieux adaptés à mes explications afin de maintenir une compréhension claire. L'objectif n'était pas de faire travailler une intelligence artificielle sur un sujet qui me passionne, mais uniquement de l'utiliser comme outil d'aide à la rédaction et à la recherche.

1 Introduction

1.1 Cadre et problématique

Comme dit plus haut, je n'ai ni bagage universitaire ni le temps nécessaires à une création complète d'un avion à partir de zéro. C'est pourquoi j'ai choisi un avion existant comme base, sur lequel je conduirai des tests sur la portance et l'aérodynamisme par mise en simulation sur ordinateur, avant de construire un modèle réduit fonctionnel. J'ai choisi un avion suisse : le Pilatus PC-7 Mk II. Cet avion est le modèle utilisé par l'armée pour l'entraînement des futurs pilotes de F/A-18 depuis 35 ans et il est aussi le symbole iconique de la PC-7 Team de Suisse. L'autre facteur définissant le choix de cet avion a été qu'il n'existe aucun modèle libre de droit permettant la construction de son modèle réduit — je parle ici des fichiers 3D permettant soit son impression, soit son usinage en vue d'une création fonctionnelle.

Les problématiques de mon travail sont les suivantes : quels phénomènes physiques sont à l'origine des forces aérodynamiques, et comment peut-on les observer, analyser et comparer à travers des simulations numériques sur un modèle réduit ? Je me concentrerai uniquement sur l'analyse aérodynamique externe, c'est-à-dire les formes, les surfaces et les profils d'aile, sans entrer dans les détails liés à la propulsion, aux commandes de vol ni aux systèmes embarqués.

1.2 Objectifs de ce travail

Voici les différents objectifs de mon travail :

1. Comprendre et expliquer les principes aérodynamiques fondamentaux, notamment la portance et la traînée, à l'aide du livre d'Anderson et de Giovannini.
2. Comment les variations de l'angle des ailes et de l'épaisseur du profil de l'aile influencent-elles le rapport portance/traînée et le moment de force sur le modèle du PC-7 Mk II ?
3. Déterminer expérimentalement, par simulation CFD - *Computational Fluid Dynamics*, la forme d'aile la plus efficiente (en termes de rapport portance/traînée et de moment de force) pour le PC-7 Mk II à vitesse de croisière.
4. Peut-on reproduire fidèlement, à l'échelle réduite, les effets aérodynamiques observés en

grandeur réelle, en tenant compte du nombre de Reynolds ?

5. Créer un modèle réduit fonctionnel du PC-7 Mk II, intégrant la meilleure configuration d'aile issue des simulations, en utilisant OpenVSP et Fusion 360.
6. Faire voler ce modèle réduit en évitant de le crasher ou d'autres désagréments de ce genre et noter mes impressions pendant le vol.
7. Mettre à disposition en open source l'ensemble des données, fichiers 3D, scripts, résultats de simulation et plans de construction, pour contribuer à la communauté.

2 Contextes et bases théoriques

Avant d'avancer plus loin dans mon travail et afin d'une meilleure compréhension, voici les différents concepts de base permettant de suivre mes explications ultérieures sur les forces aérodynamiques et l'analyse de mes résultats. Pour écrire cette section, je n'ai utilisé que le livre d'Anderson [17], le livre de Giovannini [14] et quelques pages Wikipédia.

2.1 Caractéristiques d'un fluide

Le mot "fluide" peut être utilisé pour désigner à la fois un liquide et un gaz. Dans le cadre de ce travail, le mot fluide désignera un gaz, car l'étude porte sur l'aérodynamisme, c'est-à-dire l'étude des mouvements de l'air autour des objets. Pour distinguer un fluide d'un solide, il faut observer sa déformation sous l'effet d'une force appliquée tangentiellement. Pour un solide, il en résulte une déformation finie et cette force appliquée est généralement proportionnelle à la quantité de déformation. Pour un fluide, il en résulte une continue augmentation de la déformation et la force est généralement proportionnelle au taux de changement de cette déformation.

2.2 Pression

La pression est la force normale exercée par unité de surface, résultant du taux de variation temporelle de la quantité de mouvement des molécules de gaz qui heurtent ou traversent une paroi. Une chose importante à retenir est qu'il n'est pas nécessaire d'avoir une surface de 1 m^2 pour parler de pression. On définit généralement la pression en un point du fluide ou en un point d'une surface solide, et la pression peut varier d'un point à un autre. La quantité de mouvement est un principe physique qui indique que pour un objet avec une certaine vitesse, une quantité de mouvement est associée à cet objet et tend à être conservée tant qu'aucune interaction extérieure n'intervient. Dans le cas des molécules de gaz, chaque collision avec une surface transfère une petite quantité de mouvement à cette surface, et la force exercée dépend de la vitesse des molécules et de leur fréquence de collision. La pression est donc le résultat de ce transfert continu de quantité de mouvement vers la surface. Ce sont donc bien des N/m^2 .

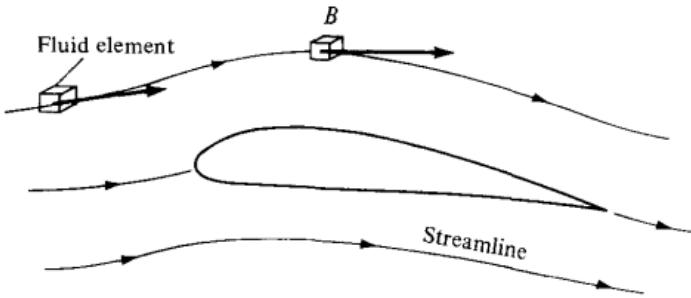


FIGURE 2.1 – Illustration des lignes de flux d'un fluide autour d'un profile. Tiré de : [17], p.15

2.3 Vitesse d'un fluide

La vitesse d'un fluide est définie comme la vitesse d'un élément infinitésimal du fluide traversant un point fixe dans l'espace (voir fig 2.1). À noter que c'est la seule quantité vectorielle dans le cas de l'aérodynamique, car la vitesse se décrit par la direction et la valeur elle-même ; les autres grandeurs sont toutes scalaires. Comme cette vitesse est définie en chaque point et qu'elle varie en chaque point de par sa direction et son intensité, on utilise toujours, lors de calcul, la vitesse *au loin* du fluide. On peut l'utiliser dans ces deux formes : vectoriel ou scalaire.

2.4 Viscosité

La viscosité se conçoit comme une sorte de "colle" plus ou moins forte présente dans chaque fluide et qui est responsable de son écoulement. Plus un fluide a une viscosité élevée, plus il s'écoule lentement. L'exemple du verre d'eau que l'on verse comparé à la pâte à tresse l'illustre bien : l'un s'écoule beaucoup plus facilement que l'autre. De manière rigoureuse, on définit la viscosité d'un fluide comme l'ensemble des phénomènes de résistance au mouvement d'un fluide lors de son écoulement. C'est-à-dire que la viscosité diminue la liberté d'écoulement du fluide et dissipe son énergie. À noter que la température joue un rôle important ici. La température correspond à l'agitation moyenne des particules qui constituent le fluide. Pour un gaz, cela correspond à l'énergie cinétique moyenne des molécules. Si la température d'un liquide d'une certaine viscosité est augmentée, sa viscosité diminue. Une plaque de chocolat chauffée montre exactement le même effet : d'abord elle fond, puis, plus on chauffe, plus elle devient liquide et facile à manipuler.

2.5 Écoulements

2.5.1 Régime compressible

La compressibilité est la propriété d'un fluide à changer de volume lorsqu'il subit une variation de pression. Cette compression peut s'accompagner d'une élévation de température suivant les substances.

Un écoulement compressible est un écoulement dans lequel la densité du fluide varie au cours de l'écoulement par des changements de pressions. La densité d'un fluide est la masse par unité de volume du fluide. C'est le type d'écoulement que l'on rencontre lors de vol subsoniques ou supersoniques.

Selon Anderson, un écoulement devient compressible de manière significative lorsque le nombre de Mach (exprime le rapport de la vitesse d'un objet dans un fluide à la vitesse du son) dépasse environ 0.3, car les variations de densité deviennent alors non négligeables. Cela signifie que dans ces cas, la masse volumique du fluide ne peut plus être considérée comme constante, et il faut prendre en compte des équations plus complexes pour modéliser son comportement.

2.5.2 Régime incompressible

Un écoulement incompressible est un écoulement utopique ou imaginaire, car il n'en existe pas dans la réalité ; il aide en revanche à formuler des principes mathématiques dans lesquels la densité du fluide est supposée constante, quel que soit l'endroit de l'écoulement ou la pression exercée. Autrement dit, le volume du fluide ne change pas, peu importe les forces qu'il subit.

Dans la réalité, tous les fluides sont compressibles à un certain degré. Toutefois, pour des vitesses faibles (d'après Anderson pour des nombres de MACH inférieures à 0.3), les variations de densité sont tellement petites qu'on peut raisonnablement les négliger. Dans ce cas, on parle d'écoulement incompressible par approximation, ce qui permet de simplifier fortement les équations gouvernant le mouvement du fluide.

2.6 Couche limite

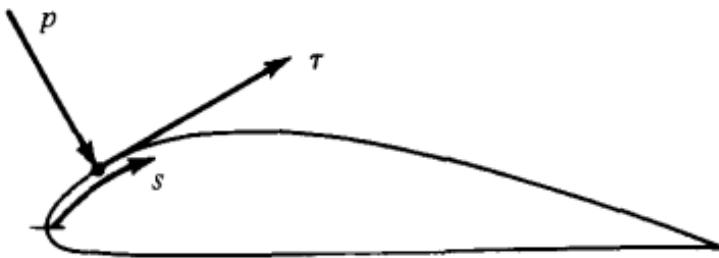
Le principe de la couche limite décrit l'interaction entre l'air et une paroi. Les molécules d'air situées directement au contact du profil sont immobiles, tandis que leur vitesse augmente progressivement avec la distance au profil, jusqu'à atteindre celle de l'écoulement libre (vitesse *au loin*). Ces différentes vitesses de molécules à certaines distances du profil constituent des couches d'air. On peut ainsi expliquer la force de frottement, non pas par le frottement présumé d'une seule couche de flux d'air avec le profil, mais comme le frottement entre les différentes couches d'air. Cette interaction entre couches de flux d'air induit aussi les différentes zones de

pression.

2.7 Bref historique de l'aérodynamique

Comprendre comment un oiseau vole dans le ciel ou comment exactement un dauphin se déplace dans l'eau a été une source de fascination depuis l'Antiquité. Aristote s'est interrogé sur la nature de l'air et de son mouvement. Mais ce ne sera qu'au XVIIe siècle que les premières tentatives rigoureuses de compréhension des forces agissant sur des corps en mouvement dans l'air apparaissent. Isaac Newton dédiera son deuxième livre entièrement à la dynamique des fluides par une approche mécanique du mouvement des corps dans un fluide, ce qui donnera une bonne base, même si ces modèles restent très simplifiés et limités [11]. Ce n'est qu'au XVIIIe siècle qu'un Suisse appelé Daniel Bernoulli établit le principe fondamental, qui portera son nom : plus la vitesse d'un fluide augmente, plus sa pression diminue [17]. Un peu plus tard, Leonard Euler, un Bâlois, pose ses équations (qui portent son nom) et qui décrivent le mouvement des fluides incompressibles et sans viscosité.

Au XIXe siècle, les équations d'Euler sont enrichies et généralisées pour les fluides visqueux par Navier et Stokes. C'est à cette période que le physicien George Cayley identifie les quatre forces impliquées dans le vol aérien : la poussée, la traînée, la portance et la force de gravité. Elles seront expliquées dans les prochaines sections (Sect. 2.8)[10]. Au XXe siècle, tout s'accélère considérablement : ce siècle marque l'ère des grandes avancées théoriques et expérimentales. En premier lieu, les frères Wright, après plusieurs échecs, remettent toutes leurs connaissances de l'aérodynamique en cause et vont passer deux ans dans un hangar dans l'Ohio, avec un tunnel, deux moteurs et plus de 200 profils d'ailes à mesurer, pour finalement réaliser le premier vol motorisé du monde en 1903[17]. C'est au début de la Première Guerre mondiale que le physicien Ludwig Prandtl introduit le concept de couche limite (Sect. 2.6) ; il élabore également la première modélisation opérationnelle de l'écoulement autour des profils alaires. La couche limite permet de poser les bases de la plupart des calculs modernes de frottements, transferts de chaleur et séparations de flux. À partir de là, les souffleries deviennent de plus en plus efficaces, puis les ordinateurs prennent le relais, car, comme expliqué plus tard, toutes ces forces sont issues de la pression, de la viscosité, de la vitesse et de la densité à chaque point de l'aile, et donc le calcul répété en chaque point — soit l'intégrale — permet le calcul de ces forces. À partir des années 1950, l'apparition des vols supersoniques impose une révision des bases de l'aérodynamique, car les effets compressibles deviennent dominants [16]. Ces calculs par ordinateur sont appelés CFD - *Computational Fluid Dynamics* et permettent de modéliser des écoulements avec une grande précision, offrant des analyses détaillées de profils d'ailes ou de, à quelques expressions près, n'importe quel écoulement.



$p = p(s)$ = surface pressure distribution
 $\tau = \tau(s)$ = surface shear stress distribution

FIGURE 2.2 – Illustration de la contrainte de cisaillement et de la pression autour d'un profil.
Tiré de : [17], p.18

2.8 Forces

La première chose à bien comprendre est que toutes les forces aérodynamiques sont issues de ces deux caractéristiques : la distribution de la pression (Sect. 2.2) et la distribution de la contrainte de cisaillement sur la surface de l'objet. La contrainte de cisaillement est, contrairement à la pression, une force tangentielle agissant parallèlement à la surface de l'objet (Voir fig. 2.2). Elle résulte du frottement visqueux entre l'air et la surface dans la couche limite. Il faut préciser que l'air en contact avec la surface du profil a une vitesse forcée de zéro à cause de l'adhérence moléculaire, ce qui crée un gradient de vitesse dans la couche limite et génère ainsi des efforts de cisaillement responsables de la traînée de frottement.

Ces forces aérodynamiques peuvent se comparer à une expérience familière : sous la douche, lorsqu'on interrompt l'eau pour se savonner (afin d'économiser notre ressource la plus précieuse), il arrive qu'un pain de savon, arrondi par l'usage, échappe de la main mouillée si l'on serre trop fort. Cette pression exercée autour du savon, combinée à la souplesse de la peau, engendre alors un frottement cinétique, autrement dit une contrainte de cisaillement.

La limite de cette comparaison, c'est qu'ici il n'existe aucune couche limite. En réalité, l'air en contact direct avec la surface de l'aile est totalement immobile ; seules les couches supérieures de cet air se déplacent, créant un frottement cinétique avec les couches inférieures. L'autre différence, c'est qu'ici le pain de savon obtient une accélération et saute donc de notre main. Mais ce n'est pas vraiment un flux d'air qui arrive sur lui : c'est nous qui appuyons de manière plus ou moins statique, ce qui crée un mouvement vers l'avant. Pour un avion, en revanche, c'est le flux d'air qui vient sur lui et génère une résistance en raison de sa surface frontale perpendiculaire au flux mais cela ne crée pas un mouvement vers l'avant.

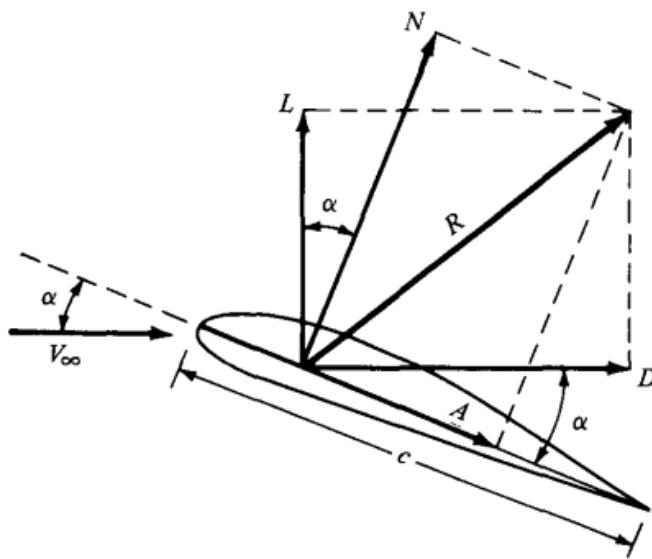


FIGURE 2.3 – Illustration des résultantes aérodynamiques et de leurs composantes. Tiré de : [17], p.16

2.8.1 Portance

La portance est la composante de la force résultante sur l'avion perpendiculaire au vent relatif, c'est-à-dire le flux d'air rencontré par l'aile en mouvement (V_∞ sur la Fig. 2.3). Elle permet à l'avion de s'opposer à la force de gravité, de décoller et de se maintenir en vol. Une incompréhension courante de ce phénomène provient d'une application trop simplifiée du principe de Bernoulli : ce n'est pas seulement la différence de pression entre l'intrados et l'extrados qui génère la portance, mais bien l'intégration complète des pressions sur l'ensemble de la surface du profil. La portance résulte en grande partie de cette différence de pression, créée par la forme du profil (souvent asymétrique) et par l'incidence, c'est-à-dire l'angle entre le profil et le vent relatif (α sur la Fig. 2.3). Cette différence de pression est elle-même liée à l'écoulement de l'air autour du profil : l'air circule plus vite sur l'extrados et plus lentement sur l'intrados, générant une zone de basse pression au-dessus et une zone de haute pression en dessous de l'aile. Cette description repose sur l'hypothèse d'un écoulement incompressible (Sect. 2.5.2), c'est-à-dire sans variation de densité dans le fluide, ce qui permet d'appliquer le principe de Bernoulli. Celui-ci relie pression et vitesse d'un fluide de densité constante : une augmentation de vitesse entraîne une diminution de pression et vice versa. Toutefois, cette explication idéale reste simplifiée, car la portance réelle découle de l'intégration complète des distributions de pression et des contraintes de cisaillement sur toute la surface de l'aile.

Une fois tout ceci intégré, ce n'est pas la portance que calcule l'intégrale mais la résultante de la portance et de la traînée. Selon la figure 2.3, on peut en réalité voir deux paires de forces, selon le repère choisi. La première paire se constitue à partir du profil même de l'aile et elle est composée de la force axiale et de la force normale par rapport à la corde du profil – *ligne*

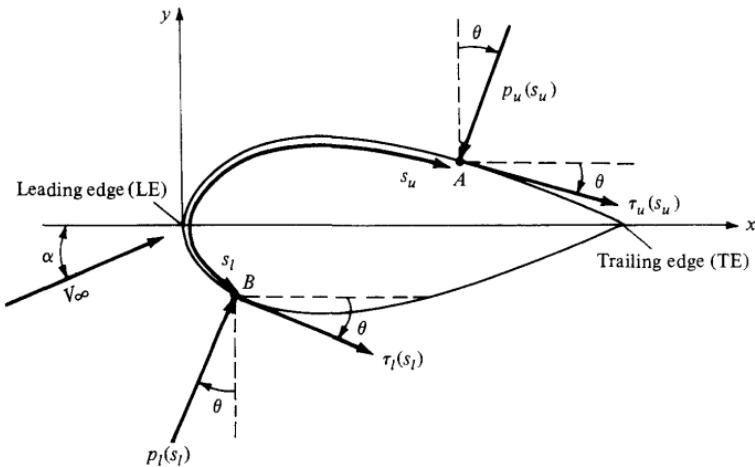


FIGURE 2.4 – Illustration de l'intégration de la pression et de la contrainte de cisaillement avec la nomenclature des équations, tiré de : [17], p.17

reliant le *bord d'attaque* et le *bord de fuite*, LE-TE. Ces forces seront dénotées N et A . La deuxième paire de forces contient la portance et la traînée. Ces forces sont simplement une autre décomposition de la résultante R selon un autre repère : le vent. La portance est la composante normale au vent et la traînée est la composante parallèle. D'où la dénotation des deux paires de forces.

