



## Rapport final, TMO-02

Projet de 1<sup>ière</sup> année - Transport et mobilité, 2015- 2016

### SIMULATION PARAMETRIQUE DE LA CONFIGURATION AERODYNAMIQUE DU BEE-PLANE

Amélioration de l'outil de simulation

Client : Xavier DUTERTRE

Référent pédagogique : Cédric Mehl

*Gaël COLAS, Hugo KERN, Jeremy KULCSAR, Calvin McSWEENY,  
Abdessamad MNAOUER*

# Sommaire

<b>1 Contexte du projet</b>	<b>2</b>
1.1 Histoire de l'aéronautique . . . . .	2
1.2 Contexte actuel . . . . .	3
1.3 Le BEE-PLANE™ . . . . .	4
1.4 Les aéronefs à fuselage détachable . . . . .	5
1.5 Le dimensionnement . . . . .	5
<b>2 Formulation de la problématique et des objectifs</b>	<b>7</b>
2.1 Elaboration de la problématique . . . . .	7
2.2 Objectifs du groupe TMO-02 . . . . .	8
2.2.1 Méthode du Ludion . . . . .	8
2.2.2 Tableau des objectifs du groupe TMO-02 . . . . .	9
<b>3 Le code</b>	<b>11</b>
3.1 Philosophie du code . . . . .	11
3.1.1 Une approche modulaire . . . . .	11
3.1.2 Le fonctionnement du code . . . . .	12
3.2 Etude comparative du logiciel en construction avec le logiciel <i>Aircraft design software</i> . . . . .	13
3.2.1 Description du logiciel . . . . .	13
3.2.2 Etude comparative . . . . .	15
<b>4 Livrables</b>	<b>16</b>
4.1 Cahier des charges . . . . .	16
4.1.1 Caractéristiques attendues . . . . .	16
4.1.2 Etat à la reprise des travaux . . . . .	16
4.2 1 <sup>er</sup> livrable : le guide du code . . . . .	17
4.2.1 Utilisation du guide . . . . .	17
4.2.2 Contenu du guide . . . . .	17
4.3 2 <sup>ème</sup> livrable : la refonte de la structure du code . . . . .	19
4.3.1 Etat des lieux de sortie du code . . . . .	19
<b>5 Etude expérimentale : écoulement de l'air autour d'une aile du Bee-Plane™</b>	<b>21</b>
<b>6 Nouvelles fonctions implémentées dans le code</b>	<b>24</b>
6.1 Fonctions rattachées à la Physique du code . . . . .	24
6.1.1 Conditions extérieures : la fonction <b>masse_vol_atm (T,alt)</b> . . . . .	24
6.1.2 Estimation de la consommation : la fonction <b>consommation()</b> . . . . .	25
6.1.3 Limitations par ces fonctions . . . . .	25
6.2 Fonctions rattachées à la base de données . . . . .	26
<b>7 Planning - Organisation de l'équipe</b>	<b>29</b>
7.1 Planning prévisionnel . . . . .	29
7.2 Organisation interne . . . . .	29
7.3 Interaction avec les membres du projet extérieurs au groupe . . . . .	30
7.3.1 Au niveau de l'école Centrale Paris . . . . .	30

7.3.2	Au niveau universitaire	31
<b>8</b>	<b>Les difficultés rencontrées par le groupe TMO-02 jusqu'à présent</b>	<b>32</b>
<b>9</b>	<b>Bibliographie</b>	<b>33</b>
<b>A</b>	<b>Présentation non-exhaustive des logiciels existants</b>	<b>34</b>
A.1	Le logiciel CATIA	34
A.2	Le logiciel MapleSim	34
A.3	Le logiciel AutoCad	35

# 1 Contexte du projet

## 1.1 Histoire de l'aéronautique

L'aviation a connu des débuts difficiles à la fin du XIX<sup>ème</sup> siècle. Malgré le scepticisme de la population, ce domaine a évolué et a atteint le stade de développement que nous connaissons aujourd'hui. Voici un historique des événements marquants dans l'histoire de l'aéronautique :

- 1873 premier ballon à air chaud par les frères Montgolfier
- Georges Cayley est le précurseur de l'aviation en découvrant les principes de l'aérodynamique et les forces de traînée et de portance
- 1902 les frères Wright expérimentent leur premier avion
- Premier vol commercial en 1914 aux Etats Unis ; la Poste commence elle aussi à s'intéresser à ce concept mais la guerre retarde les projets
- Pendant la première guerre mondiale on constate l'apparition de l'aviation militaire : les avions se spécialisent (reconnaissance, chasseurs et bombardiers...)
- Entre deux guerres : premières exploitations commerciales
  - 1918 : Naissance de la société des lignes *Latécoère*, future *Compagnie générale aéropostale*
  - 1919 : service de transport entre Paris et Londres et fondation de la compagnie aérienne néerlandaise *KLM*
- Seconde Guerre mondiale : inventions révolutionnaires (turbine, fusée, radar)
- 1950 : arrivée des premiers avions de ligne à turbopropulseurs puis à réaction. Ceux-ci permettront une diminution de 30% du coût du passager au kilomètre ainsi qu'une hausse significative de la vitesse de croisière, qui passe de 450 à 800 km.h<sup>-1</sup>
- 1950 : premier vol du Fairchild XC-120, prototype d'avion de transport militaire, développé par l'armée américaine dans les années 1950. Il a la caractéristique de disposer d'une soute cargo démontable située sous le fuselage. Cet avion, précurseur du Beeplane, restera à l'état de prototype. En effet ce concept, conçu pour réaliser des économies en kérosène et en temps de rotation au sol, était en décalage avec le contexte économique du secteur aéronautique de l'époque : le prix du baril de pétrole était faible relativement à son prix post-chocs pétroliers et les aéroports n'étaient pas congestionnés comme ils le sont aujourd'hui
- 1962 : le Concorde est développé, premier vol effectué en Mars 1969
  - Sa longueur de 62 mètres lui permet d'emporter 128 passagers sur une distance maximale de 6 400 km à une vitesse de M 2,05. Quatre réacteurs Olympus de 17 250 daN de poussée propulsent cet avion de 25 m d'envergure et d'une masse au décollage de 181 000 kg
  - Intérêt limité car pollution sonore, grosse consommation : préférences des compagnies pour les gros porteurs
- S'en suit depuis une quinzaine d'année, notamment avec l'essor d'une conscience de l'impact de l'activité humaine sur la planète, une évolution dans la façon de concevoir l'aviation :

- Des avions électriques : le transport aérien est pour le moment incompatible avec cette technologie car l'énergie stockée est trop faible. Cependant cette technologie peut être utilisée pour les drones. L'Union Européenne, *EADS* et de nombreux industriels et laboratoires se sont associés dans des projets <*VoltAir*>
- Des avions solaires : *Solar impulse* est un avion dont les ailes sont recouvertes de panneaux solaires ce qui permet de recharger les batteries en vol

## 1.2 Contexte actuel

Actuellement le transport aérien correspond à plus de 3 milliard de passagers dans le monde par an. Le trafic double en volume tous les dix ans. On comptabilise près de près de 30 millions de vols par an. L'augmentation constante du nombre de vols et de passagers ne pose pas de problèmes en vol mais les aéroports sont saturés.

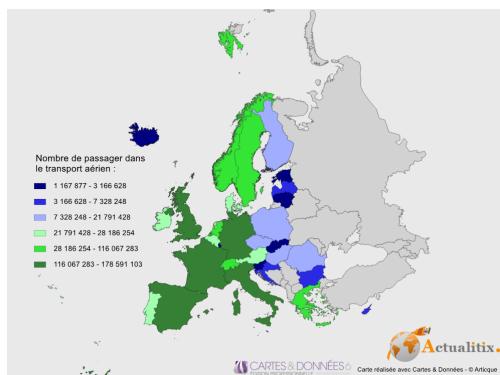


FIGURE 1.1 – Volume de passagers en Europe

Les exigences actuelles dans l'aviation civile sont nombreuses et tout avion qui est commercialisé doit les respecter. L'ensemble des contraintes au niveau du cahier des charges de tout nouveau projet d'avion et peut se résumer aux rubriques ci-jointes :

- Niveaux du bruit
- Empreinte carbone
- Rapidité du voyage
- Sécurité du transport

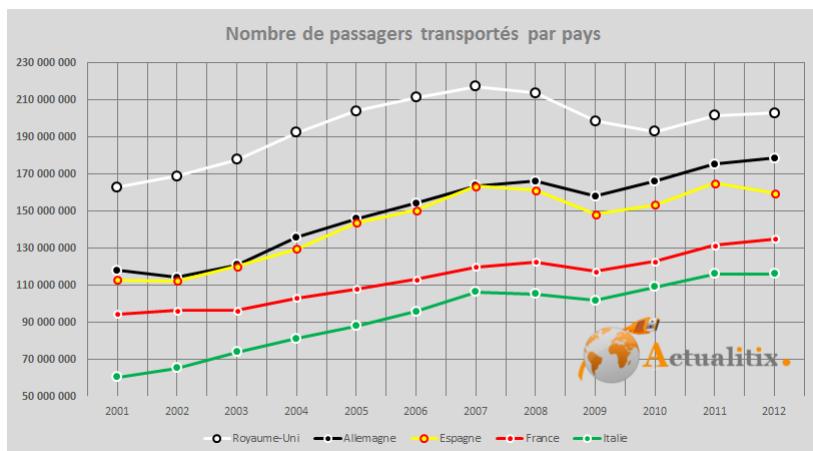


FIGURE 1.2 – Evolution du flux annuel de passagers dans le secteur aérien

Il faut donc repenser le transport aérien pour satisfaire aux exigences dans ce domaine.

### 1.3 Le Bee-Plane<sup>TM</sup>

Les objectifs d'un programme de vols sont de faire voler l'avion le plus possible et de le remplir au mieux, avec comme contraintes le respect des temps de rotation (turnover), des horaires des aéroports et des éventuels créneaux attribués au transporteur (slots). Le temps de roulage moyen pour des liaisons court/moyen-courrier représentant une part significative de la durée du trajet, tout comme le temps réduit à attendre le remplissage des réservoirs et le déchargement/chargement, qui engendrent une saturation des aéroports, c'est au niveau de cette phase au sol qu'il faudra chercher à gagner en efficacité.

Pour optimiser les temps de rotation, notre client a repris le concept du **Fairchild XC-120** (un aéronef à fuselage détachable), prototype d'avion de transport militaire composé d'un fuselage et d'une soute démontable, pour concevoir le BEE-PLANE<sup>TM</sup>. Cet avion est composé de deux parties majeures : le fuselage et la nacelle où se trouvent les passagers. L'avantage de cette conception est que le temps entre deux vols est considérablement réduit car les nacelles (appelées baskets) sont attachées aux fuselages dès l'atterrissement. De plus, comme le temps de roulage diminue, il y a une économie de kérosène et donc d'argent.



FIGURE 1.3 – Fairchild CX-120 en phase de chargement

Xavier Dutertre, fondateur du projet Beeplane, a établi un certain nombre d'objectifs en termes de caractéristiques à faire vérifier à l'avion :

- Une diminution de 30% du coût d'achat initial de l'avion mais aussi du coût de l'heure de vol pour la compagnie aérienne
- Une diminution de 50% de la consommation en carburant, des émissions de nuisances sonores et des émissions de  $CO_2$  en comparaison avec un moyen-courrier actuel de même capacité (du type *Airbus A321*)

En effet, le BEE-PLANE<sup>TM</sup> s'inspire de l'*Airbus A321* dans ses dimensions : il se veut être un moyen-courrier moderne, mais ne pourra transporter que 150 passagers par rotation. Cet avion est constitué d'un ensemble propulseur appelé le « Bee », comprenant l'ensemble de navigation, les éléments de propulsions et de gouverne et les ailes, ainsi qu'un « panier », fuselage contenant la charge utile. Ce qui rend cet avion novateur est la capacité de désolidariser le panier, contenant la charge utile (passagers ou cargo), du reste de l'avion. Le BEE-PLANE<sup>TM</sup> sera propulsé par deux turbopropulseurs TP400 et un turboréacteur CFM56-3C1. Les deux turbopropulseurs, moins gourmands en kérosène et donc avantageux pour la phase de vol en palier, étant insuffisants faire décoller le BEE-PLANE<sup>TM</sup> à eux seuls, un turboréacteur, est monté sur le fuselage sur le Bee pour aider au décollage et à la montée (avant d'être éteint pour le reste de la phase de vol).

	<i>Airbus A321</i>	BEE-PLANE <sup>TM</sup>
Envergure (m)	45	40
Longueur (m)	55	25
Capacité (nb de passagers)	220	150
Masse à vide (t)	70	70
Masse maximale au décollage (t)	140	100
Vitesse de croisière (km/h)	950	800
Ensemble de propulsion	2*275 kN (Turbofans PW400)	100 kN (2 turbopropulseurs TP400 et un turbofan CFM56)

TABLE 1.1 – Etude comparative de l'*Airbus A321* et du BEE-PLANE<sup>TM</sup> sur certaines caractéristiques techniques fondamentales

## 1.4 Les aéronefs à fuselage détachable

Le **Fairchild XC-120** est un prototype d'avion modulable dont s'est inspiré notre client M. Dutertre mais cet avion n'a jamais été commercialisé car trop en avance sur son temps. Cependant il n'y a que très peu de documentation sur cet avion. Ainsi il est très difficile de se servir de ce cas pour notre projet.

## 1.5 Le dimensionnement

Pour nous assurer de la cohérence et de la faisabilité des objectifs précédents, nous avons réalisé des études de dimensionnement, appelées « estimations de Fermi ».

Tout d'abord, une estimation du gain en temps apporté par le système de panier amovible. En réduisant le temps passé pour les phases « parasites » (autre que directement lié au vol), on diminue ainsi le coût de l'heure de vol : pour remplir les objectifs économiques du Bee-Plane.

Phase du vol	Bee-Plane <sup>TM</sup> (minutes)	Moyen-courrier classique (minutes)
Roulage	0	2*10=20
Chargement	10	15
Déchargement	20	15
Nettoyage cabine	0	10
<b>Total pour 1 rotation</b>	<b>30</b>	<b>60</b>

TABLE 1.2 – Tableau comparatif des temps des différentes étapes

Ce qui entraîne un gain de 30 minutes par rotation.

Puis, une estimation du gain en carburant. Le système de panier amovible permet effectivement d'économiser le carburant consommé lors des phases de roulage et d'attente : pour remplir les objectifs écologiques du Bee-Plane.

Appareil	Bus	A321
<b>Consommation</b>	26L/100km	860L/h
<b>Temps passé sur le taxiway</b>		15min
<b>Distance parcourue sur le taxiway</b>	2*500=1km	
<b>Consommation par rotation</b>	2,6L	215L

TABLE 1.3 – Tableau comparatif entre différentes caractéristiques un bus et un A321

Ce qui entraîne un gain de 215L de kérosène par rotation.