La portance L se calcule selon la formule :

$$L = \cos(\alpha)N - \sin(\alpha)A \quad (2.1)$$

Où N et A désignent respectivement les composantes normale et axiale de la force résultante par rapport à la direction du vent. α est l'angle entre le vent au loin et la corde du profil. N et A sont les normes des vecteurs associés.

La question suivante est de calculer N et A . Le but de ce calcul est d'avoir la force totale opérant parallèlement à la corde et la force totale opérant perpendiculairement à la corde. Naïvement, on pourrait penser qu'il faut simplement intégrer uniquement la pression pour N et la contrainte de cisaillement pour A , mais comme le montrent la figure 2.2 et la figure 2.4, la pression agira toujours perpendiculairement à la surface du profil et la contrainte de cisaillement sera tangentielle à la surface du profil. Il faut donc, pour chaque points infinitésimal du profil, décomposer ces deux forces dans notre repère issu de la corde du profil et tout additionner pour trouver la force normale et la force axiale. Nous connaissons théoriquement la distribution de pression et de contrainte de cisaillement autour de notre profil (voir Figure 2.2) notées $\rho(s)$ et $\tau(s)$, que nous abrégerons ρ et τ .

Il faut savoir que l'on va toujours mesurer l'angle positivement dans le sens inverse des aiguilles d'une montre et négativement dans l'autre sens, comme pour le moment cinétique (voir Figure 2.5). De cette convention, il faut écrire une équation pour l'extrados (indice u pour



FIGURE 2.5 – Illustration du sens et du signe conventionnel pour un moment cinétique, tiré de : [17], p.19

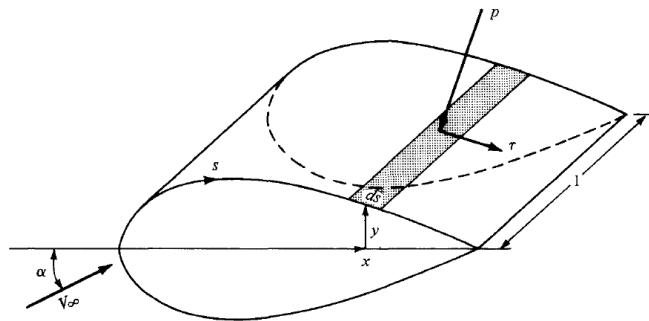


FIGURE 2.6 – Illustration d'écart infinitésimal autour d'un profil en 3 dimensions.

"upper part") et une autre pour l'intrados du profil et rajouter des signes négatifs pour rester cohérent avec la figure 2.4.

$$dN'_u = -\cos(\theta)\rho_udS_u - \sin(\theta)\tau_udS_u \quad (2.2)$$

$$dA'_u = -\sin(\theta)\rho_udS_u + \cos(\theta)\tau_udS_u \quad (2.3)$$

d est un écart infinitésimal. $[']$ exprime une partie de N . Reprenons la figure 2.4. Ces calculs se basent sur des quantités vectorielles, ainsi il faut faire attention à regarder l'orientation de la force et la décomposer dans le bon sens sur les axes x , y pour comprendre l'origine des signes négatifs dans la force décomposée. Pour N' et la distribution de pression, si l'on prend une force perpendiculaire à la surface du profil et orientée vers celui-ci, une fois décomposée sur l'axe y , cette composante est négative. Par conséquent, pour n'importe quel θ , puisque sur un intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$ cosinus est toujours positif, on doit introduire un signe négatif pour que cette force soit dirigé vers le haut. Pour N' et la distribution de contrainte de cisaillement, lorsque la force est au dessus de l'horizontal, sinus est négatif et la composante y est positive, il faut ainsi multiplié par un signe négatif, lorsque la force est au dessous de l'horizontal, sinus est positif mais la composante y est négative, donc avec le signe négatif on obtiens bien du positif. Pour A' et la distribution de pression, c'est la même explication que pour N' et la distribution de contrainte de cisaillement, simplement que ce n'est pas sur l'axe y mais l'axe x . Et finalement pour A' et la distribution de contrainte de cisaillement, la composante x est toujours positive. Nous vivons dans un espace en 3 dimensions, ainsi il suffit de multiplié par un écart infinitésimal de la surface de l'aile S pour obtenir une force par unité d'envergure (voir figure 2.6).

Voici les expressions mathématiques de l'intrados (indice l pour "lower part") pour des

points infinitésimaux pour N' et A' :

$$dN'_l = \cos(\theta)\rho_l dS_l - \sin(\theta)\tau_l dS_l \quad (2.4)$$

$$dA'_l = \sin(\theta)\rho_l dS_l + \cos(\theta)\tau_l dS_l \quad (2.5)$$

Pour N' et la distribution de contrainte de cisaillement, l'explication est la même que pour sa comparse de l'extrados. Sinon il faut toujours décomposer les forces en regardant bien le sens par rapport à l'axe sur lequel elle est décomposée et savoir que cosinus est toujours positif et que sinus est négatif pour des angles négatifs. En ayant bien ça en tête, le reste est positif de lui-même.

Il s'agit maintenant d'additionner l'extrados et l'intrados et de l'écrire sous forme d'intégrale ; du bord d'attaque ("leading edge" LE) au bord de fuite ("trailing edge" TE).

$$N' = \int_{\text{LE}}^{\text{TE}} -\cos(\theta)\rho_u dS_u - \sin(\theta)\tau_u dS_u + \int_{\text{LE}}^{\text{TE}} \cos(\theta)\rho_l dS_l - \sin(\theta)\tau_l dS_l \quad (2.6)$$

Nous pouvons ainsi regrouper les dS et mettre les signes négatifs en évidence :

$$N' = - \int_{\text{LE}}^{\text{TE}} (\cos(\theta)\rho_u + \sin(\theta)\tau_u) dS_u + \int_{\text{LE}}^{\text{TE}} (\cos(\theta)\rho_l - \sin(\theta)\tau_l) dS_l \quad (2.7)$$

$$A' = \int_{\text{LE}}^{\text{TE}} (-\sin(\theta)\rho_u + \cos(\theta)\tau_u) dS_u + \int_{\text{LE}}^{\text{TE}} (\sin(\theta)\rho_l + \cos(\theta)\tau_l) dS_l \quad (2.8)$$

Nous avons tout ce qu'il nous faut pour écrire une description complète de la portance.

On multiplie d'abord N' et A' par l'envergure l de par la figure 2.6 pour obtenir les forces totales N et A :

$$N = lN', \quad A = lA' \quad (2.9)$$

Puis on les substitue dans l'équation 2.1 :

$$L = \cos(\alpha)lN' - \sin(\alpha)lA' \quad (2.10)$$

En remplaçant N' par A' par leurs expressions intégrales des équations 2.7 et 2.8 en mettant en évidence l , on obtient :

$$L = l \left\{ \cos(\alpha) \left[- \int_{\text{LE}}^{\text{TE}} (\cos(\theta)\rho_u + \sin(\theta)\tau_u) dS_u + \int_{\text{LE}}^{\text{TE}} (\cos(\theta)\rho_l - \sin(\theta)\tau_l) dS_l \right] - \sin(\alpha) \left[\int_{\text{LE}}^{\text{TE}} (-\sin(\theta)\rho_u + \cos(\theta)\tau_u) dS_u + \int_{\text{LE}}^{\text{TE}} (\sin(\theta)\rho_l + \cos(\theta)\tau_l) dS_l \right] \right\} \quad (2.11)$$

2 Contextes et bases théoriques

On regroupe les contributions correspondant à l'extrados - u et à l'intrados - l séparément :

$$L = l \left\{ \int_{LE}^{TE} [-\cos(\alpha)(\cos(\theta)\rho_u + \sin(\theta)\tau_u) - \sin(\alpha)(-\sin(\theta)\rho_u + \cos(\theta)\tau_u)] dS_u + \int_{LE}^{TE} [\cos(\alpha)(\cos(\theta)\rho_l - \sin(\theta)\tau_l) - \sin(\alpha)(\sin(\theta)\rho_l + \cos(\theta)\tau_l)] dS_l \right\} \quad (2.12)$$

On développe les termes entre crochets et on réduit en regroupant avec les termes ρ et τ :

$$L = l \left\{ \int_{LE}^{TE} [-\rho_u(\cos(\alpha)\cos(\theta) - \sin(\alpha)\sin(\theta)) - \tau_u(\sin(\alpha)\cos(\theta) + \cos(\alpha)\sin(\theta))] dS_u + \int_{LE}^{TE} [\rho_l(\cos(\alpha)\cos(\theta) - \sin(\alpha)\sin(\theta)) - \tau_l(\sin(\alpha)\cos(\theta) + \cos(\alpha)\sin(\theta))] dS_l \right\} \quad (2.13)$$

De par les identités trigonométrique suivantes, nous pouvons simplifier l'expression 2.13, tirées de : [12] :

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta), \quad \sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \quad (2.14)$$

Et ainsi :

$$L = l \left\{ - \int_{LE}^{TE} (\rho_u \cos(\alpha + \theta) + \tau_u \sin(\alpha + \theta)) dS_u + \int_{LE}^{TE} (\rho_l \cos(\alpha + \theta) - \tau_l \sin(\alpha + \theta)) dS_l \right\} \quad (2.15)$$

Finalement, on peut sommer les contributions extrados/intrados et noter une variable combinée dS qui est la somme formelle de dS_l et dS_u le long de la corde.

$$L = l \int_{LE}^{TE} [(\rho_l - \rho_u) \cos(\alpha + \theta) - (\tau_l + \tau_u) \sin(\alpha + \theta)] dS \quad (2.16)$$

C'est une manière de définir la portance mais l'on peut aussi la définir à l'aide de coefficient sans dimension. On va déjà définir la pression dynamique - q_∞ , dont ρ_∞ et V_∞ sont la densité et la vitesse dans le courant libre *au loin* de l'objet.

$$q_\infty = \frac{1}{2} \rho_\infty V_\infty^2 \quad (2.17)$$

La première chose qui m'a frappé est la ressemblance de cette équation avec l'énergie cinétique.

$$E_{cin} = \frac{1}{2} m V^2 \quad (2.18)$$

J'ai donc fait une analyse de dimension entre ces deux équations :

$$\left[E_{cin} = \frac{Kg * m^2}{s^2} \right] \quad (2.19)$$

$$\left[q_\infty = \frac{Kg * m^2}{m^3 * s^2} = \frac{Kg}{m * s^2} \right] \quad (2.20)$$

Comme vu plus haut dans la section 2.2, nous savons qu'une pression c'est une force par unité de surface - $\frac{N}{m^2}$. Nous savons par la deuxième loi de Newton, qu'une force est une masse multiplié par une accélération.

$$\vec{a} = \frac{\sum \vec{F}}{m}, \quad \vec{F} = m\vec{a} \quad (2.21)$$

Nous avons donc en dimension :

$$\left[N = \frac{Kg * m}{s^2} \right] \quad (2.22)$$

Ce qui nous donne bien notre pression dynamique :

$$\left[\frac{N}{m^2} = \frac{Kg}{m * s^2} \right] \quad (2.23)$$

Et pour revenir à notre énergie cinétique nous obtenons :

$$\left[\frac{E_{cin}}{m^3} = \frac{N}{m^2} \right] \quad (2.24)$$

La pression est donc aussi mathématiquement de l'énergie cinétique par unité de volume. Ce qui n'est pas sans logique avec l'explication faite dans la section 2.2. Maintenant, on introduit le coefficient de portance. L'idée, c'est d'éviter de refaire à chaque fois l'intégration complète : autrement dit, si l'on conserve exactement le même profil, le même angle d'incidence (α) et le même nombre de Reynolds (voir Sect. 2.9) – ce qui implique, dans notre contexte, la même vitesse et les mêmes dimensions – alors la distribution de pression et de contrainte de cisaillement reste identique. On peut donc réutiliser directement le coefficient sans recalculer toute l'intégrale.

Comme mentionné au tout début de cette section, la portance – tout comme les autres forces aérodynamiques – provient uniquement de la distribution de pression et de la distribution de contrainte de cisaillement sur la surface. En observant l'équation 2.17 de la pression dynamique, on remarque qu'un facteur vitesse intervient. Les distributions évoquées plus haut dépendent donc elles aussi de la vitesse.

Si cette vitesse n'apparaît pas explicitement dans l'équation 2.13, c'est parce qu'elle est "cachée" à l'intérieur des distributions de pression et de contrainte de cisaillement, qui ne sont pas des constantes mais bien des fonctions dans notre cas. C'est d'ailleurs parfaitement intuitif : un avion doit atteindre une certaine vitesse pour décoller, faute de quoi la portance générée est insuffisante.

On définit donc le coefficient de portance comme :

$$C_L = \frac{L}{q_\infty S} \quad (2.25)$$

Où S désigne la surface alaire de l'aile.

Nous avons donc cette autre manière de définir la portance :

$$L = \frac{1}{2} \rho_{\infty} S C_L V_{\infty}^2 \quad (2.26)$$

2.8.2 Traînée

La traînée est la force qui s'oppose au mouvement de l'avion et qui est parallèle au vent relatif. Elle comprend deux composantes : la traînée de forme et la traînée de frottement. La première résulte de la différence de pression entre l'avant et l'arrière de l'aile, c'est-à-dire des forces de pression intégrées. La seconde est causée par les contraintes de cisaillement intégrées (voir section 2.8). L'enjeu majeur est de limiter au maximum cette traînée, ce qui permet de réduire la consommation de carburant et d'améliorer l'accélération. Comme dit plus haut, le calcul de l'intégration des distributions de pressions et de contraintes de cisaillements donne la résultante R comme l'illustre la figure 2.3. On peut donc, à partir de ces composantes N et A dans le référentiel de l'aile, définir la traînée :

$$D = \sin(\alpha)N + \cos(\alpha)A \quad (2.27)$$

Il faut toujours bien garder en tête que ce sont des quantités vectorielles et que c'est la principale origine de signes négatifs ou de non signes négatifs. On peut maintenant reprendre les équations d'intégrations 2.7 et 2.8 pour les substituer dans l'équation suivante :

$$D = \sin(\alpha)lN' + \cos(\alpha)lA' \quad (2.28)$$

Ce qui nous donne :

$$D = l \left\{ \sin(\alpha) \left[- \int_{LE}^{TE} (\cos(\theta) \rho_u + \sin(\theta) \tau_u) dS_u + \int_{LE}^{TE} (\cos(\theta) \rho_l - \sin(\theta) \tau_l) dS_l \right] + \cos(\alpha) \left[\int_{LE}^{TE} (-\sin(\theta) \rho_u + \cos(\theta) \tau_u) dS_u + \int_{LE}^{TE} (\sin(\theta) \rho_l + \cos(\theta) \tau_l) dS_l \right] \right\} \quad (2.29)$$

On regroupe les contributions de l'extrados et de l'intrados :

$$D = l \left\{ \int_{LE}^{TE} [-\rho_u (\sin(\alpha) \cos(\theta) + \cos(\alpha) \sin(\theta)) + \tau_u (\cos(\alpha) \cos(\theta) - \sin(\alpha) \sin(\theta))] dS_u + \int_{LE}^{TE} [\rho_l (\sin(\alpha) \cos(\theta) + \cos(\alpha) \sin(\theta)) + \tau_l (\cos(\alpha) \cos(\theta) - \sin(\alpha) \sin(\theta))] dS_l \right\} \quad (2.30)$$

En utilisant les identités, tirées de [12] :

$$\sin(\alpha + \beta) = \sin(\alpha) \cos(\beta) + \cos(\alpha) \sin(\beta), \quad \cos(\alpha + \beta) = \cos(\alpha) \cos(\beta) - \sin(\alpha) \sin(\beta)$$

On obtient :

$$D = l \left\{ \int_{LE}^{TE} [-\rho_u \sin(\alpha + \theta) + \tau_u \cos(\alpha + \theta)] dS_u + \int_{LE}^{TE} [\rho_l \sin(\alpha + \theta) + \tau_l \cos(\alpha + \theta)] dS_l \right\} \quad (2.31)$$

Finalement, on peut sommer les contributions extrados/intrados et noter une variable combinée dS qui est la somme formelle de dS_l et dS_u le long de la corde.

$$D = l \left\{ \int_{LE}^{TE} dS (\rho_l - \rho_u) \sin(\alpha + \theta) + (\tau_u + \tau_l) \cos(\alpha + \theta) \right\} \quad (2.32)$$

La traînée peut aussi se définir avec un coefficient de traînée sans dimension. De plus, tout comme pour le coefficient de portance (voir Sect. 2.8.1), cela facilite beaucoup les calculs notamment lorsque la vitesse est constante. Cela nous sera utile pour la suite. Donc on définit ce coefficient comme :

$$C_D = \frac{D}{q_\infty S} \quad (2.33)$$

On obtient donc

$$D = \frac{1}{2} \rho_\infty S C_D V_\infty^2 \quad (2.34)$$

2.8.3 Moment de force

Lorsque l'on vole, un autre facteur très important est la stabilité de l'avion, c'est-à-dire l'habileté de l'avion à ne pas s'incliner involontairement sur les 3 axes de l'espace. Or, toute rotation ou changement de rotation doit être causé par un moment de force. Comme nous pouvons déterminer les forces sur notre profil, nous pouvons donc calculer le moment de force.

Pour définir un moment de force, on prend la force \vec{F} et on fait un produit vectoriel avec le vecteur \overrightarrow{OF} reliant le point auquel on veut calculer le moment de force et l'endroit où la force est appliquée.

$$\vec{M} = \overrightarrow{OF} \times \vec{F} \quad (2.35)$$

Par construction du produit vectoriel ; le vecteur \vec{M} est perpendiculaire au plan que forme les vecteurs \overrightarrow{OF} et \vec{F} . Ainsi, si l'on n'a que des vecteurs contenus dans le plan xy , \vec{M} n'aura qu'une composante z . Ce vecteur pointe donc dans l'axe autour duquel tournera l'objet. Dans notre cas, il suffira de sommer tous ces vecteurs, même s'ils pointent dans des directions différentes, pour obtenir le moment de force résultant sur l'aile.

Maintenant revenons à nos distributions, ce qu'on va simplement trouver lors de chaque calcul infinitésimal c'est la force uniquement en y et uniquement en x . Ainsi il suffit de multiplier chaque force infinitésimale par son vecteur le reliant au bord d'attaque de l'aile pour trouver le

2 Contextes et bases théoriques

moment de force au bord d'attaque. Mais regardons vectoriellement ce que ça donne :

$$d\vec{M} = d\overrightarrow{OM} \times d\vec{F} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \times \begin{pmatrix} F_x \\ F_y \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ xF_y - yF_x \end{pmatrix} \quad (2.36)$$

Avant d'aller plus loin, il faut préciser dans quel sens on définit positivement un moment de force ; on le définit positivement dans le sens des aiguilles d'une montre et négativement inversement (même convention que pour le moment cinétique voir Fig. 2.5).