Par exemple sur un trajet Paris-Nice, d'une distance de 707km, le BEE-PLANE<sup>TM</sup> mettra 1h20 et consommera 1150L pour parcourir les 707km reliant les 2 villes, ce qui représente un gain de 18% par rapport à un vol réalisé avec un A321 dans les mêmes conditions.

Enfin, nous avons décidé de questionner les objectifs du projet Bee-Plane en tant qu'avion moyen-courrier pour estimer leur faisabilité, et l'effort à réaliser pour les atteindre. En utilisant certaines caractéristiques du Bee-Plane, la formule de Breguet ci-dessous permet de déterminer le rayon d'action maximal de l'avion dans le modèle simplifié dans l'hypothèse d'un vol en palier et en régime stationnaire : nécessite le découpage du trajet en tronçons d'altitude constante.

$$R = \eta_0 \frac{L}{D} \frac{Q_r}{g} \ln\left(1 + \frac{M_{fuel}}{M_{finale}}\right)$$

$\eta_0$  : Rendement global du système de propulsion

$L, D$  : Portée et trainée de l'avion, en palier

$Q_r$  : Capacité calorifique du kérosène

$M_{fuel}, M_{finale}$  : Respectivement la masse de kérosène embarquée et la masse du BEE-PLANE<sup>TM</sup> à l'atterrissement

D'où l'on obtient que le BEE-PLANE<sup>TM</sup> présente un rayon d'action estimé à 3600km, soit significativement moins comparé à un avion moyen secteur actuellement sur le marché, comme l'illustre la figure 1.4.



FIGURE 1.4 – Comparaison du rayon d'action d'un avion moyen-courrier standard à celui du BEE-PLANE<sup>TM</sup>

## 2 Formulation de la problématique et des objectifs

Trois façons de réduire la consommation du BEE-PLANE<sup>TM</sup> en kérósène ont été identifiées :

- Au travers d'une innovation technologique de rupture, mais ceci entrerait en conflit avec l'idée de faire du BEE-PLANE<sup>TM</sup> un avion économique, de par l'empreinte qu'aurait la phase de recherche sur le prix de vente.
- Une diminution de la consommation lors des phases autres que le vol. Un brevet a été récemment déposé par Airbus à ce sujet.
- Une optimisation de l'aérodynamique de l'avion, d'où la conception et l'utilisation d'un code.

L'objectif qui nous a été donné par le client de notre projet peut se comprendre de la manière suivante :

- Il faut chercher à rendre le code plus modulable possible, ce qui peut passer par la création d'une base de données.
- Il faut de plus appliquer la dernière version du code au BEE-PLANE<sup>TM</sup> afin d'améliorer le comportement aérodynamique de l'avion, après l'avoir appliqué à d'autres avions existants pour comparer les sorties du code aux données existantes sur ces avions.
- Il faut enfin comprendre comment réorganiser le code, en partie en identifiant et en supprimant les redondances.

Au niveau de l'Ecole Centrale Paris, le projet enjeu BEE-PLANE<sup>TM</sup> se subdivise en deux groupes, TMO-01 et TMO-02, qui tout en travaillant ensemble sur le projet, se sont distribuées les tâches à accomplir pour atteindre les objectifs du client.

### 2.1 Elaboration de la problématique

#### Comment optimiser le comportement aérodynamique du BeePlane ?

Notre ligne directrice tout au long de ce projet est l'optimisation aérodynamique du BEE-PLANE<sup>TM</sup>, avion conceptuellement différent des avions de ligne actuels en ce que la cabine est contenue dans un fuselage disjoint, et donc détachable, du reste de l'avion. Plus précisément, ce qui nous est demandé est d'améliorer un logiciel préexistant, conçu et modifié par des groupes de projet enjeux antérieurs, de modélisation du comportement global d'un avion. Ce code doit permettre de tester le comportement en vol à la fois du BEE-PLANE<sup>TM</sup>, mais aussi celui de tout autre avion que l'on souhaiterait étudier. Ceci implique d'assurer une certaine polyvalence au code, puisqu'il devra prendre compte des éléments novateurs apportés au BEE-PLANE<sup>TM</sup>, tel que sa cabine amovible, tout en acceptant des configurations d'avions plus standards, et absentes sur le BEE-PLANE<sup>TM</sup>.

Enfin, il faudra aussi tenir compte des possibles améliorations futures qui pourront être apportées au BEE-PLANE<sup>TM</sup> comme le concept dérivé du MINI-BEE, Bee-plane de taille réduite et à

décollage vertical. Ceci aussi nous incite à produire un code ergonomique et le plus intuitif possible, afin de ne pas gêner l'évolution du BEE-PLANE™ au fil des nouvelles contraintes que l'on voudra lui imposer.

Nous avons décidé de synthétiser l'ensemble des contraintes à prendre en compte dans ce projet sous la forme du diagramme ci-dessus. Cette phase d'analyse des contraintes nous a semblé essentielle puisqu'elle nous guidera dans tous nos choix ultérieurs, en nous assurant que la conduite du projet reste fidèle aux exigences du client.

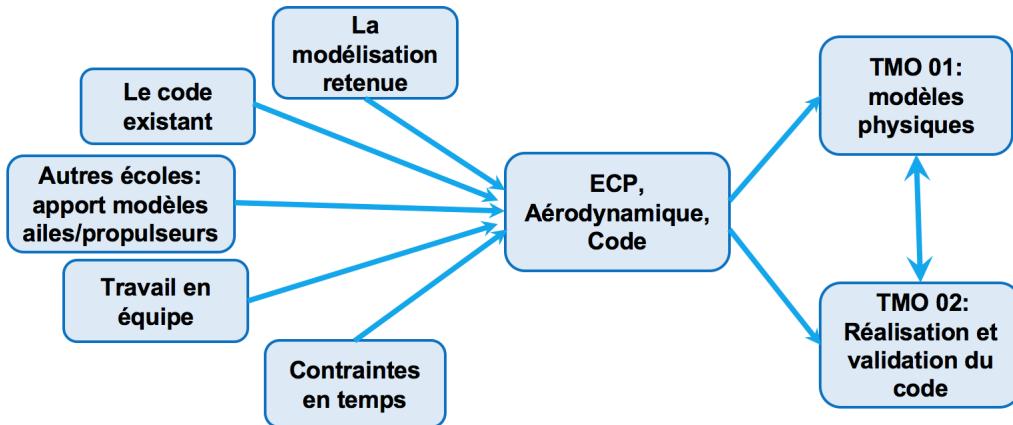


FIGURE 2.1 – Diagramme des contraintes du groupe TMO-02

Cette phase préalable nous a permis de réaliser la nécessité d'une utilisation optimale du travail existant, que ce soit sous la forme du code ou de protocoles expérimentaux, tout en assurant l'apport d'une valeur ajoutée au projet qui soit soutenue tout au long de l'année. Le travail réalisé par les équipes antérieures ne doit donc pas être vu comme un fardeau, malgré la difficulté à s'immerger dans un code et donc un mode de pensée probablement différent du nôtre, mais comme un atout sur lequel s'appuyer. En apportant un œil nouveau à un travail de plusieurs années, nous avons eu aussi plus de facilité à repérer ce qui serait à améliorer au code existant dans les phases ultérieures de notre projet.

## 2.2 Objectifs du groupe TMO-02

Au sein du projet BEE-PLANE™, le travail du groupe TMO-02 consiste à réaliser, ou du moins améliorer, et valider un code déterminant les paramètres aérodynamiques d'un avion.

### 2.2.1 Méthode du Ludion

La méthode du Ludion est un outil de travail permettant d'appréhender un projet le plus sereinement possible. Il vise à établir des solutions à la demande d'un client en établissant préalablement les raisons qui ont mené le client à demander notre soutien, les éléments au sujet du contexte dans lequel nous allons devoir travailler, et enfin les buts que nous souhaiterions nous fixer. Cette méthode a l'avantage de structurer notre démarche d'appropriation du sujet et nous a permis en ce qui concerne le projet BEE-PLANE™ d'appréhender des aspects que nous n'aurions pas identifié sinon. Ainsi, c'est en étudiant les attentes du client et en observant l'état du code qui nous a été transmis que nous avons compris qu'en plus du travail d'amélioration des capacités du code nous aurions à établir un support facilitant la prise en main du code.

En utilisant cette méthode, nous espérons nous épargner les problèmes d'appréhension de l'organisation et de la logique du code auxquelles ont eu à faire face les groupes antérieurs. Cette démarche s'inscrit donc dans une logique de remise en question de la ligne directrice du projet.

Elle ne vise pas à remettre en cause le travail déjà réalisé, mais à s'assurer que par les moyens mis en œuvres nous atteindrons les objectifs établis et respecterons les exigences du client.

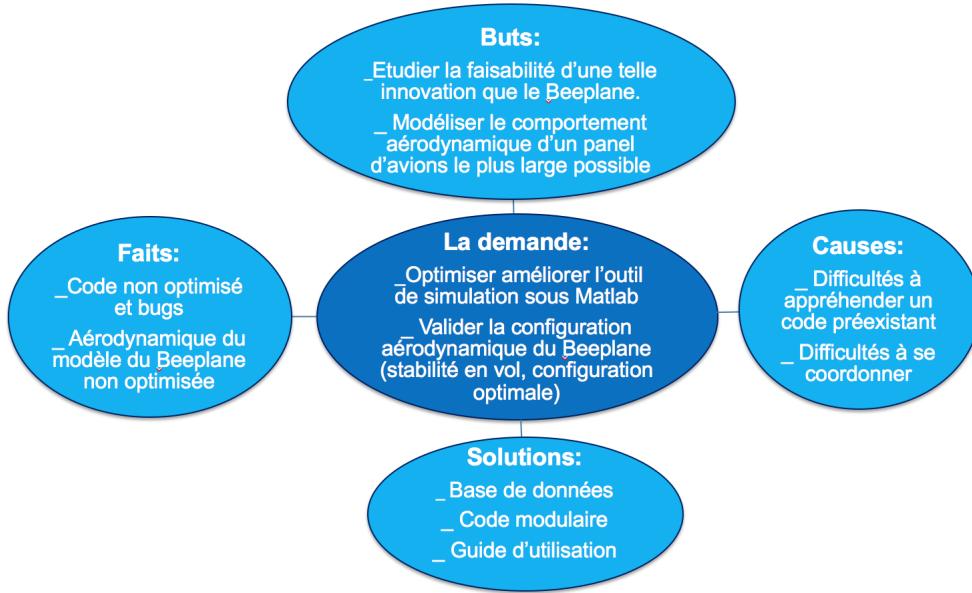


FIGURE 2.2 – Synthèse graphique de l'application de la méthode du Ludion au projet BEE-PLANE™

### 2.2.2 Tableau des objectifs du groupe TMO-02

C'est en suivant cette démarche d'élargissement et d'identification des points clés du projet que nous avons aboutis aux objectifs pour notre groupe. En particulier, la comparaison du travail déjà réalisé par les groupes antérieurs aux exigences du client nous a permis d'identifier les aspects du projet sur lesquels nous devrons le plus insister pour espérer apporter de la valeur ajoutée.

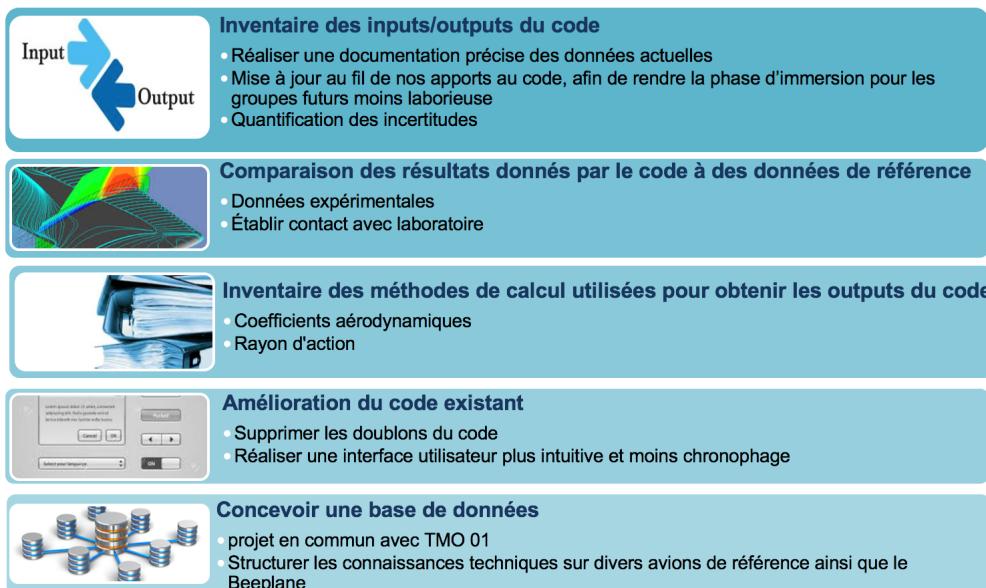


FIGURE 2.3 – Tableau récapitulatif des objectifs fixés pour l'année

Notre travail se scinde en six grandes tâches qui seront pour certaines réalisées en parallèle tout au long de l'année. L'objectif de cette structuration des objectifs est de s'assurer de ne pas être bloqué dans une phase de notre travail par un autre sous projet qui n'aurait pas été abouti à temps.

Une phase préliminaire d'inventaire des entrées et des sorties du code actuel nous a semblés nécessaire pour comprendre ce que le code pouvait déjà réaliser. Cette étape nous permettra aussi de débuter un autre aspect de notre projet qui est la rédaction d'un guide d'utilisation du code. En effet, comme il l'a déjà été mentionné, la facilité de compréhension et d'appréhension du code nous a semblé être un point clé pour avancer dans notre projet.

Ce guide, qui sera mis à jour à chaque modification majeure du code, comprendra entre autre pour chaque sous-programme, son historique, les entrées et sorties, et la logique qui a régi sa conception.

S'ajoute à cette 1<sup>ère</sup> étape une phase de comparaison des résultats renvoyés par le code à des données de référence établies expérimentalement ou issues de la théorie. Nous prendrons donc probablement contact avec un laboratoire spécialisé dans l'étude du comportement aérodynamique d'un avion pour nous assurer de notre démarche. Cette phase est fondamentale en ce qu'elle permettra d'identifier ce qui est valide dans le code existant et ce qui est à retravailler ou à reconcevoir. S'en suivra une phase d'amélioration du code à proprement parlé, que se soit en terme de suppression de redondances, de précision des résultats renvoyés, ou de diversité des paramètres pris en compte. Cette étape, réalisée en parallèle avec l'établissement de l'inventaire des méthodes de calculs utilisées pour obtenir les sorties du code, sera en partie consacrée à réaliser une interface utilisateur plus intuitive et moins chronophage.

Enfin nous avons prévu de réaliser, conjointement avec le groupe TMO-01, une base de données des caractéristiques physique des différents composants d'un avion. Cette base de données vise à simplifier la phase d'amélioration du code par comparaison avec des données expérimentales ainsi que la phase d'utilisation à proprement parler puisqu'à terme il n'y aura plus à entrer manuellement chaque paramètre de l'avion comme c'est actuellement encore le cas.