Nous pouvons donc reprendre nos équations 2.2 à 2.5 pour les appliquer dans 2.36. Comme pour N et A , il faut distinguer l'intrados de l'extrados et M' signifie que c'est un moment par unité d'envergure.

$$dM'_u = dS_u (-x(\cos(\theta)\rho_u + \sin(\theta)\tau_u) + y(-\sin(\theta)\rho_u + \cos(\theta)\tau_u)) \quad (2.37)$$

$$dM'_l = dS_l (x(\cos(\theta)\rho_l - \sin(\theta)\tau_l) + y(\sin(\theta)\rho_l + \cos(\theta)\tau_l)) \quad (2.38)$$

x et y sont des fonctions qui sont tout à fait connues qui définissent le profil (voir section 2.11). Il convient également de noter que cette expression n'est valable que pour le calcul du moment au bord d'attaque. En effet, dans les fonctions NACA (voir Sect. 2.11), le bord d'attaque correspond au point $(0, 0)$ du graphe, de sorte que les coordonnées des points du profil constituent directement les vecteurs issus de l'origine. Il suffit alors de sommer ces contributions et de les exprimer sous forme intégrale.

$$\begin{aligned} M'_{LE} &= \int_{LE}^{TE} dS_u (-x(\cos(\theta)\rho_u + \sin(\theta)\tau_u) + y(-\sin(\theta)\rho_u + \cos(\theta)\tau_u)) \\ &\quad + \int_{LE}^{TE} dS_l (x(\cos(\theta)\rho_l - \sin(\theta)\tau_l) + y(\sin(\theta)\rho_l + \cos(\theta)\tau_l)) \end{aligned} \quad (2.39)$$

Si l'on additionne formellement dS_u et dS_l et nommant ce résultat dS , on obtiens :

$$M_{LE} = l \int_{LE}^{TE} dS [x(\cos(\theta)(\rho_l - \rho_u) - \sin(\theta)(\tau_u + \tau_l)) + y(\sin(\theta)(\rho_l - \rho_u) + \cos(\theta)(\tau_u + \tau_l))] \quad (2.40)$$

l est l'envergure de l'aile.

On définit aussi un coefficient sans dimension pour le moment.

$$C_M = \frac{M}{q_\infty S l} \quad (2.41)$$

Nous avons donc :

$$M = \frac{1}{2} \rho_\infty C_M S l V_\infty^2 \quad (2.42)$$

2.9 Nombre de Reynolds

Le nombre de Reynolds est un paramètre important. Tout d’abord, il est sans dimension, c’est-à-dire qu’il n’a pas d’unité, ce qui permet de comparer différents systèmes, par exemple entre un modèle réduit et un avion réel. Concrètement, il compare deux types de forces présentes dans un écoulement : les forces d’inertie, qui traduisent la tendance du fluide à conserver sa vitesse et sa direction lorsqu’il est en mouvement, et les forces visqueuses, qui correspondent aux frottements internes du fluide et tendent à dissiper l’énergie cinétique et à homogénéiser les vitesses entre couches de flux voisines. Le nombre de Reynolds exprime ainsi l’importance relative de ces deux forces, comme le montre sa formule :

$$R_e = \frac{\rho V L}{\mu} \quad (2.43)$$

Où ρ est la densité, V la vitesse, L la longueur (la corde de l’aile) et μ la viscosité du fluide (l’air). À faible Reynolds, l’écoulement est *laminaire*, c’est-à-dire que les particules fluides suivent des trajectoires lisses en couches, chaque couche se déplaçant de manière régulière par rapport aux couches adjacentes avec peu ou pas de mélange. À haut Reynolds, l’écoulement devient *turbulent*. Ce type d’écoulement se caractérise par des variations chaotiques de pression et de vitesse. La turbulence, c’est-à-dire les effets chaotiques dans la trajectoire des particules d’air, résulte de l’excès d’énergie cinétique dans certaines parties de l’écoulement du fluide. Entre ces deux types d’écoulements, il existe un régime transitoire, où de petites perturbations apparaissent. Dans notre cas, le calcul du nombre de Reynolds permettra de montrer qu’il n’était pas dans nos moyens de construire un avion modèle réduit du PC7 MkII, qui possède des performances aérodynamiques comparable au modèle taille réel.

2.10 Présentation du Pilatus PC-7

Le Pilatus PC-7 Turbo Trainer est un avion d’entraînement à turbopropulseur biplace, développé par la société suisse Pilatus Aircraft. Il a été conçu comme successeur du Pilatus P-3 et a volé pour la première fois en 1978, entrant en service en 1982. Le PC-7 est devenu un succès commercial mondial, utilisé par plus de 20 forces aériennes comme principal appareil d’entraînement de base. Le PC7-MKII est simplement la nouvelle version de cet avion [13]. Il possède d’excellentes performances, une avionique moderne, et est polyvalent, ce qui est parfait pour la formation de pilote militaire.

2.11 Profils NACA

Les profils *National Advisory Committee for Aeronautics* se déclinent en plusieurs types que l’on appelle séries. Il existe des séries à 4, 5, 6, voire plus de chiffres. Les séries 4, 5 et 4/5

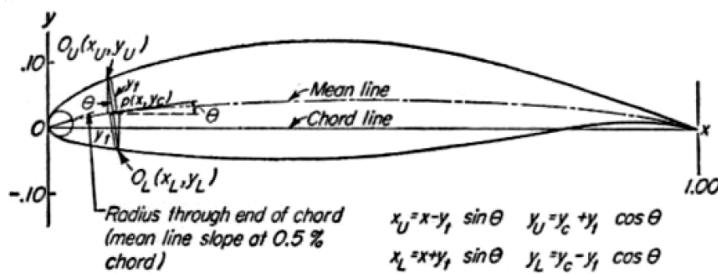


FIGURE 2.7 – Illustration d'un profil NACA avec la ligne centrale géométrique et la corde, tiré de [22]

modifiées ont été créées à partir d'équations analytiques décrivant la cambrure par rapport à la ligne géométrique centrale de l'aile, ainsi que la distribution d'épaisseur le long de la corde [22] [19]. Les séries supérieures sont issues de méthodes plus théoriques, où l'on choisit directement la répartition de pression autour de l'aile pour ensuite en déduire la forme.

Je ne détaillerai pas ici les fonctions permettant de dessiner un profil, car cela n'apporte pas réellement quelque chose à ce travail, si ce n'est prouver que l'on peut exprimer un profil par une fonction mathématique. Je me contenterai donc de décrire les deux séries utilisées pour la modélisation de mon avion dans OpenVSP.

Série 4

La série 4 s'écrit comme suit :

NACA XXXX

Où les X représentent des chiffres. Le premier chiffre est la cambrure maximale (en % de la corde), le deuxième indique la position de cette cambrure maximale (en dixièmes de corde), et les deux derniers donnent l'épaisseur maximale (également en % de la corde).

Série 6

NACA XX(X)-XXX

La série 6 a été développée afin de maximiser la zone sur laquelle l'écoulement reste laminaire autour du profil. Contrairement aux séries 4 et 5, obtenues à partir d'équations analytiques, la série 6 utilise une méthode théorique améliorée, similaire à celle de la série 1, qui consiste à spécifier la distribution de pression souhaitée puis à calculer la géométrie correspondante. Cette approche permet de réduire considérablement la traînée sur une plage restreinte de coefficients de portance.

La nomenclature de la série 6 est plus complexe que celle des séries précédentes, notamment en raison des nombreuses variantes possibles. Par exemple, considérons le profil NACA 641-212 $a = 0,6$:

- **6** : indique la série 6, optimisée pour un écoulement laminaire plus important que celui des séries à quatre ou cinq chiffres.
- **4** : position de la pression minimale en dixièmes de corde (ici 0,4).
- **1** (en indice) : plage de faible traînée maintenue pour des coefficients de portance $\pm 0,1$ autour du coefficient de portance de conception.
- **2** : coefficient de portance de conception (0,2).
- **12** : épaisseur maximale en pourcentage de la corde (ici 12 %).
- $a = 0.6$: fraction de la corde sur laquelle la distribution de pression est uniforme (ici 60 %). Si non précisé, $a = 1$ par défaut (uniforme sur toute la corde).

Ainsi, la série 6 permet une optimisation fine du profil pour des régimes spécifiques, mais sa lecture et son interprétation demandent plus de soin que les séries précédentes.

3 Méthodologie générale

3.1 Présentation des logiciels et langage de programmation utilisés

Avant de détailler la méthodologie employée dans ce travail, il est essentiel de présenter les principaux outils logiciels et langage de programmation utilisés. Ceux-ci jouent un rôle central à la fois dans la modélisation, l'analyse et le traitement des résultats. Chaque logiciel apporte des fonctionnalités spécifiques qui, combinées, permettent de mettre en place une chaîne de calcul cohérente et reproductible.

3.1.1 OpenVSP

Open Vehicle Sketch Pad est un logiciel de modélisation d'avions basé sur la géométrie analytique : les objets y sont décrits par des fonctions mathématiques. Il permet à l'utilisateur de créer des modèles 3D d'avion définis par des paramètres. Il permet ensuite de convertir ce modèle dans des formats compatibles avec d'autres outils d'analyse. Il contient plusieurs autres modules comme VSPAERO qui permet d'effectuer des analyses aérodynamiques directement depuis OpenVSP. [7]

Le prédecesseur de OpenVSP a été développé par J.R Gloudemans et d'autres pour la NASA dans les années 90. Le 10 janvier 2012, OpenVSP a été publié dans un projet open source sous la *NASA Open Source Agreement* (NOSA) version 1.3. Le développement a été poursuivi et dirigé par Rob McDonald depuis 2012 et a été supporté par la NASA et l'AFRL entre autres par la suite.

OpenVSP a une interface graphique au démarrage basée sur FLTK (voir Fig. 3.1). L'interface se compose d'une fenêtre d'espace de travail avec la visualisation (à gauche) et d'un navigateur de géométrie (à droite). L'espace de travail est l'endroit où le modèle est affiché et le navigateur permet de sélectionner les composants du modèle de l'espace de travail (ex. ailes, fuselage...). Ces composants peuvent être sélectionnés et modifiés (par ex. changer le profil de l'aile, la taille de la base et du bout de l'aile, la position, tout), supprimés ou ajoutés. Lorsque l'on sélectionne un composant, une fenêtre de navigation géométrique du composant s'ouvre et permet de le modifier. Il offre une multitude de géométries de base, courantes dans la modéli-

3 Méthodologie générale

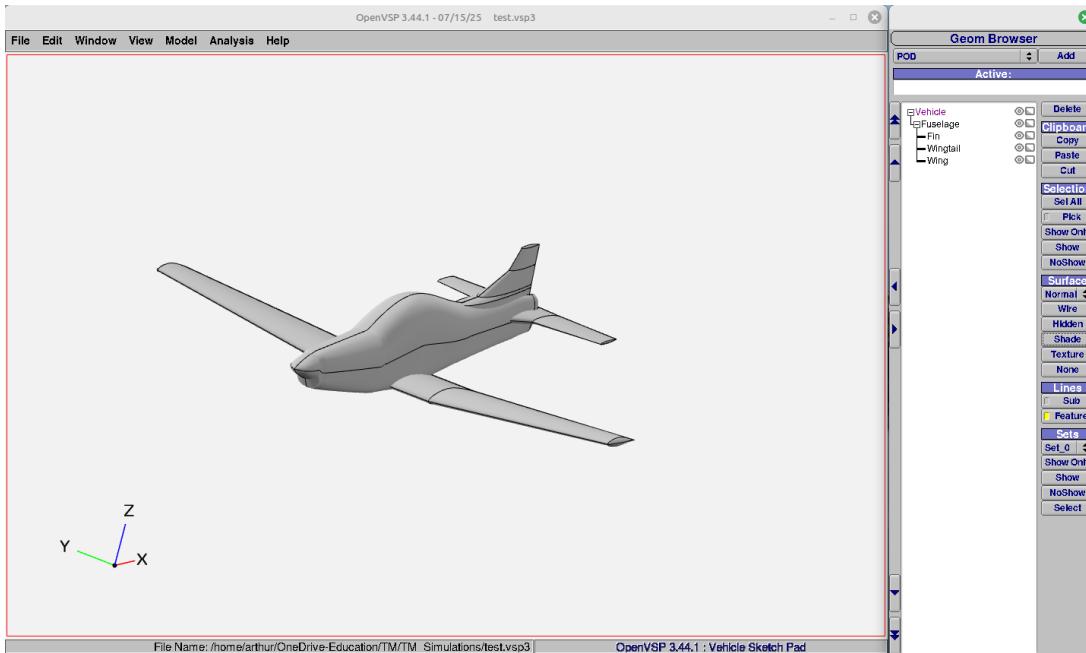


FIGURE 3.1 – Illustration de l’interface graphique de OpenVSP

sation aéronautique, que l’on peut modifier et assembler pour créer des modèles. Les ailes, les nacelles, le fuselage et les hélices sont quelques-unes des géométries disponibles. Des composants avancés tels que les corps de révolution, les conduits, la géométrie conforme et autres sont également disponibles.

OpenVSP peut également être utilisé au moyen d’APIs - *Application Programming Interface* avec Python par exemple. Une API permet d’interagir avec le logiciel via des lignes de code, plutôt qu’en utilisant l’interface graphique. Il comporte plusieurs outils d’analyses comme VSPAERO, CompGeom, TrimmedSurface et d’autres. J’ai cité ces trois outils, car je les ai utilisés lors de ce TM. CompGeom permet de générer un maillage à partir d’un modèle. Pour représenter un objet en trois dimensions, plusieurs méthodes existent. Le maillage en est une : il consiste à décrire l’objet en le découpant en un grand nombre de facettes triangulaires. Cette étape est indispensable pour les simulations numériques, car le solveur utilise ce maillage pour effectuer ses calculs. Le recours à des surfaces planes simplifie considérablement le traitement et réduit la complexité des calculs. Il est toutefois nécessaire d’employer un maillage suffisamment fin pour garantir des résultats précis et cohérents. TrimmedSurface permet de générer des fichiers 3D du modèle ajusté (trimé), qui peuvent être exportés vers d’autres logiciels tout en conservant les dimensions correctes. J’y reviendrai plus clairement lors de leurs utilisations dans la méthode (Sect. 3.2).

3.1.2 Fusion 360

Fusion 360 est un logiciel de modélisation 3D développé par Autodesk. Il est basé sur la modélisation paramétrique et surfacique, ce qui signifie que les objets sont définis à l’aide de

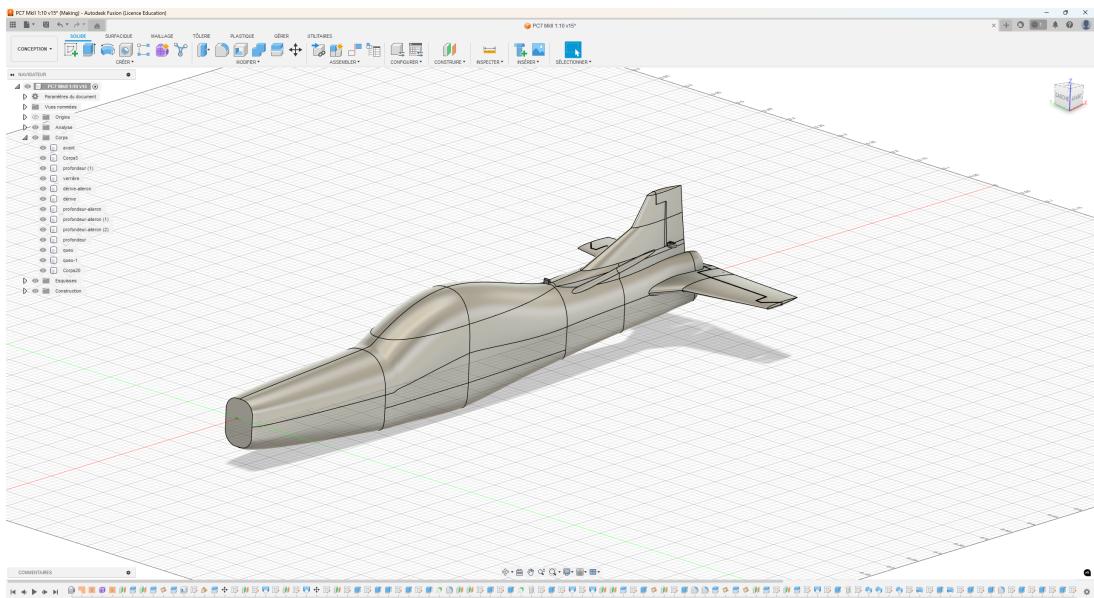


FIGURE 3.2 – Illustration de l’interface graphique de Fusion 360

paramètres géométriques et de contraintes. Contrairement à OpenVSP qui est spécialisé dans l'aéronautique, Fusion 360 est un outil de conception généraliste qui permet de créer des modèles 3D et de planifier tout ce qui concerne sa réalisation, cela va de l'export de fichiers pour de l'impression 3D à des CNC à commande numérique en passant par des circuits électriques [4].

Fusion 360 a été publié pour la première fois en 2013 par Autodesk comme une solution intégrée dans le cloud, combinant modélisation, rendu, simulation et fabrication dans une seule plateforme. Il s'est rapidement forgé une bonne réputation grâce à sa compatibilité avec différents formats industriels (STEP, IGES, STL, etc.) et sa facilité d'utilisation, notamment pour l'impression 3D et la fabrication assistée par ordinateur (FAO).

L'interface de Fusion 360 se compose d'un espace de travail central où s'affiche le modèle et d'une barre d'outils en haut permettant d'accéder aux fonctions principales (esquisses, extrusion, assemblages, etc.). La conception commence en général par une esquisse 2D sur un plan, qui peut ensuite être transformée en volume par extrusion, révolution ou autres opérations. Les composants créés peuvent être assemblés pour former des systèmes plus complexes avec des contraintes cinématiques et mécaniques. L'approche paramétrique permet de modifier facilement les dimensions ou les relations géométriques sans devoir redessiner tout le modèle.

Fusion 360 propose également des modules intégrés, comme la simulation mécanique, la génération de trajectoires pour l'usinage CNC, ou encore des outils de rendu réaliste pour la visualisation. Il est possible d'exporter les modèles vers de nombreux formats compatibles avec des logiciels tiers ou directement vers des machines de fabrication. Comme pour OpenVSP, Fusion 360 offre aussi des possibilités d'automatisation via des scripts en Python ou JavaScript, ce qui permet de créer des fonctions personnalisées ou d'automatiser certaines étapes répétitives de conception.

Dans le cadre de ce TM, Fusion 360 a été utilisé exclusivement pour transformer un modèle aérodynamique en un avion de modélisme imprimable en 3D et réellement fonctionnel, mais également pour effectuer des rendus numériques de mise en scène de l'avion (Cf. page de titre, Fig. 3.8). Le modèle intègre non seulement l'espace nécessaire pour les composants électriques, mais aussi différentes adaptations mécaniques destinées au contrôle des parties mobiles de l'appareil.

3.1.3 Python

Python est un langage de programmation polyvalent et de haut niveau d'abstraction, il est utilisé partout dans le monde que ce soit pour de l'automatisation, de la création d'outils personnalisés, de l'analyse de données, de la gestion de réseau de neurones, etc. Dans le cadre de ce TM, Python a été employé pour automatiser certains calculs et faciliter l'analyse des résultats obtenus avec les différents logiciels. Grâce à ses bibliothèques scientifiques telles que *NumPy* et *Matplotlib*, il a été possible de manipuler les données, de réaliser les calculs et de produire les différents graphes nécessaires.

Python a également servi d'interface avec certains outils tiers, notamment via des scripts permettant de traiter les fichiers exportés depuis OpenVSP ou Fusion 360. Ces scripts ont permis d'automatiser des tâches répétitives comme le nettoyage de données ou la préparation de fichiers pour les simulations ou encore la récupération des données.

Dans le cadre de ce TM, Python a donc constitué un outil de support essentiel pour l'automatisation des simulations aérodynamiques, le post-traitement des résultats et la validation des modèles, en complément direct d'OpenVSP et de Fusion 360.

3.2 Méthode

Cette section décrit la méthodologie réellement employée pour obtenir les résultats, depuis la modélisation du PC-7 Mk II jusqu'à la réalisation des simulations aérodynamiques. La démarche s'est structurée en plusieurs étapes, décrites ci-dessous.

3.2.1 Modélisation du fuselage et des ailes

L'idée de base était d'utiliser un modèle pré-modélisé du Pilatus téléchargé sur internet. Mais le modèle présentait plusieurs limites : il ne respectait ni les proportions exactes ni les profils d'ailes utilisés par le véritable avion. De plus, il était fourni au format STL – un type de fichier qui représente un objet en 3D uniquement à l'aide de triangles – incompatible avec OpenVSP pour la création de géométries paramétriques. Pour surmonter ces contraintes, j'ai entrepris une remodélisation complète du PC-7 à partir d'un plan de référence imprimé (voir Fig. 3.4). Les dimensions manquantes ont été obtenues par proportionnalité, tandis que le fuse-

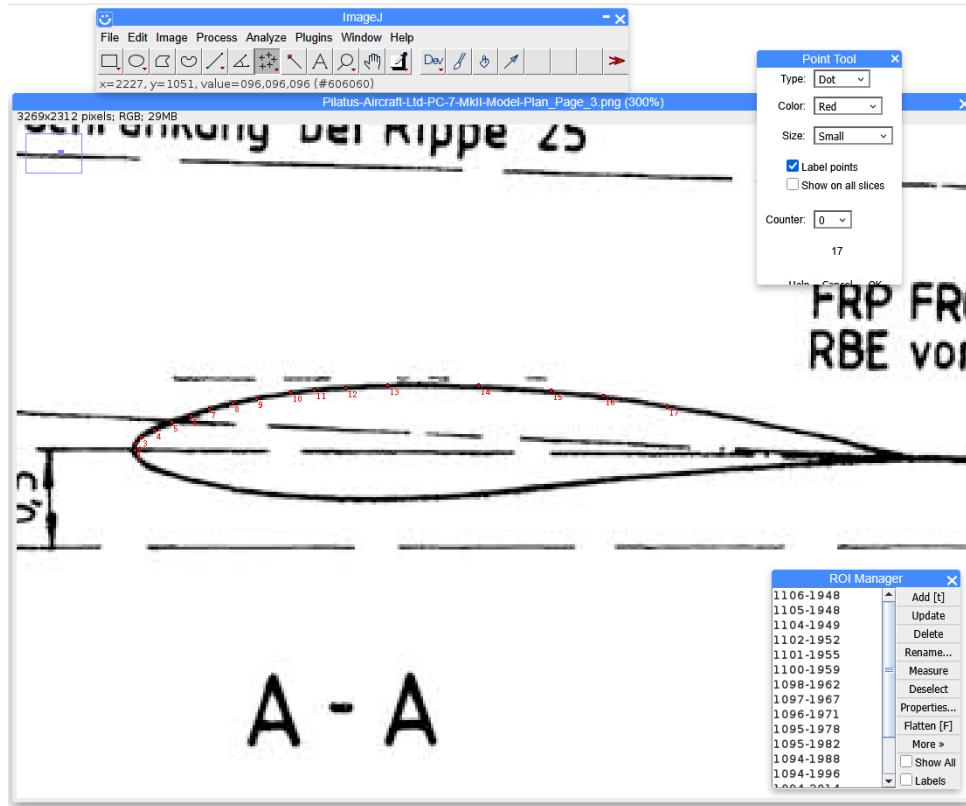


FIGURE 3.3 – Photo d’écran d’extraction des points avec ImageJ [6] à partir du plan d’ensemble (image de fond, voir Fig. 3.4)

lage, la dérive et l’empennage horizontal ont été reconstruits en respectant le plan et en utilisant directement les profils indiqués.

L’aile principale a nécessité un traitement particulier, car son profil devait être identifié avec précision. Pour cela, les coordonnées des points du profil ont été extraits à partir du plan au moyen du logiciel ImageJ [6], qui permet de placer des points sur une image et d’en récupérer les coordonnées (voir Fig. 3.3). Ces données ont ensuite été reformatées à l’aide d’un script Python afin de les rendre compatibles avec Airfoiltools, un site web mettant à disposition une importante base de données de profils d’ailes (voir Fig. 3.5) [1]. Conformément aux consignes du *UIUC Airfoil Data Site* [15], le script a positionné les points en plaçant le bord d’attaque à l’origine et le bord de fuite sur l’abscisse à 1, afin de pouvoir comparer le profil extrait avec ceux de la base de données.

Après l’importation des points et leur mise en forme, Airfoiltools a permis d’identifier le profil le plus proche, à savoir le NACA 64(2)-415 (voir Fig. 3.6). Les écarts mineurs observés entre les points extraits et le profil idéal proviennent principalement de la méthode d’acquisition : les coordonnées ont été relevées manuellement sur le plan à l’aide d’environ soixante points, puis reliées entre elles. Ce processus introduit forcément de petites imprécisions par rapport à la courbe théorique. Il faut également souligner que l’avion modélisé n’est pas une reproduction parfaite du plan de référence : certaines valeurs ont dû être approximées en raison des limitations liées au plan et aux outils employés. Toutefois, celui-ci a servi de base solide,

3 Méthodologie générale

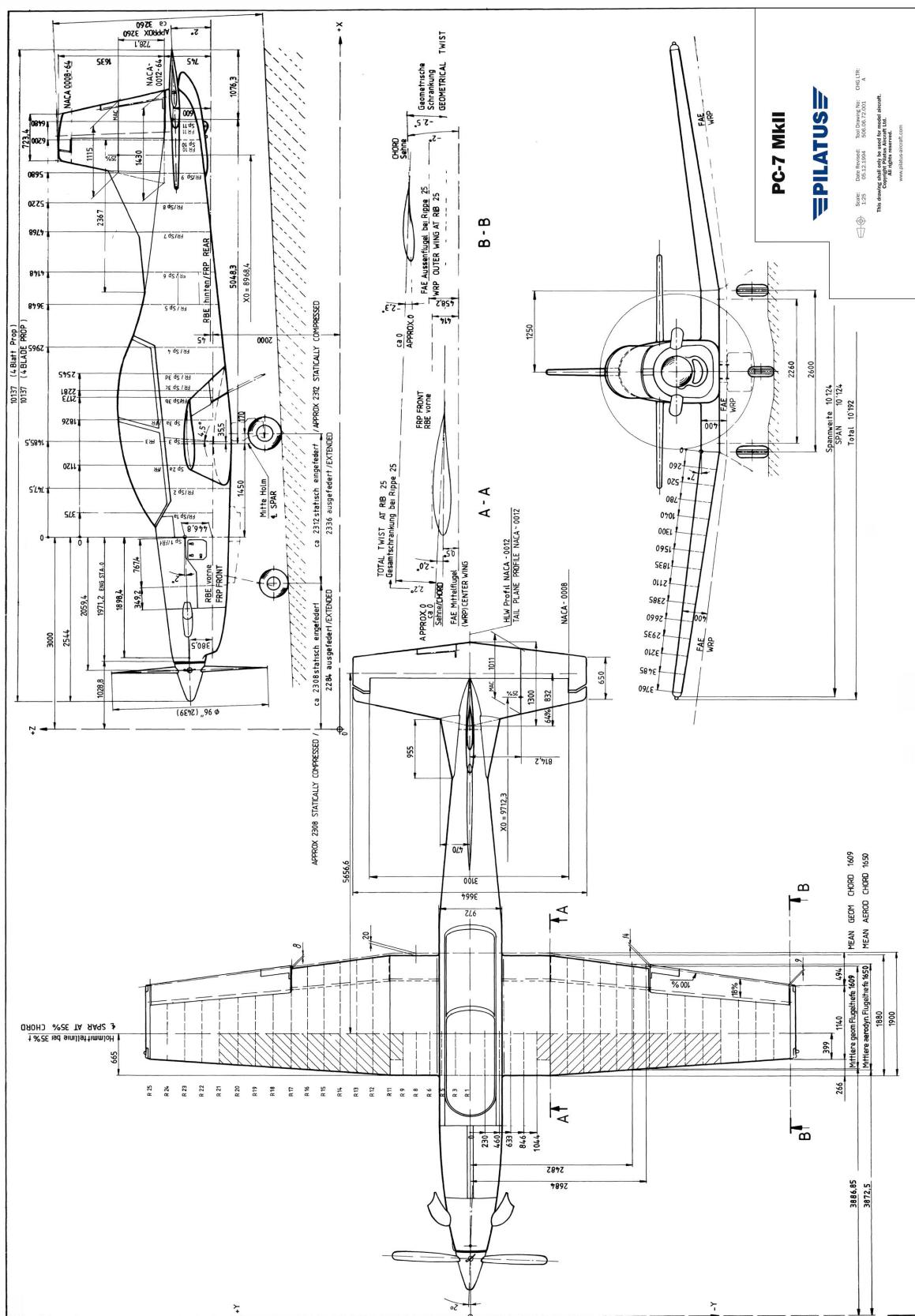


FIGURE 3.4 – Plan de référence utilisé pour l'ensemble des dimensions et des profils aérodynamiques, tiré de [21].

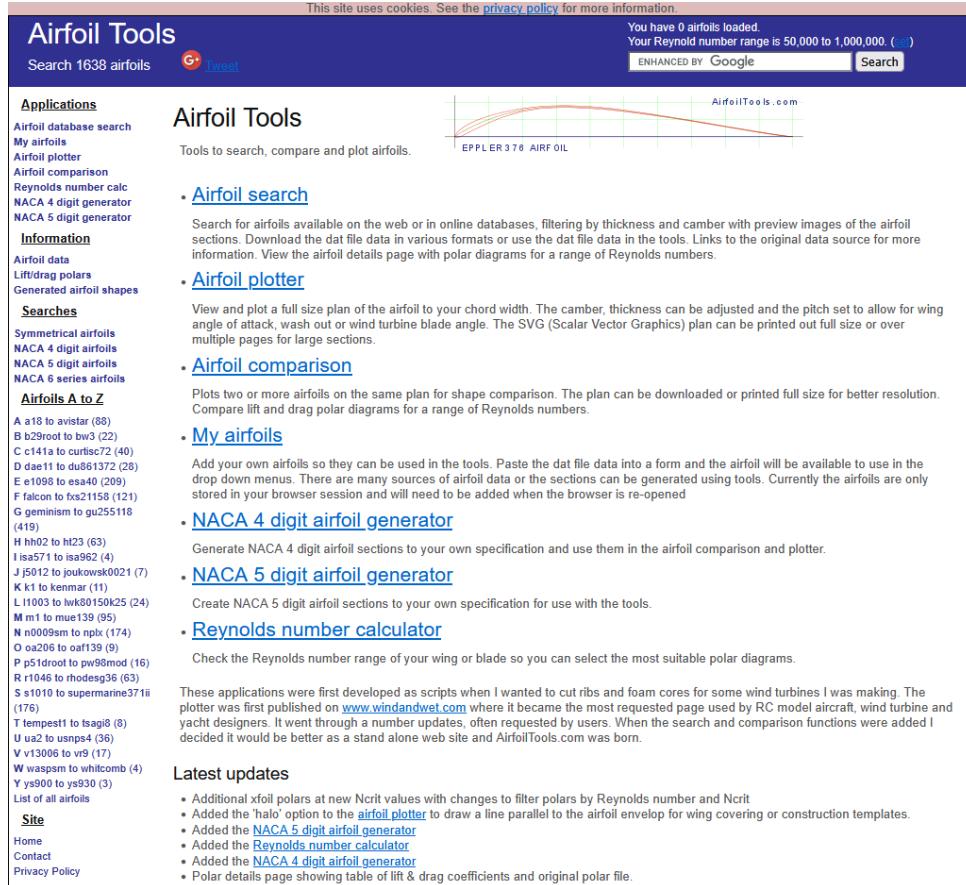


FIGURE 3.5 – Photo d'écran du site de Airfoiltools [1]

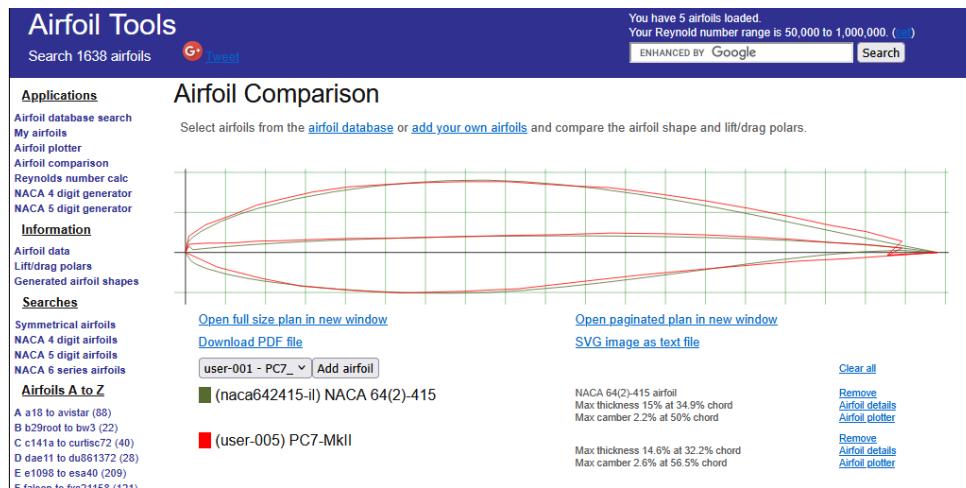


FIGURE 3.6 – Photo d'écran du site de Airfoiltools avec le résultat de la recherche comparative du profil extrais du plan (profil rouge) et le profil trouvé de la base de donnée (profil bleu)



FIGURE 3.7 – Illustration du paramètre à optimiser sur le PC7-MKII, tiré de : [9]

évitant de partir entièrement de zéro, et les écarts constatés n'affectent pas la validité générale du modèle.

3.2.2 Paramétrage des simulations

Les simulations ont été effectuées avec VSPAERO, en considérant l'avion statique et en faisant souffler le flux d'air sur le modèle. Ma toute première interrogation dans ce TM était surtout axée sur la portance ; comment varie la portance en changeant de profil, par exemple. Mais pour faire un modèle réduit fonctionnel il faut prendre en compte les autres variables aérodynamiques : la traînée et la stabilité de l'avion. J'ai donc choisi deux paramètres à optimiser : l'angle des ailes externes avec les ailes horizontales (voir Fig. 3.7) et l'épaisseur du profil.

Les variables testées étaient :

- L'angle des ailes externes : $0, 5, 7, 10, 15, 20^\circ$
- Les profils de l'aile principale : NACA 64(2)-415, 64(2)-425, 64(2)-435
- L'angle du flux vertical : $\alpha \in [-10, 15]^\circ$
- L'angle du flux latéral simulant le roulis : $\beta \in 0, 15, 30^\circ$
- Échelle du modèle réduit : 1 :10

Il y a donc 18 modèles à simuler. J'ai choisi l'échelle pour des raisons pratiques en premier lieu - pouvoir faire rentrer le modèle réduit dans une voiture - et ensuite pour la simplicité du nombre.

J'avais initialement effectué quelques tests en faisant varier uniquement l'angle d'incidence α . Les résultats ont montré que le meilleur compromis, en termes de rapport portance/traînée et de stabilité, correspondait à un angle des ailes externes de 0° . Ce résultat est par ailleurs intuitif, puisqu'une augmentation de l'angle induit une plus grande perte de pression au niveau des bords d'ailes. Afin d'aller plus loin, j'ai voulu étudier l'effet d'un mouvement de roulis en modifiant manuellement l'orientation du modèle entre chaque situation simulée.

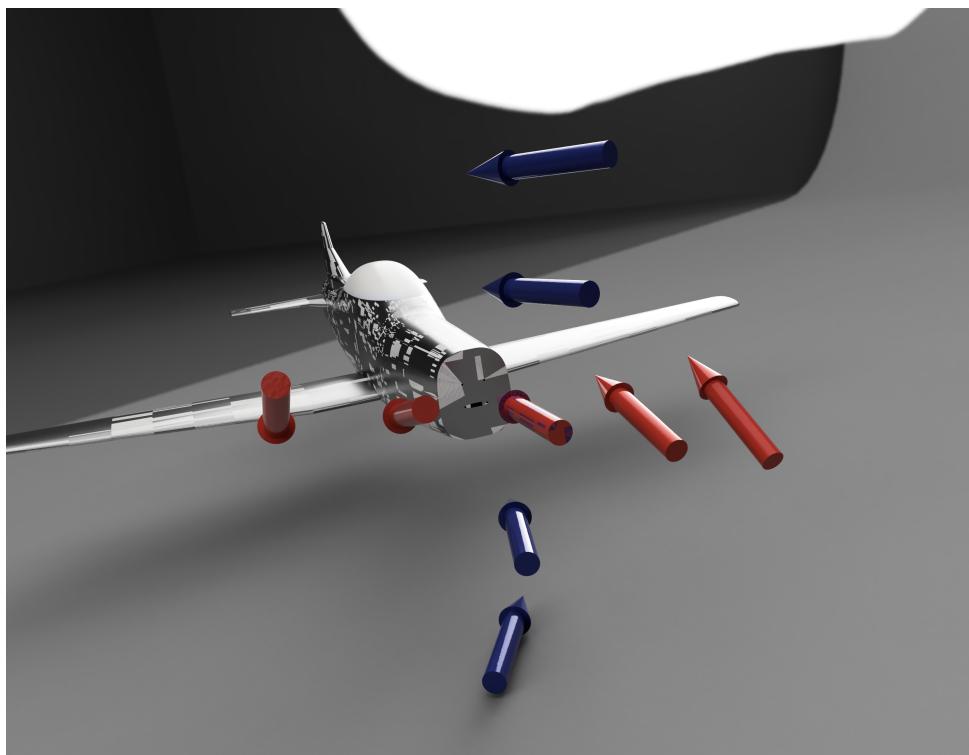


FIGURE 3.8 – Illustration des variations des composantes α en rouge, β en bleu de directions du flux d'air selon un plan vertical au fuselage de l'avion et un plan horizontale à l'avion, réalisé sur fusion 360.

Il existe trois types de rotations avioniques : le roulis, le tangage et le lacet. Le roulis correspond à une rotation autour de l'axe longitudinal (du nez à la queue de l'avion). Le tangage est la rotation autour de l'axe transversal (d'une aile à l'autre) et modifie ainsi l'assiette de l'appareil. Enfin, le lacet est une rotation autour de l'axe vertical, faisant pivoter le nez à gauche ou à droite. Néanmoins, VSPAERO avait des difficultés à interpréter correctement ces modifications de modèle.

Pour représenter cette variation de direction du flux, on introduit l'angle β , qui correspond à une rotation autour d'un axe perpendiculaire à celui de α (voir Fig. 3.8). Comme le modèle est statique et soumis à un écoulement d'air de direction fixe, appliquer un roulis à l'avion revient mathématiquement à imposer un angle β au flux incident dans le repère tridimensionnel. Pour limiter le temps de calcul, j'ai choisi de simuler uniquement trois valeurs de β . Je n'ai pas considéré d'angles négatifs, car le modèle étant parfaitement symétrique, les résultats s'en déduisent directement.

3.2.3 Calcul des paramètres obligatoires pour les simulations

Voici un récapitulatif des différents paramètres de simulations :

- MACH 0.052

3 Méthodologie générale

- ReCref 3.364793929×10^6 (nombre de Reynolds multiplié par la surface de référence, essentiel pour VSPAERO)
- $-10^\circ \leq \alpha \leq 15^\circ$
- $\beta \in \{0, 15, 30\}$
- Profils : NACA 64(2)-415, NACA 64(2)-425, NACA 64(2)-435
- Angles des ailes externes $\in \{0, 5, 7, 10, 15, 20\}$
- Modèle à l'échelle 1 :10

Le MACH est simplement la vitesse du flux *au loin* exprimée en un facteur de la vitesse du son. Cette vitesse découle de la démonstration théorique suivante. Comme les simulations sont effectuées à vitesse de croisière constante. De par ce référentiel, il n'y a par définition pas d'accélération et donc, par la deuxième loi de Newton, une somme des forces nulle. Dans le cas de l'avion, il n'y a que 4 forces qui agissent sur celui-ci : la portance, la force de gravité, la poussée et la traînée. Comme la somme de toutes ces forces est nulle et qu'elles sont, lorsque l'avion vole parallèlement au sol - *en croisière*, toutes perpendiculaires, elles sont parfaitement égales deux à deux. Ainsi, la portance est égale à la force de gravité :

$$\frac{1}{2}\rho S C_L v^2 = mg \quad (3.1)$$

Or la vitesse du flux d'air, soit de l'avion, apparaît dans l'équation de la portance. Il suffit de l'isoler :

$$v = \sqrt{\frac{2mg}{\rho S C_L}} \quad (3.2)$$

Les profils NACA de la série 6 précisent le coefficient de portance théorique associé, indiqué juste après le tiret et exprimé en dixièmes (voir Sect. 2.11). Les trois profils testés comportent tous la valeur 4, ce qui correspond à un coefficient de portance idéal de $C_L = 0,4$. La masse volumique de l'air est également connue, puisque les simulations sont effectuées dans des conditions standards : air à pression normale, température de 20 °C, viscosité dynamique $\mu = 1,8 \times 10^{-5} \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$ et masse volumique $\rho = 1,293 \text{ kg/m}^3$ d'après les tables de référence [12]. La surface alaire S est fournie directement par OpenVSP et vaut 0,162 m².