# 3 Le code

## 3.1 Philosophie du code

Le projet BEE-PLANE<sup>TM</sup> n'est pas un projet nouveau pour les différentes écoles qui travaillent avec Xavier Dutertre. Datant de quelques années, à ce projet était déjà associé une base de données - fruit du travail des étudiants des années antérieures - lorsque nous avons commencé à travailler dessus. C'est dans cette base que se trouve à notre disposition le code MATLAB réalisé par les anciens de l'ECP, code que nous analyserons dans cette partie du rapport.

### 3.1.1 Une approche modulaire

L'objectif du projet étant un avion modulable, c'est-à-dire constitué de plusieurs pièces indépendantes mais cohérentes qui sont rattachées entre elles (charge utile séparable du fuselage), c'est tout naturellement que le code adopte une forme similaire. Nos antécédents sont toutefois allés plus loin dans l'idée : ils ont en effet imaginé un code « bloc » pour chaque membre de l'avion (aile gauche, aile droite, etc...), que celui-ci représente une pièce entière ou non du Bee-Plane<sup>TM</sup>. L'avion était alors décomposé en 14 blocs principaux :

1. Train avant
2. Train arrière droit
3. Train arrière gauche
4. Aile droite
5. Aile gauche
6. Turbopropulseur droit
7. Turbopropulseur gauche
8. Turboréacteur
9. Carlingue Bee
10. Carlingue basket
11. Empennage horizontal droit
12. Empennage horizontal gauche
13. Empennage vertical droit (absent sur le Beeplane)
14. Accroche bee/basket

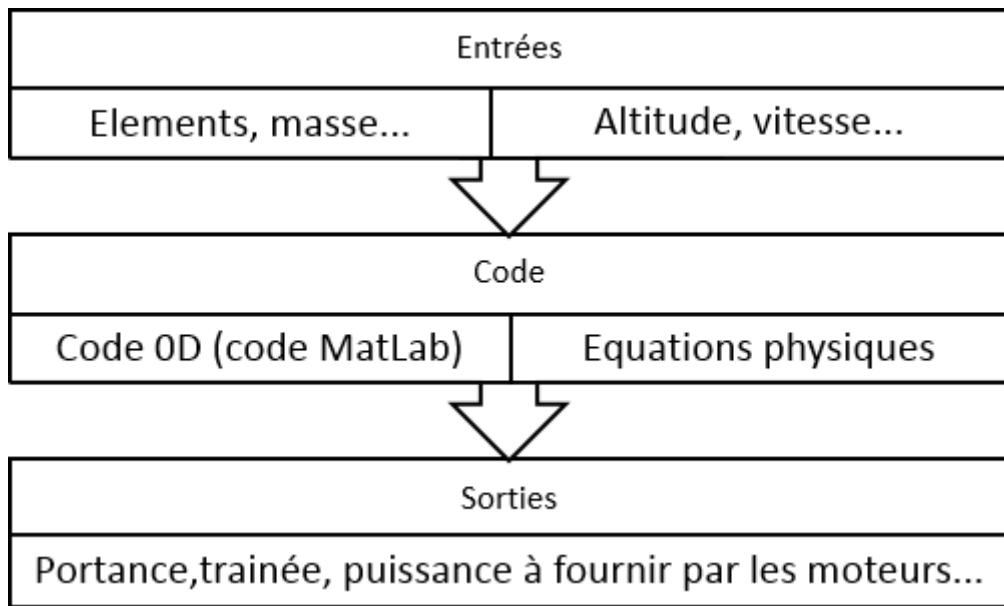
Si cette représentation de l'avion sous forme de blocs ne permet pas de tenir compte de manière fidèle les interactions entre les sous-systèmes de l'avion, elle permet toutefois une simplification des calculs engendrés.

Chaque membre de l'avion est alors caractérisé par différents éléments. Le protocole à suivre pour implémenter un nouvel élément dans le code est d'ailleurs d'y entrer successivement ces éléments :

- Nom
- Type
- Masse
- Incidence
- Longueur
- Profil NACA (uniquement pour les ailes) **MPTT** :
  - M : cambrure maximale
  - P : position du maximum de cambrure
  - TT : épaisseur maximale du profil
- Position du point de raccord
- Rotation de l'élément par rapport à un axe universel prédéfini

### 3.1.2 Le fonctionnement du code

Il s'agit d'un code de dimensionnement modulaire et adaptable, exécuté grâce à des logiciels de simulation afin de déterminer le profil avec toutes les caractéristiques du BEE-PLANE™. Ceci est une étape primordiale dans la conception d'un avion, quel qu'il soit. Les entrées du code sont des caractéristiques physiques de l'avion (éléments, masse, type d'aile), mais aussi des caractéristiques de vol (altitude, vitesse, décollage ou atterrissage). Grâce à ces données, on va pouvoir faire des cas tests à partir des résultats fournis par les logiciels de simulation (MATLAB ...).



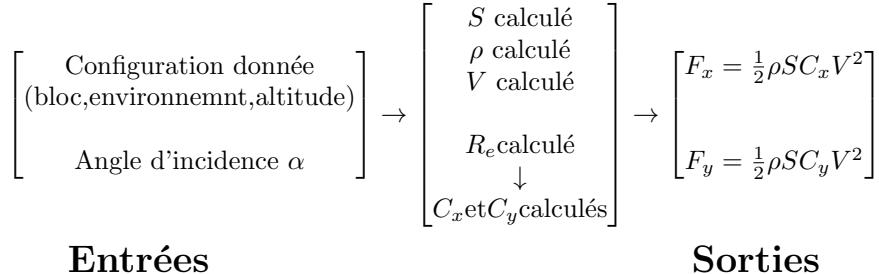
Paramétriser le profil aérodynamique d'un avion n'est pas chose aisée, et nécessite la connaissance de ce qui caractérise ce profil et le fluide qui l'entoure. C'est pourquoi le code a été conçu pour renvoyer une multitude de paramètres.

Premièrement, le code calcule les coefficients caractérisant les conditions extérieures. Il renvoie la pression atmosphérique, la température et la densité en fonction des données du problème

### 3.2 Etude comparative du logiciel en construction avec le logiciel Aircraft design software

(essentiellement l'altitude à laquelle se trouve l'avion). Cela est nécessaire, car le calcul des caractéristiques aérodynamiques de l'avion dépend de ces conditions extérieures.

Ensuite, il calcule et renvoie tous les coefficients aérodynamiques de l'avion nécessaires à la caractérisation du profil aérodynamique : coefficients de portance, de trainée (dont induite), nombre de Reynolds et surface mouillée. C'est cette partie des résultats qui est la plus importante pour répondre à notre problématique.



**Entrées**

**Sorties**

Toutefois, dans le souci de répondre aux questions annexes à la problématique, nous obtenons les caractéristiques mécaniques de l'avion : corde moyenne des ailes, surface alaire, masse des composants de la structure, foyer de chaque bloc (fuselage, ailes, ...) puis foyer global (après assemblage). Ces coefficients nous permettront de répondre aux questions que sont celles de la consommation de l'avion et du calcul du rayon moyen.

Par exemple, pour calculer la consommation par kilomètre parcouru, l'équipe de l'année dernière a choisi de passer par la consommation spécifique, cette caractéristique des moteurs permettant d'aboutir à la consommation qui nous intéresse.

A partir de cette consommation spécifique peut se définir la consommation horaire  $CH[kg/h]$  :

$$CH = F \times S$$

Où  $F[N]$  est la poussée nécessaire pour déplacer l'avion et  $CS[kg/(daN.h)]$  la consommation spécifique.

A ce moment, nous pouvons alors passer à la consommation kilométrique  $CK[kg/km]$  :

$$CK = CH \times \frac{D}{V}$$

Où  $D$  et  $V$  désignent respectivement la distance l'aéroport de départ et d'arrivée et la vitesse de vol. Ce calcul nous fournit donc la consommation par kilomètre parcouru de l'avion.

Autre exemple, l'équipe nous précédant a aussi choisi de trouver la masse du Bee-Plane par pondération des masses de l'A321 et de l'A320, deux avions aux caractéristiques proches du notre.

## 3.2 Etude comparative du logiciel en construction avec le logiciel *Aircraft design software*

### 3.2.1 Description du logiciel

Pour établir le code de ce projet, le logiciel principalement utilisé est MATLAB . Toutefois, il est important de noter que beaucoup d'autres logiciels sont utilisés dans l'industrie aéronautique.

### 3.2 Etude comparative du logiciel en construction avec le logiciel Aircraft design softwareLe code

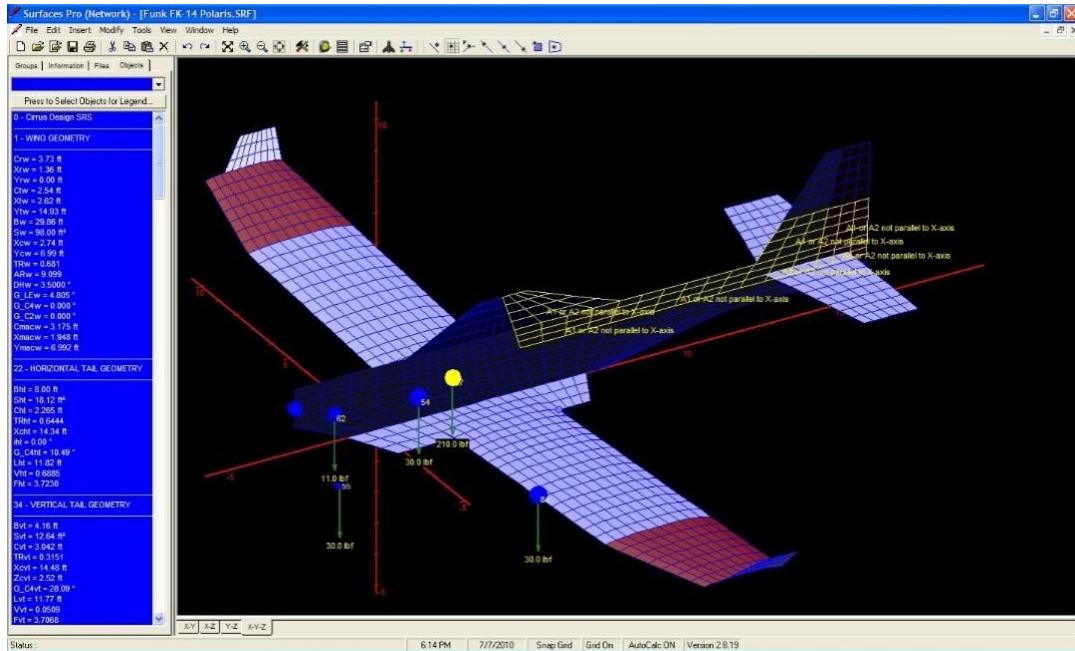


FIGURE 3.1 – Capture d'écran du logiciel *Aircraft design software*

Un des logiciels couramment utilisé pour le design d'aéronefs est le logiciel *Aircraft design software*. Il s'agit d'un logiciel professionnel très utilisé dans le monde industriel de l'aéronautique.

Ce logiciel est composé de différents modules (module d'analyse, d'analyse statistique, de modélisation, de conception 3D) qui, pris ensemble couvrent toutes les étapes de l'étude conceptuelle d'un projet. Les module de conception et d'optimisation sont ceux qui nous intéressent le plus pour cette étude puisque leurs fonctionnalités sont analogues à celles que nous souhaiterions proposer au travers du logiciel que nous concevons :

- Le **module de modélisation** permet de répondre à des questions du type : quelle doit être la surface de l'aile et des empennages pour assurer à la fois une sustentation suffisante et une bonne stabilité ? Quelle sera sa qualité aérodynamique, Quelles seront ses performances pour les différentes phases de vol ? Est-il possible de vérifier les exigences du cahier des charges ? L'étude est réalisée en trois phases, chacune permettant d'affiner la modélisation de l'avion en tenant compte de d'avantage de paramètres qu'à la phase précédente. Ainsi la 1<sup>ère</sup> phase se restreint à déterminer la géométrie d'ensemble de l'appareil et la puissance nécessaire pour atteindre un point de stabilisation, tandis que la dernière phase distingue les capacités de l'avion en fonction des différentes phases de vol.
  - Le **module optimisation** qui, en prenant en entrées la plage de valeurs acceptée par le concepteur pour chacune des caractéristiques de l'avion, renvoie la configuration optimale de l'avion parmi toutes les configurations possibles obtenues à l'issue de la phase de modélisation. A titre d'exemple la version actuelle (datant de 2015) de ce module permet de comparer simultanément 10000 configurations possibles pour un même appareil.
  - Les sorties du code sont réparties en six grands ensembles :
    - Géométrie comprenant la dimension des surfaces portantes
    - Devis de masse intégrant la masse à vide, masse maximale de vol par exemple
    - Propulsion
    - Aérodynamique contenant la traînée totale ainsi traînée spécifique pour chaque vitesse de vol
    - Performances tel que l'autonomie et la vitesse de croisière

### 3.2 Etude comparative du logiciel en construction avec le logiciel Aircraft design software

- Centrage avec des paramètres tel que les centres de gravité spécifiques

Ainsi, ce logiciel peut être utilisé pour comprendre le processus de conception d'un avion, tester des nouveaux concepts, explorer les phases de vol d'un appareil existant et analyser les effets de modifications apportées sur un appareil existant.

#### 3.2.2 Etude comparative

L'une des premières questions qui nous est venue à l'esprit dans ce projet a été la nécessité ou non de construire un nouveau logiciel. En effet, de nombreux logiciels présents sur le marché semblent déjà répondre aux exigences que nous souhaiterions faire vérifier à notre code. La réponse qui nous a été donné a été l'utilité pédagogique de concevoir soi-même le logiciel que l'on souhaite utiliser. Dans notre cas, il est vrai que connaître parfaitement les capacités et les limites du code est fondamental lorsque nous souhaitons l'utiliser pour optimiser l'aérodynamique du BEE-PLANE<sup>TM</sup>. En effet, toute modélisation étant in erronée, puisque ce n'est qu'une simplification d'une réalité vue sous un angle donné, ce n'est qu'en connaissant les limites de la structure du logiciel de modélisation utilisé que nous pouvons évaluer la précision de modélisation d'un avion tel que le BEE-PLANE<sup>TM</sup>.

Nous avons toutefois décidé de comparer les capacités de notre logiciel à celles du logiciel *Aircraft design software*. A première vue, cette étude nous a semblés quelque peu banale et inutile puisqu'elle ne nous apprenait rien de nouveau : le logiciel professionnel est bien plus performant et précis que le logiciel que nous concevons. Il prend un nombre d'entrée bien plus grand que le notre et renvoi des sorties, telles que l'évolution de la trainée spécifique en fonction de la vitesse de vol, que nous n'avons pas intégré à notre code.