Pour estimer la masse totale du modèle réduit, le fichier a été tranché avec Ultimaker Cura, un logiciel indispensable pour convertir un modèle STL en instructions d'impression 3D. En appliquant un remplissage de 8%, Cura a fourni une masse estimée de 200 g pour la structure en plastique, calculée à partir de la densité du matériau extrudé. En tenant compte des composants électroniques, évalués entre 310 g et 410 g, la masse totale de l'avion atteint environ 0,61 kg.

Autre paramètre important, les angles des ailes externes ne peuvent pas être rentrés directe-

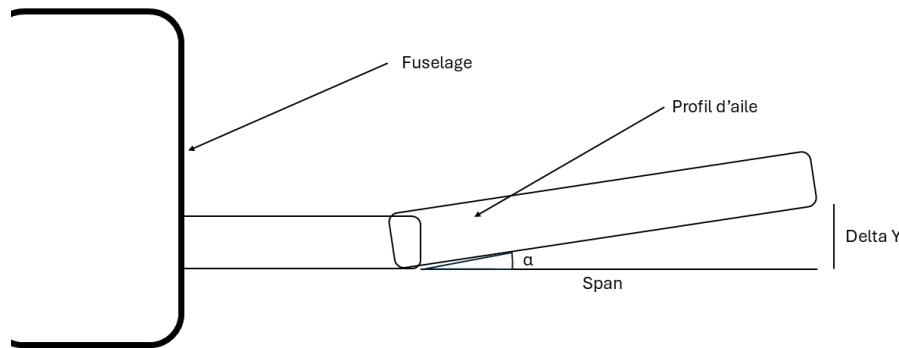


FIGURE 3.9 – Illustration du paramètre permettant d’ajouter un angle aux ailes externes

ment dans OpenVSP. Il existe un paramètre précis nommé $\Delta Y/c$ qui se calcule en regardant la Fig. 3.9. On peut ainsi dire que tangente de α est égale à ΔY divisé par $Span$:

$$\tan(\alpha) = \frac{\Delta Y}{Span} \quad (3.3)$$

Or OpenVSP demande ΔY sur C nous faisons donc :

$$\frac{\tan(\alpha)Span}{C} = \frac{\Delta Y}{C} \quad (3.4)$$

De par le modèle complet contenu dans OpenVSP, les valeurs demandées sont calculables.

Pour finir, ReCref veut simplement dire de calculer le Reynolds multiplié par la Surface de référence notée Cref.

$$ReCref = \frac{\rho VL^2}{\mu} Cref \quad (3.5)$$

Tout ceci est connu à ceci près que L est simplement l’envergure des ailes.

3.2.4 Automatisation des simulations

Le principal problème rencontré au départ était l'impossibilité d'automatiser les simulations dans OpenVSP. Comme une simulation complète nécessitait environ six heures de calcul sur mon ordinateur, lancer manuellement les 18 cas d'étude aurait demandé un temps excessif, d'autant que je ne pouvais pas rester en permanence devant l'écran pour surveiller chaque exécution. Une fois mon modèle finalisé et après m'être formé grâce à *OpenVSP Ground School* [8], j'ai entrepris de chercher une solution dans le Google Group d'OpenVSP [5], mais aucune méthode partagée n'existe pour automatiser OpenVSP en Python. De même, ni les forums ni les tutoriels disponibles en ligne n'apportaient de réponse satisfaisante.

3 Méthodologie générale

J'ai alors consulté la documentation officielle de *OpenVSPpytools*, la librairie Python permettant d'interagir avec OpenVSP [3], mais elle ne fournissait pas les indications dont j'avais besoin. Ce n'est qu'en explorant directement les fichiers d'installation d'OpenVSP que j'ai découvert que le lancement d'une simulation depuis l'interface graphique correspondait en réalité à l'exécution d'une commande dans le terminal. Cette observation m'a fourni une piste pour exécuter VSPAERO de manière automatisée. Restait toutefois un dernier obstacle : le solveur VSPAERO requiert des fichiers préparatoires spécifiques, sans lesquels il échoue dès le démarrage. La solution a consisté à utiliser un fichier *.vspscript*, exécutable via une fonction de la librairie OpenVSPpytools, permettant de générer automatiquement ces fichiers. Sur cette base, j'ai pu mettre en place un script Python complet d'automatisation, détaillé en Annexe 7.2.

1. Création de dossiers pour chaque simulation
2. Copie des fichiers du modèle et des paramètres dans chaque dossier
3. Calcul des géométries nécessaires pour VSPAERO
4. Lancement automatique des 18 cas de simulation
5. Extraction et compilation des résultats dans une base de données

Chaque simulation fournit des coefficients aérodynamiques tels que la portance C_L , la traînée C_D et la stabilité selon la norme du vecteur des coefficients des moments de forces C_M ainsi que les composantes de ce vecteur. La comparaison des résultats pour différentes configurations a permis de déterminer la combinaison d'angles et de profils offrant le meilleur rapport L/D pour le modèle réduit ainsi que les meilleures courbes de stabilités. Les variations du flux d'air simulant le roulis ont également été prises en compte pour tous les coefficients.

Les simulations sont paramétrées et tout le processus du script a pris 12 jours, 5 heures et 36 minutes. Pour finir toute cette méthode, je termine par un script allant chercher tous les résultats dans les dossiers, puis constitue une base de données (voir Annexe 7.3). La suite appartient à l'analyse de ces données.

4 Résultats et analyse des simulations

Après avoir procédé à la modélisation de plusieurs profils d'aile et de configurations géométriques, les simulations numériques ont permis d'évaluer et de comparer la performance aérodynamique de chaque modèle. L'objectif principal est d'identifier, parmi toutes les variantes testées, celles qui offrent le meilleur compromis entre portance, traînée et stabilité en vol.

La suite du chapitre présente donc d'abord l'analyse de la convergence des calculs, étape indispensable pour garantir la fiabilité des tendances obtenues. Ce travail permet ensuite d'expliquer la méthodologie appliquée pour traiter les valeurs aberrantes et de démontrer comment les résultats ont été interprétés et comparés en vue d'un classement pertinent des différents modèles.

4.1 Convergence des calculs

Pour commencer, il est essentiel de bien comprendre le mécanisme d'analyse utilisé par VSPAERO. Comme expliqué dans la section 2.8.1, la répartition de pression autour d'un profil permet de calculer les forces aérodynamiques. Le rôle de VSPAERO est donc précisément de déterminer cette répartition de pression. Pour cela, on génère un maillage qui remplace le modèle 3D de résolution théoriquement infini, afin d'éviter d'avoir une infinité de points où calculer la pression. Les points de pression ainsi créés ne sont pas indépendants : ils sont liés entre eux par des équations mathématiques. Concrètement, on attribue une variable à chaque sommet du maillage, ce qui conduit à l'obtention d'un système d'équations de très grande taille, comportant de nombreuses inconnues. C'est pour cette raison qu'une grande puissance de calcul est nécessaire, bien plus que ce qu'il serait possible de réaliser à la main.

Il existe plusieurs méthodes pour résoudre un tel système d'équations. VSPAERO utilise une méthode vectorielle, dont je ne détaillerai pas ici le fonctionnement. Quel que soit l'algorithme, toutes ces approches reposent sur un principe commun : un processus itératif. Celui-ci consiste à répéter les calculs de manière successive, en affinant progressivement la solution jusqu'à atteindre une valeur stable et unique. Étudier ces itérations permet de vérifier si le résultat est bien convergent, ou encore d'exploiter le graphe pour améliorer la valeur finale. Lors de mes premières analyses, j'ai constaté que certaines valeurs étaient fortement erronées, s'écartant nettement du reste des résultats. Représenter les valeurs en fonction de leur numéro d'itération s'est donc révélé indispensable pour comprendre l'origine de ces écarts.

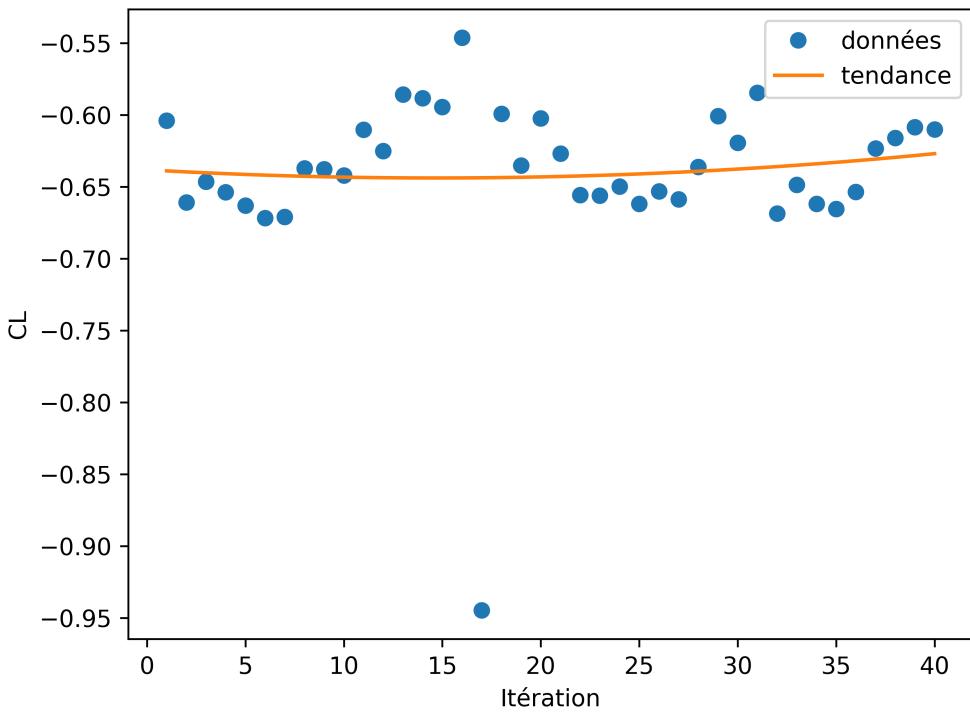


FIGURE 4.1 – Graphe des valeurs du coefficient de portance obtenus en fonction de leurs numéros d’itérations pour le modèle d’angle de l’aile externe à 0°, d’épaisseur d’aile de 10% et d’ α et β à 0°.

Comme illustré à la Figure 4.1, la valeur obtenue à l’itération 17 est complètement aberrante. Ce n’est d’ailleurs pas un cas isolé. J’ai choisi cet exemple particulier car il reste lisible visuellement, là où d’autres écarts sont si extrêmes qu’ils forcent une échelle excessive du graphique. Dans le cas présent, seule l’itération 17 est fausse, ce qui ne pose pas vraiment problème en soi puisque je considérais toujours la dernière itération, mais dans le cas global on voit bien plusieurs valeurs aberrantes à la dernière itération. On peut quand même observer une bonne répartition d’erreurs dans toutes les valeurs brutes (voir Fig. 4.2).

Les courbes d’itérations peuvent suivre différents types de tendances convergentes. La plus courante correspond à une fonction du type $f(x) = k - e^{-x}|k \in \mathbb{R}$, mais il existe aussi des formes oscillantes où les écarts consécutifs diminuent progressivement [18]. Dans mon cas, la tendance observée semble hybride entre ces deux comportements, comme on le voit sur la Figure 4.1. Toutefois, l’analyse des deltas absolus montre qu’ils ne diminuent pas réellement de manière perceptible, et ne suivent pas non plus une décroissance exponentielle claire.

On pourrait penser que le nombre d’itérations est tout simplement insuffisant, et que la convergence n’a pas encore été atteinte. Mais cet argument se heurte à deux réalités : d’une part, l’ensemble des simulations a déjà nécessité un total de 12 jours de calcul ; d’autre part, plusieurs références soulignent les limites intrinsèques de VSPAERO. En particulier, le master *Software testing : VSPAERO* [20] et l’article *Evaluation of vspaero analysis capabilities for*

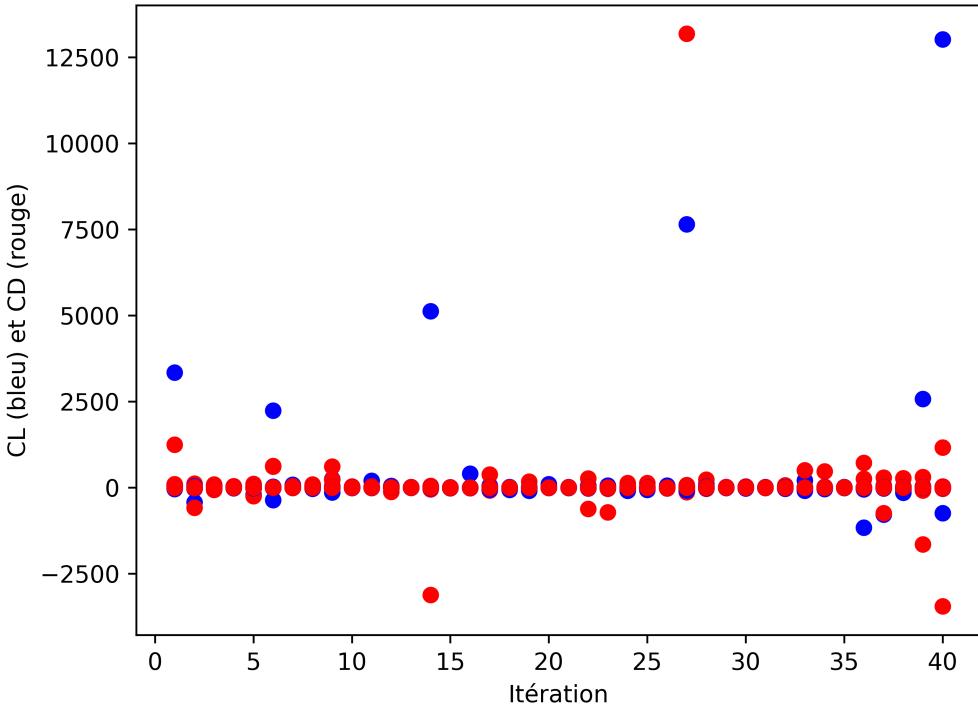


FIGURE 4.2 – Graphe des valeurs du coefficient de portance et de traînée obtenus en fonction de leurs numéros d’itérations pour toutes les valeurs brutes.

conceptual design of aircraft with propeller-blown wings [23] concluent vers la même chose : VSPAERO est un bon outil pour obtenir rapidement des résultats d’ordre de grandeur, mais il ne permet pas une grande précision. De plus, les deux textes recommandaient le même nombre d’itérations, car en augmenter le total entraînait une perte de précision encore plus marquée. Il faut enfin préciser que, contrairement aux auteurs de ces études, je n’avais pas accès à de puissants data centers, mais uniquement aux ressources limitées de mon PC personnel.

Ces « erreurs » produites par VSPAERO rendent donc risqué l’usage direct des dernières itérations : elles peuvent biaiser les tendances et conduire à de mauvais choix de modèle. C’est pourquoi j’ai imaginé une méthode visant à corriger mathématiquement les itérations aberrantes, afin de sécuriser les résultats sans pour autant les manipuler de manière artificielle.

La méthode que j’ai adoptée repose sur une règle simple. Pour chaque itération successive, je vérifie que l’écart avec la valeur précédente reste inférieur à trois fois la moyenne des écarts calculés sur les valeurs jugées correctes. Si cette condition est respectée, la valeur est conservée et utilisée lors du calcul de la courbe de tendance, sinon elle est exclue. Pour finir, si l’écart entre la valeur de l’itération 40 et la tendance excède ce seuil, cette valeur est alors remplacée par la prédiction issue de la courbe. Cette stratégie permet de ne retenir que les données cohérentes dans l’établissement d’une tendance globale, tout en minimisant l’impact des points aberrants sur le classement des modèles. La procédure complète est détaillée dans le code

4 Résultats et analyse des simulations

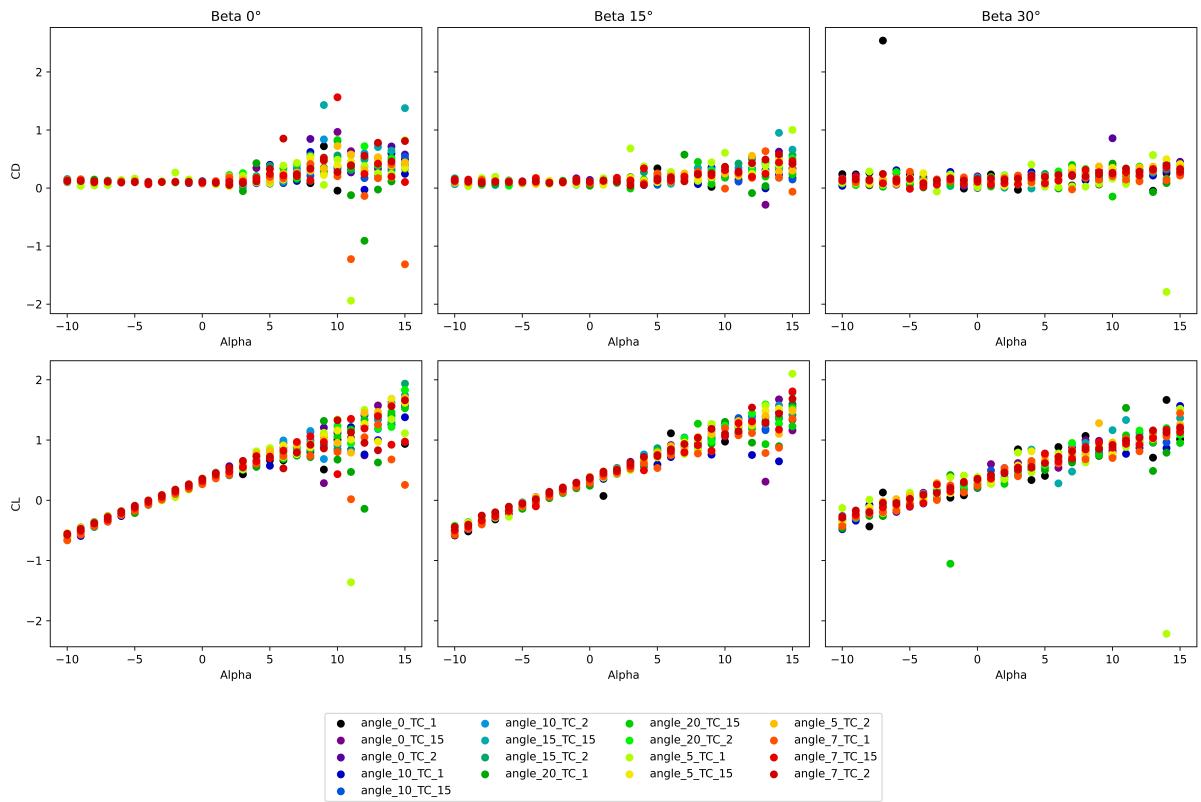


FIGURE 4.3 – Graphes des coefficients de portance et de traînée en fonction de alpha pour chaque beta, après correction des données. Les points de couleurs différents incarnent les différents modèles. Le nombre juste après "angle" est l'angle des ailes externes (voir Fig. 3.7), le second nombre est l'épaisseur du profil en pourcentage de la corde - *Thickness* en anglais.

fourni en Annexe 7.5. Cette approche s'est avérée efficace, car elle limite les biais liés aux erreurs ponctuelles tout en garantissant une cohérence mathématique des résultats (voir Fig. 4.3). Elle demeure cependant dépendante de la qualité et de la régularité des premières itérations : si celles-ci sont trop éloignées, la correction ne peut plus être pleinement fiable. Après comparaison avec la Figure 4.2, on observe moins de divergences sur les premières valeurs. On peut donc raisonnablement appliquer cette méthode.