Toutefois cette étude a été à terme très instructive puisqu'elle nous a permis de desceller deux erreurs fondamentales dans notre manière d'aborder le code :

- Au lieu de vouloir atteindre une précision de modélisation maximale dès le premier jet, nous devrions diviser l'étude en phases comme il est réalisé dans le logiciel *Aircraft design software*. L'apport de nouvelle entrée à chaque phase, au lieu d'une implémentation de l'intégralité des entrées dès le début du calcul, rendrait le code plus performant et plus lisible, et nous permettrait de d'identifier plus rapidement quelle portion du code ne fonctionne pas si tel était le cas.
- Si jusqu'à présent nous nous sommes borner à modéliser la phase de vol en croisière du BEE-PLANE<sup>TM</sup>, il faudra à terme prendre en compte les phases de montée et de descente, puis de décollage et d'atterrissage, pour réellement avoir une idée du comportement aérodynamique du BEE-PLANE<sup>TM</sup>. En effet les conditions physiques extérieures, ainsi que l'influence de chacune des parties de l'avion, varie grandement en fonction de la phase de vol.

Cette étude comparative nous a donc permis de mieux comprendre la posture à avoir vis à vis du code existant.

## 4 Livrables

### 4.1 Cahier des charges

#### 4.1.1 Caractéristiques attendues

Nous avons défini trois livrables différents pour répondre aux attentes de notre client. Bien que la réalisation d'un code soit le livrable principal, nous avons décidé de le compléter par deux autres livrables qui en faciliteront l'utilisation. Ainsi les livrables seront au nombre de trois :

- Le guide d'utilisation du code : chaque fonction y sera détaillée avec ses modifications, son rôle, son lien avec le reste du code et éventuellement ce qu'il faudra modifier pour l'améliorer.
- Le code en lui-même composé de fonctions de calcul des paramètres aérodynamiques et mécaniques, et des caractéristiques du vol (consommation...).
- Un dossier comparant les cas tests effectués avec les valeurs théoriques et indiquant la précision obtenue

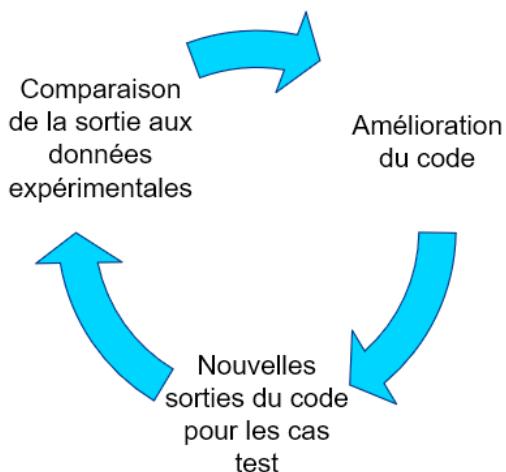


FIGURE 4.1 – Travail incrémental, permettant une comparaison permanente du rendu du code à des données théoriques ou expérimentales fiables

#### 4.1.2 Etat à la reprise des travaux

Le code actuel est composé de nombreuses fonctions commentées mais qui ne sont pas facilement compréhensibles et dont les liens entre elles sont obscurs. Certaines fonctions sont en plusieurs exemplaires avec pour seule différence la modification de la forme d'un élément de l'avion. Or le but du code devra être de pouvoir utiliser dans la même fonction différentes formes de fuselages ou d'éléments constitutifs de l'appareil. Ainsi ce défaut devra être réglé. De plus certaines fonctions sont fausses car les formules qu'elles utilisent le sont. Le débogage de ces fonctions sera nécessaire.

## 4.2 1<sup>er</sup> livrable : le guide du code

Ce guide est notre réponse au constant de la difficulté grandissante des groupes enjeux à comprendre les tenants et les aboutissants du code réalisé par leurs prédecesseurs. Nous avons ainsi fait le tri dans les 140 fonctions non répertoriées et très peu commentées qui nous avaient été fournies en début de projet, pour n'en conserver que 75 dans notre guide d'utilisation, que nous jugeons d'une plus grande utilité dans la compréhension du code dans son ensemble.

Ce guide, présenté sous forme de fiches afin d'être le plus concis et intuitif possible, doit permettre aux futurs utilisateurs du code et des bases de données XML de rapidement comprendre leur fonctionnement afin de pouvoir aisément travailler dessus. Il constitue à la fois un recueil explicatif de toutes les fonctions, un historique, et un guide d'utilisation du code.

### 4.2.1 Utilisation du guide

Chaque fiche, indépendante du reste du guide, permet à partir de la donnée du nom de la fonction et de son domaine, de retrouver l'ensemble des données nécessaires sur la fonction en question afin de la retravailler.

Elle reprend par ailleurs l'ensemble des hypothèses réalisées dans la modélisation de l'aspect du problème abordé par la fonction afin de faciliter l'identification des sources de divergence entre les sorties du code et de possibles études expérimentales réalisées en parallèle. Les plages de variations des paramètres importants utilisés dans la fonction sont aussi reprises afin d'expliquer les limites du code. Par exemple, nous avons restreint la modélisation de l'atmosphère dont l'altitude ne dépasse pas 16 km, ce qui impacte par exemple l'évaluation de la masse volumique de l'air environnant l'appareil, nécessaire pour évaluer les forces qui agissent dessus.

Pour que ce guide conserve son utilité, il doit être mis à jour à chaque nouvelle modification du code. Plus précisément, il faut toujours s'assurer que les informations sur les entrées et les sorties des fonctions, éléments sensibles du code, sont à jour. De même l'historique des fonctions du code doit être géré avec le plus grand soin, afin de conserver une vision d'ensemble de l'avancée du code.

### 4.2.2 Contenu du guide

Nous avons décidé d'organiser les fonctions du code en fonction de l'aspect de la modélisation de l'aérodynamique de l'avion auxquelles elles se rattachent. Ainsi, le guide est organisé en 8 grands domaines :

- Les fonctions **vitesse**  
Celles-ci permettent d'évaluer la vitesse du son dans l'air ainsi et la compressibilité de l'air en fonction de l'altitude de vol.
- Les fonctions **conditions extérieures**  
Ces fonctions permettent d'implémenter dans le code une modélisation de l'atmosphère dans lequel évolue l'appareil. Nous pouvons ainsi implémenter l'évolution de grandeurs fondamentales telles que la pression et la température de l'air en fonction de l'altitude de vol.
- Les fonctions **coefficients**  
Ces fonctions calculent les efforts appliqués par le milieu extérieur sur les différents composants de l'appareil (portance et trainées).
- Les fonctions **barycentre, foyer, masse**  
Ces fonctions reprennent les principales caractéristiques structurales des composants de l'appareil, afin d'en déduire des propriétés mécaniques globales sur l'avion telles que la masse totale (à vide et remplie).

- Les fonctions **surface**

Ces fonctions reprennent les informations des composants de l'appareil nécessaires à l'évaluation des forces aérodynamiques qui s'y appliquent.

- Les fonctions **paramètres**

Ce sont des fonctions exclusivement utilisées pour modéliser la forme des ailes, tel que son profil ou son allongement.

- Les fonctions **test**

Ces fonctions permettent d'automatiser les études réalisées sur le code, afin de réduire le temps de préparation et d'implémentation de chacune d'entre elles.

- Les fonctions **XML**

Ces fonctions permettent de rattacher le code à la base de donnée utilisé.