4.2 Résultats comparatifs

La Figure 4.3 illustre les variations des coefficients de portance (C_L) et de traînée (C_D) en fonction de l'angle d'incidence alpha, pour chaque valeur de beta considérée dans l'analyse, après correction des données initiales. Chaque point représenté sur la figure possède une couleur distincte permettant d'identifier le modèle associé. Grâce au processus de correction des données décrit précédemment, la figure ne contient plus de valeurs aberrantes : l'ensemble des points sont dans des intervalles cohérents du point de vue physique. Les points sont tous compris entre -2 et 2. On peut ainsi observer comment la portance et la traînée évoluent avec alpha pour chaque beta, et interpréter les écarts de comportement liés aux variations d'angle et d'épaisseur.

Cette visualisation a surtout permis d'évaluer si la méthode de correction était effective sur les valeurs. Globalement, une plus grande dispersion des valeurs est observée pour des angles d'incidence α supérieurs à 5° , traduisant des comportements aérodynamiques plus complexes, de par les valeurs erronées générées par VSPAERO.

La Figure 4.4 présente les courbes de tendance du rapport portance-traînée (L/D) et du coefficient de moment C_M en fonction de l'angle d'attaque α , pour différentes valeurs de β . Ces deux grandeurs sont essentielles à optimiser : L/D car il définit directement l'efficacité aérodynamique globale, et C_M car il renseigne sur la stabilité général. L'objectif est donc d'obtenir des courbes régulières, physiquement cohérentes, permettant d'identifier les configurations les plus performantes et les plus stables. On constate cependant que certaines courbes présentent des asymptotes, ce qui rend nos fonctions non continues. Cet effet est visible par exemple pour $\beta = 0$, où la courbe L/D présente une asymptote vertical autour de $\alpha = 15^\circ$. Or, en aérodynamique subsonique et dans un environnement de simulation isolé, ce type de comportement n'est pas physiquement réaliste. En particulier, pour le rapport L/D soit C_L/C_D , une asymptote n'apparaît que si $C_D \leq 0$, ce qui n'est pas possible dans la réalité. Cela découle du fait que lors du calcul de tendance, la courbe de C_D obtenue passe ponctuellement par des valeurs nulles ou négatives (voir Fig. 4.3). Une telle situation n'a pas de sens physique. Un coefficient de traînée nul reviendrait à dire qu'aucune force de frottement n'agit sur l'aéronef, ce qui est impossible compte tenu de son interaction permanente avec l'air. Une valeur négative impliquerait qu'au lieu de s'opposer au mouvement, la traînée aiderait celui-ci, ce qui est contradictoire par définition. Pour corriger ce problème et obtenir des courbes physiquement viables, j'ai choisis une approche consistant à calculer la tendance sur le logarithme des valeurs, puis à réappliquer une exponentielle pour retrouver l'échelle originale. Les éventuelles valeurs négatives sont traitées en les remplaçant par la moyenne des valeurs voisines. Cette méthode garantit des courbes strictement positives, continues et现实的. Les résultats corrigés sont présentés dans la Figure 4.5, et la procédure correspondante est donnée dans le code de Annexe 7.7.

On dispose désormais de courbes lissées, offrant une meilleure lisibilité graphique et reflète très bien la réalité physique. Toutefois, l'ensemble reste composé de 17 courbes distinctes (une simulation n'a pas fonctionné voir Sec. 4.3.3), ce qui rend la comparaison directe peu pratique. Pour synthétiser l'information, j'ai donc calculé pour chaque valeur de β l'intégrale du rapport L/D ainsi que la dérivée de C_M , puis j'ai utilisé ces grandeurs comme critères de classement. L'idée a été d'additionner les aires caractéristiques pour chaque β , afin de privilégier à la fois une maximisation de l'efficacité aérodynamique (via L/D) et une minimisation des variations de moment (via C'_M). Je rajoute encore la minimisation des dérivées de C_{Mx} , C_{My} et C_{Mz} afin d'avoir une précision de plus sur la réel stabilité de l'avion (voir Table 4.1). Cette procédure permet ainsi de hiérarchiser les configurations selon des critères cohérents de performance et de stabilité. Le code correspondant est présenté en Annexe 7.8.

4 Résultats et analyse des simulations

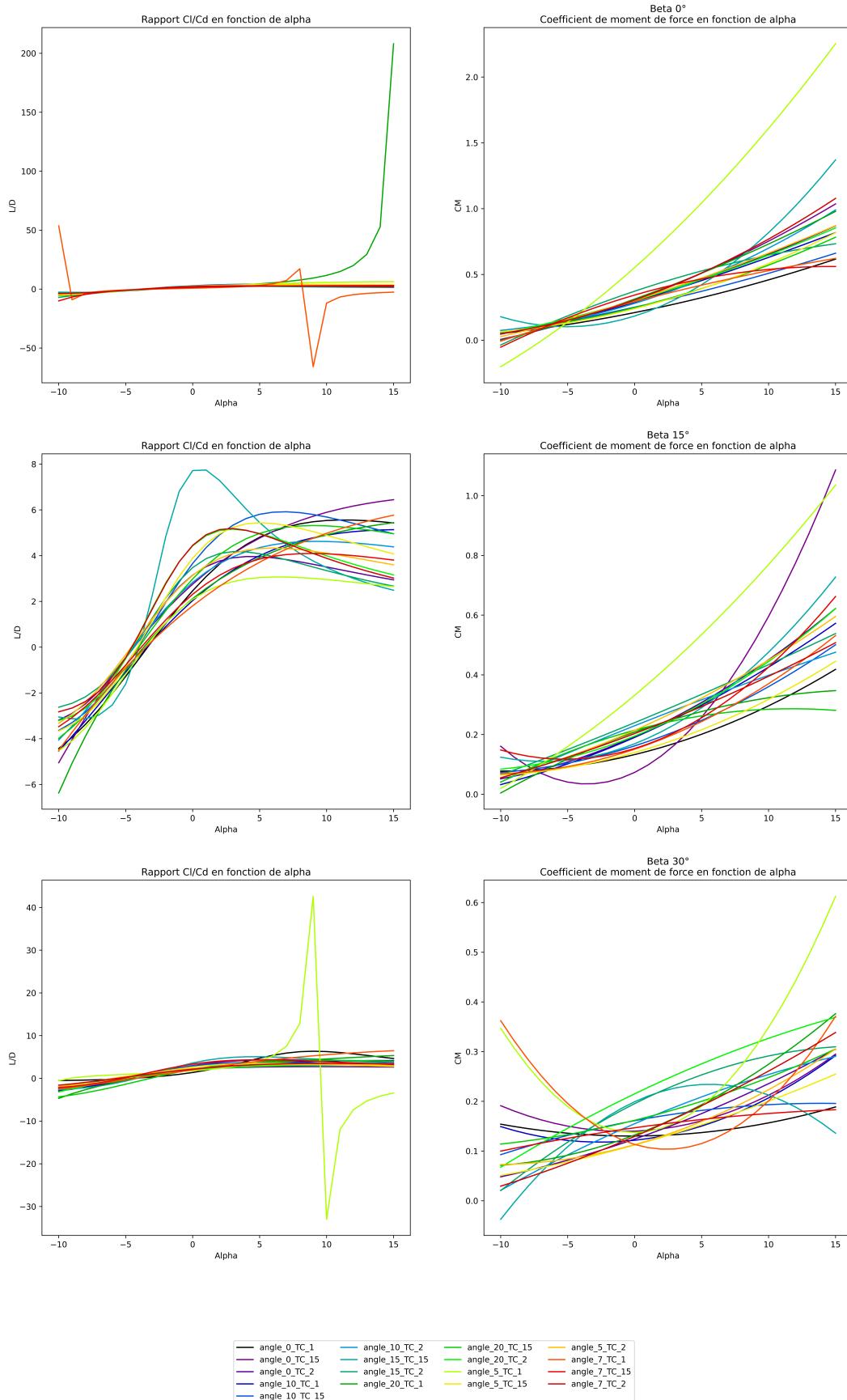


FIGURE 4.4 – Planche de graphes pour chaque beta (lignes) (variation de direction du flux d'air d'axe vertical, voir Fig. 3.8) des fonctions de tendances L/D (première colonne), C_M (dernière colonne) en fonction de alpha.

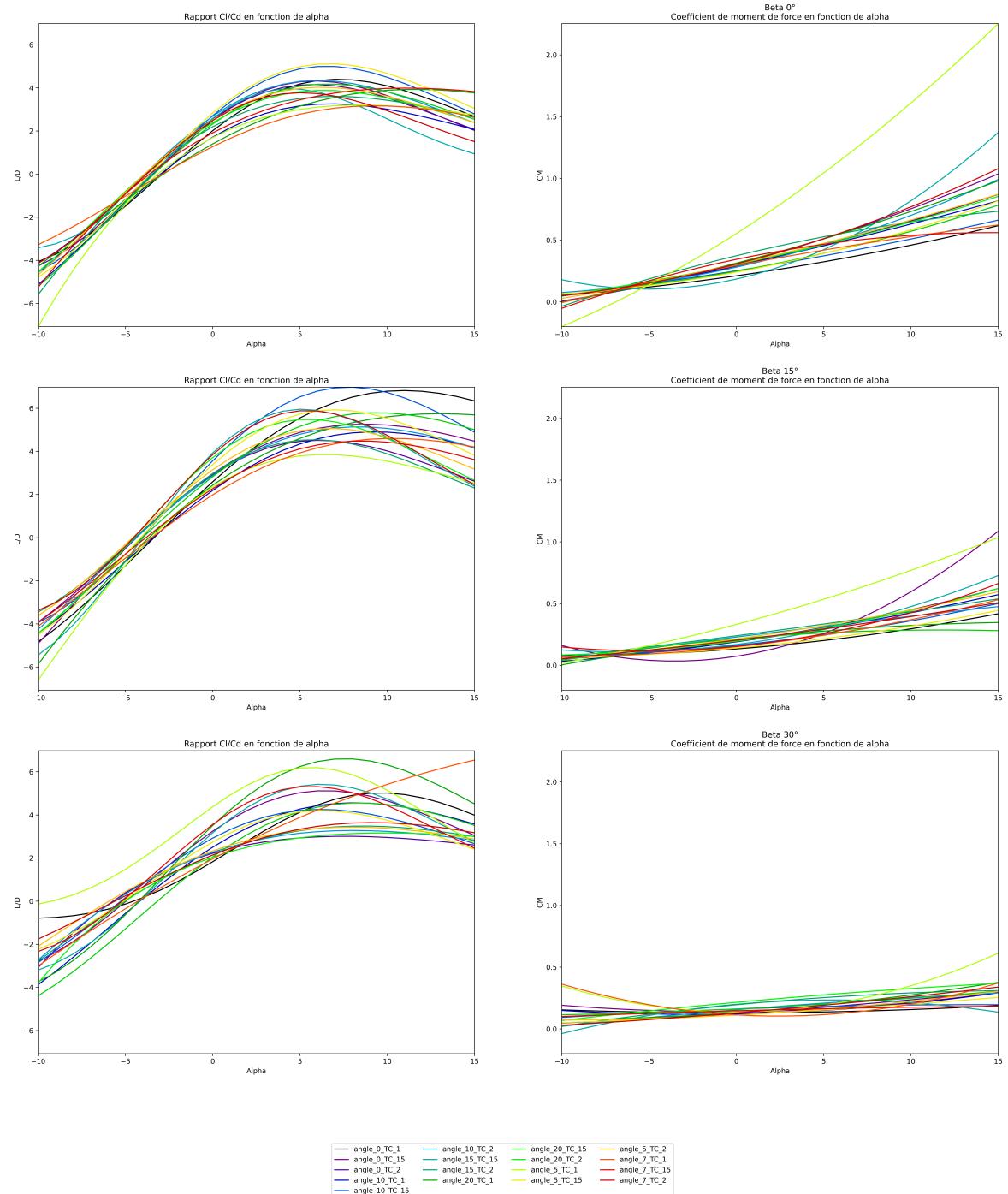


FIGURE 4.5 – Planche de graphes pour chaque beta (lignes) des fonctions de tendances L/D , la norme du vecteur C_M en fonction de α avec la méthode décrite dans la section 4.2.

Modèle	$\int_{-10}^{15} L/D(\alpha)d\alpha$ [$\beta = 0, 15, 30$]	$\int_{-10}^{15} CM(\alpha)d\alpha$ [$\beta = 0, 15, 30$]	$\int_{-10}^{15} CM'_x(\alpha)d\alpha$ [$\beta = 0, 15, 30$]	$\int_{-10}^{15} CM'_y(\alpha)d\alpha$ [$\beta = 0, 15, 30$]	$\int_{-10}^{15} CM'_z(\alpha)d\alpha$ [$\beta = 0, 15, 30$]	position (rang moyen)
angle_10_TC_15	192.64	1.13	0.04	1.36	-0.01	3.7
angle_5_TC_15	178.71	1.33	0.04	1.38	-0.01	4.3
angle_0_TC_1	178.34	0.94	0.08	0.96	0.06	5.0
angle_0_TC_15	172.29	2.06	0.04	1.29	-0.01	6.0
angle_10_TC_2	156.07	1.62	0.04	1.69	-0.01	8.0
angle_20_TC_1	176.86	1.63	-0.04	0.77	0.26	8.3
angle_5_TC_2	156.02	1.61	0.03	1.7	-0.01	8.7
angle_7_TC_2	157.29	1.79	0.03	1.78	0.04	8.7
angle_7_TC_15	160.6	1.2	0.05	1.57	-0.01	9.0
angle_15_TC_15	159.19	1.96	0.03	1.91	0.06	10.0
angle_20_TC_15	154.59	1.15	0.05	1.03	0.06	10.0
angle_20_TC_2	146.41	1.64	0.03	1.66	0.03	10.3
angle_0_TC_2	140.06	1.65	0.04	1.79	-0.06	10.7
angle_5_TC_1	152.65	3.74	0.05	1.53	0.48	11.7
angle_7_TC_1	144.38	1.1	0.07	0.91	0.25	12.0
angle_10_TC_1	133.52	1.5	0.06	1.16	0.22	13.3
angle_15_TC_2	136.14	1.53	0.04	1.53	0.0	13.3

TABLE 4.1 – Tableau des performances générales pour chaque modèle avec sa moyenne des positions de performance pour chaque beta (rang moyen).

4.3 Discussion

La discussion qui suit vise avant tout à préciser la portée des résultats, à mettre en évidence leurs limites, et à clarifier les choix méthodologiques faits tout au long de cette analyse. Je vais donc revenir sur le régime d'écoulement et les différences marquantes entre le modèle original et le modèle réduit, expliquer le sens du classement proposé, et enfin soulever les interrogations et les zones d'incertitude qui subsistent après traitement des données.

4.3.1 Comparaison des Reynolds avec le modèle original

La première chose à noter, c'est que le modèle réduit 1 :10 et l'avion original évoluent dans des régimes de Reynolds complètement différents, presque 2 ordres de grandeur d'écart. L'écoulement attendu autour du modèle réduit est donc bien moins turbulent et plus proche du laminaire, contrairement au plein échelle, où l'on aura des phénomènes de transition et de turbulence bien plus marqués. Ces phénomènes se manifestent lorsque, sous l'effet de la vitesse et de la pression du fluide, la viscosité n'est plus suffisante pour maintenir un écoulement ordonné : la couche limite devient instable, et l'interaction entre les particules du fluide entraîne alors des mouvements chaotiques caractérisés par de fortes variations locales de vitesse et de pression, ce qui marque le passage de l'écoulement laminaire à turbulent, souvent décrit par une augmentation significative du nombre de Reynolds. Pour le modèle taille réelle, le calcul du Reynolds s'effectue comme suit :

$$\frac{1,293 * 1,642 * 129,126}{1,8 \times 10^{-5}} = 1,523 \times 10^7 \quad (4.1)$$

et pour le modèle réduit :

$$\frac{1,293 * 0,1642 * 16,89}{1,8 \times 10^{-5}} = 1,9921 \times 10^5 \quad (4.2)$$

On obtient donc un rapport de Reynolds de 76,5, soit presque 2 ordres de grandeurs de différence. Ce décalage en régime rend impossible toute extrapolation directe des résultats vers l'avion grandeur nature : les graphiques et les courbes obtenus ici sont propres au modèle réduit et ne sont pas représentatifs du comportement physique du modèle réel.

Pour imaginer ce qu'il faudrait mettre en œuvre pour conserver le même Reynolds entre les deux configurations, il suffit de réécrire la formule à partir des données connues :

$$\frac{1,523 \times 10^7 * 1,8 \times 10^{-5}}{1,293 * 16,89} = 12,5m \quad (4.3)$$

Il faudrait donc construire un modèle réduit avec une envergure de 12,5m, ce qui est totalement irréaliste. Autre possibilité, booster la vitesse de vol pour palier à la petite taille du modèle :

$$\frac{1,523 \times 10^7 * 1,8 \times 10^{-5}}{1,293 * 0.1642} = 1291,22 \text{ m/s} \quad (4.4)$$

Soit, concrètement, faire voler le modèle à quasiment Mach 13. On est là dans le domaine du non-sens expérimental pour un profil d'avion conventionnel. Pour toute la suite, je choisis donc de baser l'interprétation uniquement sur les résultats obtenus avec le modèle réduit, qui reste le plus pertinent dans le cadre des futurs essais et des simulations menées.

4.3.2 Analyse des formes des courbes de tendances

Je n'ai aucune expérience préalable me permettant de privilégier un paramètre plutôt qu'un autre dans mon analyse. On sait cependant que plus le rapport L/D est élevé, moins l'avion consomme d'énergie et meilleure est son autonomie. Un bon L/D est aussi synonyme de meilleures performances en vol, notamment lors de figures acrobatiques ou de manœuvres prolongées. À l'inverse, le coefficient de moment C_M se traduit dans la maniabilité du modèle réduit : plus C_M reste faible et stable sur la plage d' α , plus le contrôle en vol s'en trouve simplifié, ce qui est essentiel pour un avion télécommandé. J'ai donc choisi d'attribuer un poids équivalent à ces deux critères – autonomie et maniabilité – dans mon classement.

Les courbes obtenues pour L/D (Fig. 4.5) rappellent les formes classiques décrites dans *Aerodynamic Characteristics of a Number of Modified NACA Four-digit-series Airfoil Sections* [19] : une augmentation progressive de l'efficacité aérodynamique avec α , suivie d'un léger palier, puis d'une chute qui correspond au décrochage de l'aile. Cette évolution est parfaitement cohérente avec la physique du vol : le profil gagne en portance à mesure que l'incidence croît, jusqu'au point où les écoulements se séparent massivement du profil, provoquant une diminution rapide du rapport portance-trainée. On retrouve donc des tendances typiques, ce qui conforte la validité relative des résultats malgré les limites du simulateur.

Concernant le coefficient de moment C_M , il correspond ici à la norme du vecteur moment aérodynamique (C_{mx} , C_{my} , C_{mz}). Contrairement à l'analyse classique qui focalise souvent sur la composante tangentielle C_{my} (indiquant la stabilité longitudinale par le signe de sa dérivée par rapport à α), la valeur considérée est une grandeur scalaire positive définie par la norme vectorielle. Cette norme traduit globalement la magnitude de la charge aérodynamique de moment mais ne renseigne pas sur la direction ni sur le signe des moments qui sont pourtant déterminants pour la stabilité effective. Cela reste un paramètre à prendre en considération dans l'optique d'avoir un modèle le plus polyvalent possible.

La tendance générale observée, montrant des pentes positives de $\frac{dC_M}{d\alpha}$, indique une augmentation globale de la charge aérodynamique du moment de force à mesure que l'angle d'attaque s'éloigne de la position d'équilibre. Toutefois, cette information ne suffit pas pour mesurer la stabilité longitudinale ou la maniabilité, car elle masque la contribution et le comportement différencié des composantes vectorielles.

Par conséquent, bien que la norme C_M soit un indicateur utile du niveau global des forces aé-

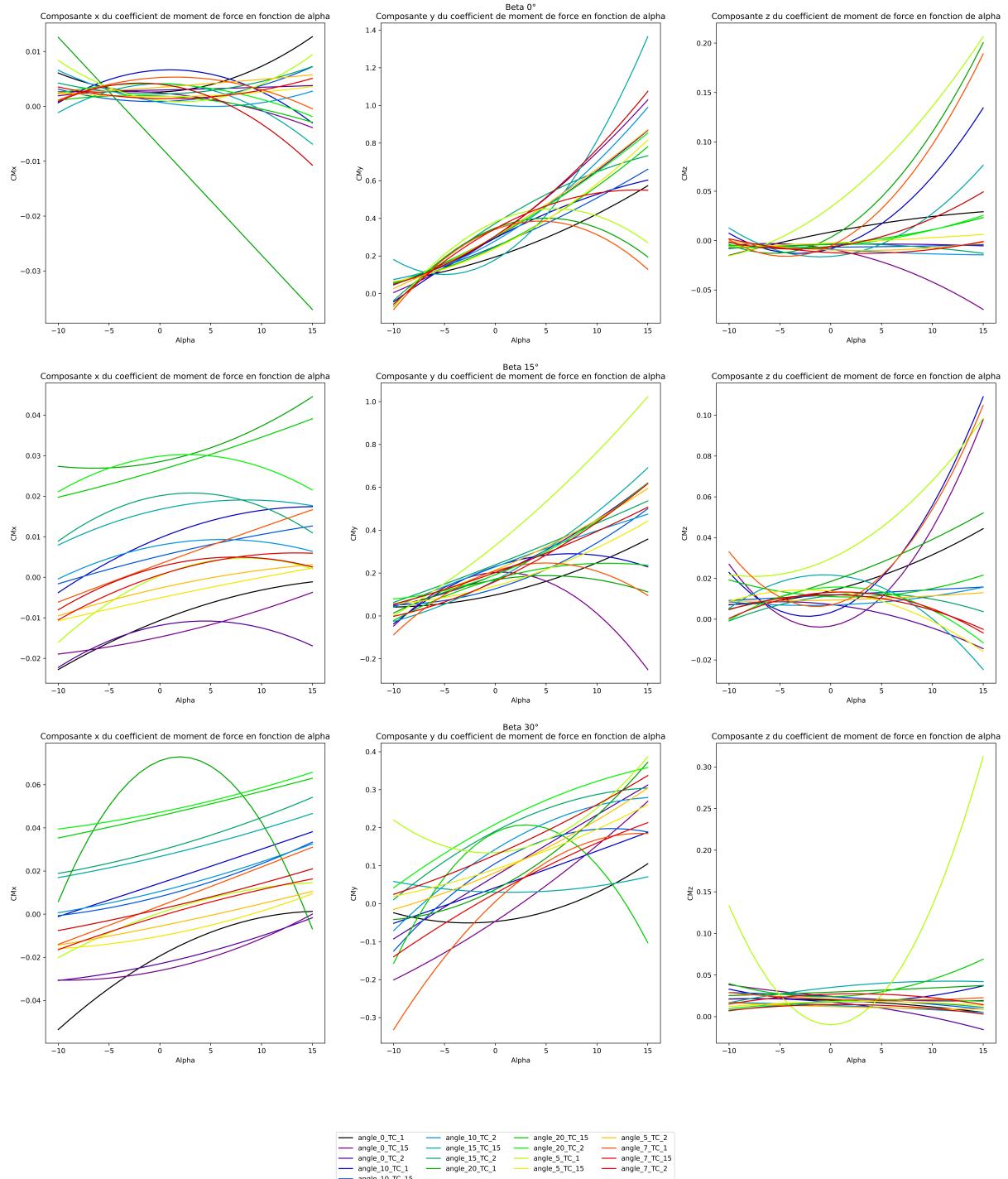


FIGURE 4.6 – Planche de graphes pour chaque β (lignes) des fonctions de tendances des composantes de C_M en fonction de α .

rodynamiques, son interprétation doit être complétée par une analyse détaillée des composantes individuelles (C_{mx} , C_{my} , C_{mz}), notamment de C_{my} , pour apprécier la stabilité et le comportement dynamique du modèle en vol. Pour obtenir le tableau 4.1, j'ai pris en compte la minimisation des aires sous la courbes des dérivées de chaque composante de C_M . La figure 4.6 montre ces différentes courbes des composantes du coefficient de moment de force.

On remarque enfin que, contrairement à C_M , les différences sur L/D entre modèles paraissent globalement moins marquées. Cela signifie qu'en termes d'efficacité énergétique, plusieurs configurations se valent, et que la vraie clé de tri repose davantage sur le critère de stabilité. De ce fait, il est donc possible d'exclure certains modèles en se basant sur leurs courbes C_M non caractéristiques : `angle_5_TC_1` (Fig. 4.5 courbe $\beta = 0^\circ$), `angle_0_TC_2` (courbe $\beta = 15^\circ$) et `angle_10_TC_2` (courbe $\beta = 0^\circ$), cette dernière étant juste à la limite d'être acceptée.

4.3.3 Limites et fiabilité relative des résultats

Cependant, il faut rappeler que les résultats issus de VSPAERO doivent être interprétés avec prudence. Les études consultées et mes propres observations montrent que ce simulateur, s'il reste efficace pour obtenir rapidement des ordres de grandeur, présente une certaine imprécision et une variabilité notable. Ces instabilités se retrouvent dans les courbes de tendance comme dans les valeurs brutes, ce qui limite la confiance que l'on peut accorder au classement final. La construction même des courbes de tendance résulte d'un compromis méthodologique entre filtrage des valeurs aberrantes et recherche de lisibilité, ce qui ajoute un biais supplémentaire (voir Section 4.1).

Un autre élément à garder en tête est que mes simulations ont toutes été réalisées en statique – un avion immobile soumis à un flux constant. Les effets dynamiques comme les turbulences, les instabilités transitoires issus de l'environnement réel ou encore l'effet de sol n'apparaissent donc pas. Ce dernier, appelé « effet de sol », correspond à la modification des caractéristiques aérodynamiques de l'appareil lorsque celui-ci évolue à faible hauteur par rapport à la surface : la présence du sol limite le développement du vortex sous l'aile, entraîne une augmentation de la portance et une réduction de la traînée, ce qui fausse les résultats par rapport à une configuration en vol libre. Ces manques accentuent la distance existante entre les résultats simulés et le comportement réel en vol. On évalue donc ici des tendances valables en croisière stationnaire, sans prendre en compte les phases critiques telles que décollage ou turbulences atmosphériques.

Enfin, le modèle (`angle_10_TC_1`) a dû être écarté car VSPAERO n'a renvoyé que des valeurs NaN (*Not a Number*). Cela ne modifie cependant pas les résultats, car ce modèle n'est pas représentatif des meilleures configurations selon nos critères au vu du placement des modèles affiliés.

4.4 Choix final et portée du classement

En tenant compte de l'ensemble de ces limites, le classement établi reste exploitable mais doit être interprété comme un outil d'aide à la décision plutôt qu'une vérité absolue. On observe par exemple que le modèle réel de référence (épaisseur de 15%, déviation des ailes externes à 7°) n'obtient qu'un rang moyen de 9. Pourtant, les deux meilleurs modèles mis en avant, avec des angles de 10° et 5° pour la même épaisseur, montrent des performances très proches. Cela suggère que la relation entre l'angle des ailes et la direction globale du flux n'est ni triviale, ni linéaire, ni quadratique, mais résulte d'effets complexes où s'entrecroisent portance, décrochage et moments de stabilité.

Ce travail apporte une base de comparaison pertinente entre modèles, permettant déjà d'exclure directement certaines configurations instables (notamment celles avec un C_M trop pentu). Mais choisir un modèle final nécessite d'accepter une part d'arbitraire, faute de validation expérimentale ou de simulation dynamique plus poussée.

Pour finir, j'ai décidé de choisir entre les 3 premiers du classement comparatif, car ce sont ceux qui présentent les rangs moyens les plus bas, très proches (différence d'environ 0,6), et leurs courbes sont tout à fait acceptables d'un point de vue physique, ce qui garantit une cohérence globale des résultats.

Parmi ces trois modèles, j'ai retenu le premier car il combine la meilleure performance en termes de rapport L/D , un bon coefficient de moment (C_M), et une répartition quasi exclusivement concentrée sur l'axe C_{M_y} , indiquant une stabilité aérodynamique équilibrée et favorable. Le deuxième modèle affiche un rapport L/D équivalent mais possède le pire coefficient C_M du groupe, ce qui nuit à sa stabilité globale malgré une répartition similaire des moments. Le troisième modèle, bien que présentant le meilleur coefficient C_M , possède un rapport L/D légèrement inférieur et une répartition des moments moins favorable avec une forte contribution sur les axes C_{M_x} et C_{M_z} , ce qui peut engendrer des comportements instables ou plus difficiles à maîtriser en vol.

Ainsi, le modèle `angle_10_TC_15` représente le meilleur compromis entre efficacité énergétique et stabilité aérodynamique tout en tenant compte de la proximité des performances des autres modèles du top 3. Ces résultats sont synthétisés dans le tableau 4.1 et illustrés par les courbes des coefficients aérodynamiques reportées dans la planche 4.5.

5 Construction et impression 3D du modèle réduit

Au moment de rendre ce rapport, l'avion n'est pas encore entièrement assemblé et n'aura donc pas effectué de vol. Voici les principaux enjeux :

5.1 Préparation du modèle pour l'impression

Après avoir sélectionné le meilleur modèle d'aile selon les critères définis dans les chapitres précédents, il a fallu adapter le modèle 3D pour l'impression. Cette étape n'est pas triviale : un modèle conçu pour la simulation n'est pas prêt à être imprimé. J'ai dû vérifier chaque composant, ajuster les épaisseurs, et m'assurer que toutes les parties étaient bien fermées pour pouvoir importer le modèle dans Fusion 360.

Pour garantir la solidité du modèle, j'ai augmenté localement certaines épaisseurs, notamment au niveau du fuselage et des points de fixation des ailes. J'ai aussi ajouté des renforts internes, invisibles sur le modèle extérieur, mais essentiels pour éviter que l'avion ne se casse au moindre choc. L'espace pour l'électronique (récepteur, servos, batterie) a été prévu dès la modélisation, avec des logements adaptés et des passages pour les câbles. Il a fallu également modéliser les pièces mécaniques permettant le contrôle de l'avion en l'air.

Il a fallu également modéliser les pièces mécaniques permettant le contrôle de l'avion en vol. Cela inclut principalement les ailerons sur les ailes, qui permettent de faire pencher l'avion autour de son axe longitudinal et ainsi de tourner ; la profondeur, située sur la partie arrière, qui contrôle l'assiette de l'avion et permet de monter ou descendre ; et la dérive, qui agit sur le lacet et permet de corriger la trajectoire autour de l'axe vertical. Chaque gouverne est reliée à un servo via une biellette ou un lien mécanique, transformant la rotation du servo en déplacement angulaire de la surface de contrôle. Ces pièces doivent être robustes et parfaitement alignées pour que les mouvements soient précis et que l'avion réponde correctement aux commandes.

5.2 Découpage et organisation des pièces

L'impression 3D nécessite de découper le modèle en plusieurs parties, car l'espace d'impression est limité et les contraintes propres à la technique FDM empêchent l'ajout de matière lorsque les angles avec la couche inférieure sont trop importants. J'ai donc séparé le fuselage en 4 sections, les ailes en 3 parties chacune, et la dérive et les empennages en éléments indépendants. Ce découpage facilite non seulement l'impression, mais aussi le montage et le transport du modèle.

Une imprimante 3D FDM fonctionne en déposant du filament fondu couche par couche. Chaque pièce a été exportée individuellement au format STL, puis importée dans le slicer (Cura) pour paramétriser l'impression. Cura est un logiciel qui transforme le modèle 3D en instructions pour l'imprimante, en définissant la trajectoire de la buse, l'épaisseur des couches, la vitesse d'impression et les supports éventuels nécessaires à la réalisation des pièces. J'ai choisi de rajouter de la complexité dans ma modélisation pour pouvoir imprimer certaines pièces en mode vase (le filament est déposé en un seul parcours continu, sans interruption de la buse). Le problème avec mon filament technique LW-PLA-HT (*LightWeight PLA High Temperature*) c'est qu'il ne permet pas la rétraction, donc impossible de faire des allers-retours de buse sans "stringing", d'où la nécessité de limiter les supports au maximum. Le stringing se produit lorsque le filament continue de couler légèrement de la buse pendant les déplacements à vide, créant ainsi de fins fils de plastique entre les parties de la pièce.

5.3 Paramètres d'impression et choix du matériau

Le choix du matériau est crucial pour un modèle réduit volant. J'ai utilisé du LW-PLA-HT pour toutes les pièces. Ce matériau a l'avantage d'être très léger grâce à sa capacité d'expansion à plus haute température que le PLA normal, tout en gardant une bonne résistance à la chaleur (jusqu'à 80°C en plein soleil) seul défaut : son prix, assez conséquent pour une bobine de 750g. Le *Polylactic Acid* (PLA) est un matériau plastique composé d'acide polylactique, dérivé de sources végétales comme l'amidon de maïs ou la canne à sucre. Son nom peut prêter à confusion, car il ne s'agit pas à proprement parler d'un acide, mais d'un polyester. Le PLA est apprécié en impression 3D pour sa facilité d'utilisation, sa faible déformation à la chauffe, sa rigidité et sa biodégradabilité, ce qui en fait un matériau à la fois pratique et respectueux de l'environnement pour la fabrication de prototypes et de pièces légères. Les paramètres d'impression ont été optimisés pour chaque pièce : couche de 0.2 mm, vitesse modérée (50 mm/s), température adaptée au matériau (235°C). J'ai aussi ajusté la ventilation pour éviter les déformations, surtout sur les grandes pièces comme les ailes.



FIGURE 5.1 – Voici un prototype du modèle réduit imprimé en 3D : il ne fera pas partie du modèle final, car il a été réalisé avec trop de matière, mais il m'a permis de tester le fonctionnement des différentes gouvernes mécaniques.

5.4 Assemblage et ajustements mécaniques

Une fois toutes les pièces imprimées, l'assemblage a commencé. J'ai utilisé de la colle forte pour toutes les parties, avec des guides en cure-dent pour aligner parfaitement les pièces, et des vis pour pouvoir démonter les ailes principales. Quelques ajustements à la lime ou au cutter ont été nécessaires, comme toujours avec l'impression 3D.

5.5 Système de guidage radiocommandé

J'ai opté pour le système le plus simple à installer : une radio numérique avec son récepteur, une batterie LiPo, un moteur brushless avec son ESC (contrôleur du moteur) et des servos moteurs pour les gouvernes. Tout a été prévu pour que le montage soit rapide et que le centre de gravité reste correct.

5.6 Difficultés rencontrées et solutions

Comme souvent en impression 3D, tout ne s'est pas passé comme prévu. Certaines pièces ont eu des défauts d'impression (warping, sous-extrusion), que j'ai dû réimprimer ou réparer.

Le poids final du modèle a légèrement dépassé l'estimation initiale, principalement à cause des renforts et des ajustements mécaniques. Mais je n'ai toujours aucune expérience de voile me permettant d'en décrire les conséquences.

6 Conclusion

6.1 Bilan global du travail

Au terme de ce travail, je peux affirmer que les objectifs initiaux ont été atteints, voire presque totalement réalisés. Premièrement, j'ai compris le fonctionnement physique de la portance et de la traînée, et j'ai pu exprimer mathématiquement comment les obtenir. En résumé, tout découle de la distribution de pression autour du profil. En revanche, comprendre comment prédire précisément cette distribution restait hors de portée dans le temps imparti et compte tenu des connaissances mathématiques et physiques requises.

Grâce aux simulations, j'ai constaté que l'augmentation de l'angle des ailes externes n'avait pas de corrélation simple et visible avec le rapport portance/traînée, de même que pour l'épaisseur du profil. De plus, le nombre de Reynolds diffère entre le modèle réduit et le modèle réel, ce qui rend leurs comportements incomparables.

Je n'ai pas encore terminé la production du modèle réduit, mais il est déjà entièrement modélisé. J'ai également créé un dépôt GitHub nommé *pc7-mk2-simtoolkit*, contenant tous les scripts Python développés ainsi que les fichiers nécessaires pour les simulations et les modélisations. Il sera mis à jour dès que j'aurai généré l'ensemble des fichiers STL destinés à l'impression 3D.

6.2 Apports personnels et perspectives pour la suite

Je suis particulièrement fier d'avoir su relier entre elles de nombreuses connaissances issues de domaines qui, à première vue, ne semblent pas directement liés : la physique des fluides, la programmation, la modélisation 3D, l'automatisation de simulations et la fabrication additive. Cette capacité à combiner ces compétences pour aboutir à un projet cohérent et concret constitue un apport personnel majeur de ce travail. Voir ces connaissances déboucher sur un résultat tangible — un objet réel dans lequel j'ai investi beaucoup de temps et d'énergie — est également une source de grande satisfaction.

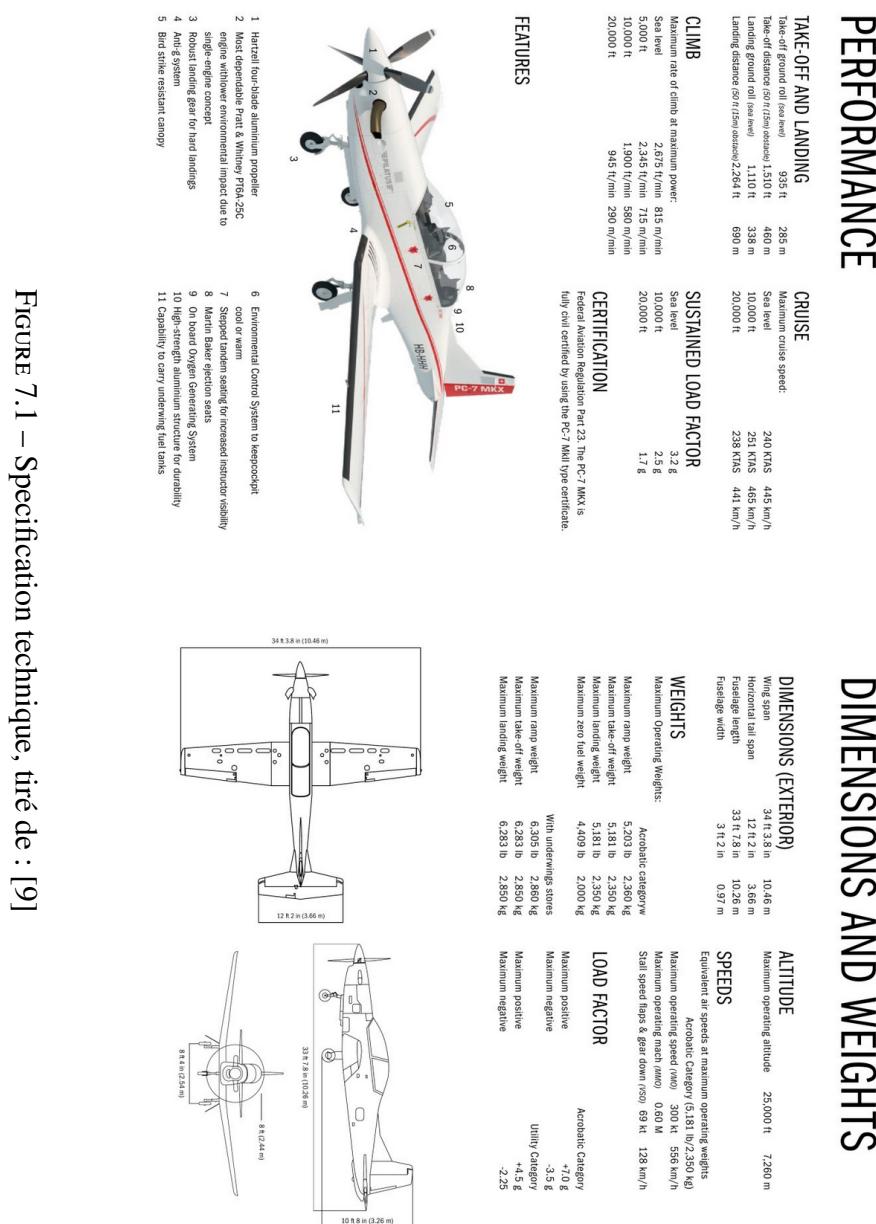
Pour la suite, ce travail ouvre plusieurs perspectives, notamment l'amélioration des simulations grâce à des moyens de calcul plus puissants, ainsi que l'expérimentation en vol réel du modèle afin d'évaluer ses caractéristiques aérodynamiques. Enfin, j'ai partagé le lien de mon dépôt GitHub sur différents forums, afin d'aider d'autres personnes confrontées aux mêmes

6 Conclusion

difficultés que moi dans l'automatisation des simulations.

7 Annexe

7.1 Plans et fiche technique



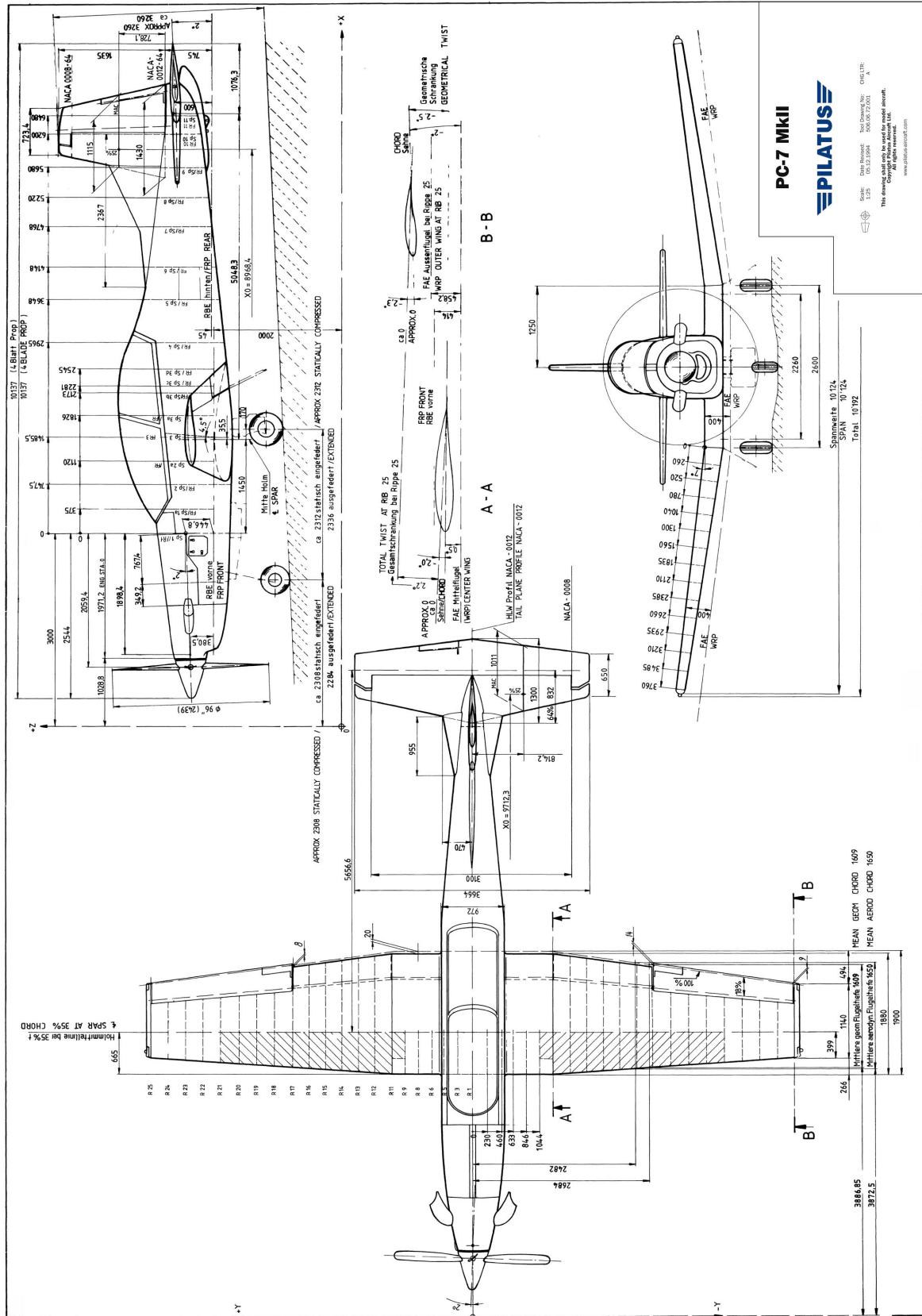


FIGURE 7.2 – Plan d'ensemble du PC7 MkII, tiré de : [21]

7.2 Script formatage coordonnees profil

```
1 import csv
2 vals = []
3 with open('Results.csv', newline='') as csvfile:
4     reader = csv.DictReader(csvfile)
5     for row in reader:
6         vals.append((row["X"], row["Y"]))
7 origin = vals[15]
8 chord = float(vals[0][0]) - float(origin[0])
9 for points in vals:
10    x = (float(points[0]) - float(origin[0])) * 1/chord
11    y = (float(points[1]) - float(origin[1])) * -1/chord
12    print(f"{x} {y}")
```

Script 7.1 – Script de formatage des coordonnees du profil; ImageJ me donnais un fichier csv contenant les coordonnees

7.3 Script automatisation des simulations

```
1 import openvsp as vsp
2 import os
3 import math
4 import shutil
5 import subprocess
6 import time
7
8 origin_vsp3 = "test.vsp3"
9 output_dir = r"C:\TM_Simulations\results"
10 delta_yc_values = [0, 5, 7, 10, 15, 20]
11 Span_ext = 0.3762
12 Chord = 0.188
13 thickoverchord = [0.10, 0.15, 0.20]
14 analysis_file = []
15
16 os.makedirs(output_dir, exist_ok=True)
17 initial_dir = os.getcwd()
18
19 def main():
20     printandwrite("[INFO]Lancement du script...")
21     printandwrite("[INFO]Creation des fichiers...")
22     try:
23         create_analysis_files()
24     except Exception as e:
25         printandwrite("[ERROR]Nous n'avons pas pu creer les fichiers d'analyse.")
```

```

26     printandwrite(f"[ERROR]Detail : {e}")
27     return
28 printandwrite(f"[INFO]{len(analysis_file)} fichiers ont ete crees.")
29 printandwrite("[INFO]Copie des fichiers de paramétrages des simulations
30    réussis.")
31 printandwrite(" [INFO]Génération des géométries...")
32 compute_geometry(analysis_file)
33 os.chdir(r"C:\TM_Simulations")
34 printandwrite("[INFO]Géométrie calculée.")
35 printandwrite("[INFO]Lancement des simulations")
36 try:
37     analyse(analysis_file)
38 except Exception as e:
39     printandwrite(f"[ERROR]Il y a eu un problème pendant l'analyse : {e}
40 })
41
42
43 def create_analysis_files():
44     for angle in delta_y_c_values:
45         for TC in thickoverchord:
46             combo_name = f"angle_{angle}_TC_{TC}"
47             combo_name = combo_name.replace("0.", "")
48             case_dir = os.path.join(output_dir, combo_name)
49             os.makedirs(case_dir, exist_ok=True)
50             copy_vspaero_settings(case_dir, f"{combo_name}_VSPGeom")
51
52             vsp.ClearVSPModel()
53             vsp.ReadVSPFile(origin_vsp3)
54
55             wing_id = vsp.FindGeom("Wing", 0)
56             #fuse_id = vsp.FindGeom("Fuselage", 0) A ne pas faire car
57             VSPAERO calcul selon le plan référence XZ et si on tourne l'avion la
58             physique se casse
59             #vsp.SetParmVal(fuse_id, "X_Rel_Rotation", "XForm", theta)
60             dyc = math.tan(math.radians(angle)) * Span_ext / Chord
61             vsp.SetParmVal(wing_id, "DeltaY", "XSecCurve_2", dyc)
62
63             for section_index in [0, 1, 2]:
64                 vsp.SetParmVal(wing_id, "ThickChord", f"XSecCurve_{section_index}", TC)
65
66                 vsp.Update()
67                 new_vsp3 = os.path.join(case_dir, f"{combo_name}.vsp3")
68                 analysis_file.append((case_dir, f"{combo_name}_VSPGeom"))
69                 vsp.WriteVSPFile(new_vsp3, vsp.SET_ALL)

```

```
68         printandwrite("[INFO]1 dossier cree")
69
70 def copy_vspaero_settings(dir, file):
71     src = "origin.vspaero"
72     dir_dst = os.path.join(dir, file)
73     dir_dst += ".vspaero"
74     shutil.copyfile(src, dir_dst)
75
76 def compute_geometry(list_dir):
77     for dir, file in list_dir:
78         src = "generate_geom.vpscript"
79         dir_dst = os.path.join(dir, src)
80         shutil.copyfile(src, dir_dst)
81         file_vsp3 = file.replace("_VSPGeom", ".vsp3")
82         vpscript_content = f'''// generate_geom.vpscript
83 void main()
84 {{
85     ReadVSPFile("{file_vsp3}");
86
87     // Run CompGeom to generate geometry
88     string compGeom = "VSPAEROComputeGeometry";
89     SetAnalysisInputDefaults(compGeom);
90     string compGeom_results = ExecAnalysis(compGeom);
91 }}
```

92 ''',

```
93         with open(dir_dst, "w") as f:
94             f.write(vpscript_content)
95             printandwrite(f"[INFO] Lancement de Compute Geometry pour {file}")
96             cmd = rf"C:\OpenVSP-3.43.0-win64-Python3.9\OpenVSP-3.43.0-win64\vsp
97 .exe -script {dir_dst}"
98             try:
99                 os.chdir(dir)
100                subprocess.run(cmd, check=True)
101            except:
102                printandwrite("Il dit que c'est une erreur mais non let's go...")
103        printandwrite(f"[SUCCESS] Termine pour {file}")

104 def analyse(list_dir):
105     total = len(list_dir)
106     avg_time = 0
107     for idx, (dir, file) in enumerate(list_dir):
108         start_time = time.time()
109         printandwrite(f"[INFO] Lancement de VSPAERO pour {file}, file {idx
110 +1} / {total}")
111         cmd = rf"C:\OpenVSP-3.43.0-win64-Python3.9\OpenVSP-3.43.0-win64\
112 vspaero.exe -omp 10 -dokt {os.path.join(dir, file)}"
```

```

111     subprocess.run(cmd, check=True, cwd=dir)
112     elapsed = time.time() - start_time
113     avg_time = elapsed if idx == 0 else (avg_time * idx + elapsed) / (
114         idx + 1)
115     remaining = (total - idx - 1) * avg_time
116
117     printandwrite(f"[SUCCESS] Analyse terminee pour {file} en {elapsed
118 :.1f} s")
119     printandwrite(f"[INFO] Temps estime restant : environ {int(
120 remaining // 60)} min {int(remaining % 60)} s\n")
121     printandwrite(f"[SUCCESS] Analyse termine pour {file}, file {idx+1}
122 / {len(list_dir)})")
123
124 def printandwrite(msg):
125     print(msg)
126     with open("output.history", "a", encoding="utf-8") as f:
127         f.write(f"{msg}\n")
128
129 main()

```

Script 7.2 – Script d’automatisation de tout le processus de simulation aerodynamiques

7.4 Extraction des données

```

1 """
2 Structure :
3 cases = {
4     "alpha-beta": {
5         "cl": [it1, it2, it3, ...],
6         "cd": [it1, it2, it3, ...],
7         "1/d": [it1, it2, it3, ...],
8         "cmx": [it1, it2, it3, ...],
9         "cmy": [it1, it2, it3, ...],
10        "cmz": [it1, it2, it3, ...]
11    }
12 }
13 """
14
15
16 import os
17 import json
18
19 def extract_iterations(file_name):
20     with open(file_name) as f:
21         lines = f.readlines()
22     cases = {}
23     for i, l in enumerate(lines):

```

```
24     if l.startswith("AoA_"):
25         a = l.split()[-2]
26         b = lines[i+1].split()[-2]
27         name = f"{a}/{b}"
28         j = i
29         while j < len(lines) and not lines[j].strip().startswith("Iter")
30     ):
31         j += 1
32         if j >= len(lines):
33             continue
34         h = lines[j].split()
35         idx = {}
36         for n, k in enumerate(h):
37             idx[k] = n
38         cases[name] = {"cl": [], "cd": [], "l/d": [], "cmx": [], "cmy":
39             [], "cmz": []}
40         for d in lines[j+1:]:
41             p = d.split()
42             try:
43                 cases[name]["cl"].append(float(p[idx["CLtot"]]))
44                 cases[name]["cd"].append(float(p[idx["CDtot"]]))
45                 cases[name]["l/d"].append(float(p[idx["L/D"]]))
46                 cases[name]["cmx"].append(float(p[idx["CMxtot"]]))
47                 cases[name]["cmy"].append(float(p[idx["CMytot"]]))
48                 cases[name]["cmz"].append(float(p[idx["CMztot"]]))
49             except:
50                 break
51     return cases
52
53 def load_all_files(folders_path):
54     dataset = {}
55     for folder in os.listdir(folders_path):
56         for filename in os.listdir(os.path.join(folders_path, folder)):
57             if filename.endswith(".history"):
58                 filepath = os.path.join(os.path.join(folders_path, folder),
59                                         filename)
60                 key = os.path.splitext(filename)[0]
61                 key = key.replace("_VSPGeom", "")
62                 dataset[key] = extract_iterations(filepath)
63     return dataset
64
65 def save_in_json_file(dataset, output_file):
66     with open(output_file, 'w') as f:
67         json.dump(dataset, f, indent=2)
68
69 def main():
70     dataset = load_all_files(r"C:\Users\arthur\OneDrive - Education Vaud\TM
```

```
68     \TM_Simulations\results")
69     print(f"{len(dataset)} fichiers traités")
70     save_in_json_file(dataset, "database.json")
71 if __name__ == "__main__":
72     main()
```

Script 7.3 – Script d'extraction des résultats et sauvegarde dans un fichier json

7.5 Notebook d'analyse des résultats

Tous les scripts suivant se trouvent dans le jupyter notebook intitulé GetGraphs sur le dépôt GitHub.

Script 7.4 – Planche Beta 1

Script 7.5 – Correction des erreurs

Script 7.6 – Exemple de Divité

Script 7.7 – Planche Beta 2

Script 7.8 – Classement des composants

Script 7.9 – Planche Beta 3

Script 7.10 – Planche Beta Coeff 1

Remerciements

Je voudrais remercier mes amis pour les innombrables discussions et réflexions qui m'ont permis d'aller plus loin, parfois même de trouver des solutions, mais aussi pour le temps qu'ils ont consacré à améliorer la clarté et la compréhension de mon texte.

Je tiens également à remercier mon grand-père, Jacques Steimer, qui a pris le temps de me relire et de me donner des commentaires constructifs. J'aimerais citer ma mère pour l'idée du PC7, mon camarade de classe Nathan Combes pour son aide précieuse liée à l'impression 3D, ainsi que ma Nani, qui a su lire à travers mon français très personnel et corriger ce langage.

Je souhaite enfin exprimer toute ma gratitude envers mon professeur de physique, Monsieur Philippe Kobel, qui a su transmettre sa passion et son epicurisme pour la physique durant ces trois années de gymnase. Pour finir, je remercie ma famille de m'avoir soutenu, y compris dans les moments d'incertitude, inévitables dans toute réflexion et dans tout travail demandant un fort investissement personnel.

Bibliographie

- [1] Airfoiltools, <http://airfoiltools.com/>, consulté le 13.07.2025
- [2] Perplexity ai, <https://www.perplexity.ai/>, consulté le 20.09.2025
- [3] , Documentation officiel openvsppytools, https://openvsp.org/api_docs/latest/, consulté le 14.05.2025
- [4] , Site de présentation de fusion 360, <https://www.autodesk.com/ch-fr/products/fusion-360/overview>, consulté le 14.05.2025
- [5] , Google group de openvsp, <https://groups.google.com/g/openvsp>, consulté le 14.05.2025
- [6] , Page officiel de *ImageJ*, <https://imagej.net/ij/>, consulté le 14.08.2025
- [7] , Site officiel de openvsp, <https://openvsp.org/>, consulté le 14.05.2025
- [8] , Site de formation de openvsp, <https://vspu.larc.nasa.gov/>, consulté le 14.05.2025
- [9] , Page officiel du *Pilatus PC7-MkII*, <https://www.pilatus-aircraft.com/en/pc-7>, consulté le 14.08.2025
- [10] , Article wikipedia *Aerodynamics*, <https://en.wikipedia.org/wiki/Aerodynamics>, consulté le 29.07.2025
- [11] , Article wikipedia *History of Aerodynamics*, https://en.wikipedia.org/wiki/History_of_aerodynamics, consulté le 06.08.2025
- [12] collectifs de la CRM, O., 2012, Formulaires et Tables, Commissions romandes de mathématique, de physique et de chimie
- [13] Gaëtan, Pilatus pc-7 turbo trainer, <https://www.avionslegendaires.net/avion-militaire/pilatus-pc-7-turbo-trainer>, consulté le 14.08.2025
- [14] Giovannini, A., Airiau, C., Brancher, P., 2016, Aérodynamique fondamentale, Éditions Cépaduès

BIBLIOGRAPHIE

- [15] Group, U. A. A., Uiuc airfoil data site, <https://m-selig.ae.illinois.edu/ads.html>, consulté le 13.07.2025
- [16] Jacqueline et François Rude, R. D., 1977, Le Monde Fascinant des Avions, Octopus Books Limited, Gründ (traduction)
- [17] JR., J. D. A., 2001, Fundamentals of Aerodynamics, third edition, McGraw-Hill Higher Education
- [18] Li, J., Huang, J., Sun, Y., Li, G., 2025, Accelerated convergence method for flow field based on dmd-pod combined reduced-order optimization model, IEEE Access, 13, 10 340–10 355
- [19] Loftin Jr, L. K., Cohen, K. S., 1948, Aerodynamic characteristics of a number of modified naca four-digit-series airfoil sections, Tech. rep., National Advisory Committee for Aeronautics (NACA)
- [20] Mariën, F., 2021, Software testing : VSPAERO, Master's thesis, Hamburg : Aircraft Design and Systems Group (AERO), Department of Automotive ...
- [21] Pilatus, Plan du pc7-mkii, <http://ipms.ch/userdata/filemanager/data/Profile/Pilatus-Aircraft-Ltd-PC-7-MkII-Model-Plan.pdf>, consulté le 13.04.2025
- [22] Series, N. F.-D., 2012, The naca airfoil series
- [23] Sheridan, C. N., Pham, D. D., Whiteside, S., 2021, Evaluation of vspaero analysis capabilities for conceptual design of aircraft with propeller-blown wings, in AIAA Aviation 2021 Forum, p. 2510