Chacune des 75 fonctions utiles du code est présentée à l'aide d'une fiche individuelle, qui reprend les principales informations sur son contenu :

- Le titre de la fonction
- Le domaine auquel elle est rattachée
- Les objectifs de cette fonction
- Les entrées du code et la manière de les entrées dans le code
- Les sorties du code et leur nature
- Une explication de la physique modélisée dans le code
- Le chemin d'accès vers cette fonction dans le code
- Une liste des fonctions auxquelles cette fonction fait appel
- L'historique de modification

Titre		Domaine
Objectifs		
Entrées du code	Comment les entrer dans le code	
Sorties du code	Nature de ces sorties	
Physique du code		
Chemin d'accès (lien FTPS)	Les fonctions auxquelles ce code fait appel	Historique des modifications (explications)

- **Coefficients aérodynamiques**
- **Surfaces**
- **Conditions extérieures**
- **Barycentres-foyers-masses**
- **Tests**
- **Vitesses**
- **Analyse**
- **Fonction XML**

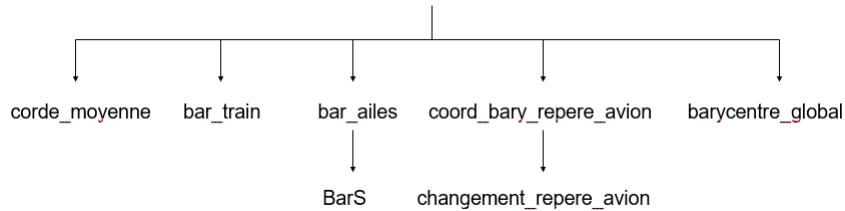
FIGURE 4.2 – Fiche type du guide et couleurs associées aux domaines

## 4.3 2<sup>ème</sup> livrable : la refonte de la structure du code

L'objectif de notre client Xavier Dutertre était que nous réalisions un code qui soit intuitif : c'est-à-dire qu'un néophyte, quelqu'un qui n'a pas travaillé sur le projet, puisse l'utiliser aisément. C'est dans cette optique que nous avons conçu le guide et c'est également pour cette raison que nous avons fait une refonte de la structure du code.

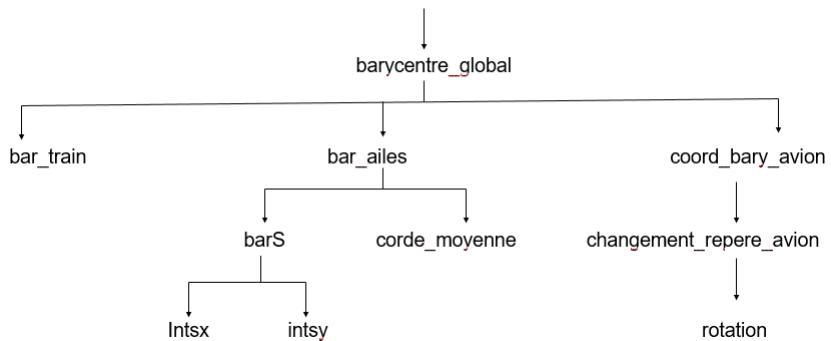
En effet, le code ayant été écrit par de nombreuses équipes qui se sont succédées, cette aspect itératif se retrouve dans le code lui-même : celui-ci se présente comme un patchwork de fonctions indépendantes non hiérarchisées. Ainsi pour réaliser une tâche simple, la structure étant « linéaire », l'utilisateur doit connaître toutes les fonctions auxquelles il doit faire appel et l'ordre dans lequel il doit le faire.

Explication sur un cas d'utilisation simple : l'utilisateur veut calculer le barycentre de l'avion. Avec l'ancienne structure linéaire, voici tous les appels aux fonctions intermédiaires, et leur ordre, qu'il devait effectuer avant de pouvoir calculer le barycentre global :



Cette complexité qui interdit à tout néophyte l'utilisation du code était en contradiction avec l'objectif de M. Dutertre. Nous avons donc souhaité repenser la structure du code. Grâce à la vision d'ensemble que nous a donné l'inventaire des fonctions, nous avons pu hiérarchiser les fonctions : nous avons ainsi fait la distinction entre les fonctions renvoyant une sortie importante pour l'utilisateur, et celles qui ne sont qu'un intermédiaire vers ce résultat.

Dans la nouvelle structure hiérarchisée « en cascade », voici comment se déroule le calcul du barycentre de l'avion :



Ici, les flèches seules témoignent de l'appel d'une fonction supérieure à une fonction intermédiaire. On remarque qu'avec cette nouvelle structure, l'utilisateur n'a plus qu'à faire appel à une seule fonction globale, le travail fastidieux d'appel aux fonctions intermédiaires se faisant automatiquement. Celui-ci n'a plus à connaître la structure du code, il a seulement besoin de chercher dans le guide le nom de la fonction globale qui renvoie le résultat voulu.

### 4.3.1 Etat des lieux de sortie du code

Pour le groupe suivant, voici la façon dont est organisé le dossier contenant le code. Il est divisé en 3 dossiers :

- le dossier « ancienne structure » : ce dossier contient la version antérieure du code, avant le passage à la nouvelle structure utilisant la BDD.
- le dossier « code organisé » : ce dossier contient le code organisé de la même façon que le guide d'utilisation pour que vous puissiez vous y retrouvez facilement dans l'ensemble des fonctions. Chacun des sous-dossiers est divisé en 2 entre « fonctions converties à la BDD » et « fonctions à convertir à la BDD ».
- le dossier « code global » contient l'ensemble des fonctions du code dans leur dernière version dans un seul répertoire pour que le code puisse être exécuté.

L'étendue des modifications apportées au code, qui est fourni dans ces dossiers, est détaillée dans les parties qui suivent.

## 5 Etude expérimentale : écoulement de l'air atour d'une aile du Bee-Plane<sup>TM</sup>

Dans le cadre de l'UE Activités Expérimentales de l'école, nous avons eu l'opportunité d'utiliser une soufflerie afin de réaliser des expériences sur une maquette de l'aile du Bee-Plane afin d'étudier expérimentalement son comportement dynamique dans un écoulement d'air. Nous désirions trouver l'angle d'attaque optimal afin de réduire la consommation de carburant, et ce en déterminant l'angle pour lequel le rapport portance/trainée est le plus faible. Pour cela, nous avons pris le fichier modélisant numériquement l'aile de Bee-Plane disponible sur le serveur, et les techniciens de l'école l'ont imprimé à l'aide d'une imprimante 3D. Suite à ça, nous avons conçu un montage permettant de mesurer la vitesse de l'écoulement, l'angle d'incidence, la portance et la trainée. Voici une photo montrant le montage et l'aile en question :



FIGURE 5.1 – Le montage expérimental

Nous avons procédé en deux temps : premièrement, à angle d'incidence fixé, nous avons observé les portances et trainées à différentes vitesses d'écoulement afin de comprendre l'influence de la vitesse de l'écoulement sur ces grandeurs, puis nous les avons ensuite observées à différents angles en gardant la vitesse d'écoulement constante, afin de finalement trouver le rapport qui nous intéressait.

Pour rendre cette expérience valable, le nommbre de Reynolds doit être le même que celui des conditions réelles de vol.

$$R_e = \frac{vL}{\nu}$$

Avec  $\nu_{air} = 1,7 \cdot 10^{-5} \text{ Pa.s}$  à  $T = 273,15 \text{ K}$ , constante de l'expérience.

$$\underbrace{v_{exp} L_{exp}}_{exerience} = \underbrace{v_r L_r}_{realite}$$

avec  $v_r = 700 \text{ km/h}$  fixée et  $L_r$  fixée par le BEE-PLANE<sup>TM</sup>.

La maquette fixant  $L_{exp}$ , il ne reste plus qu'à déterminer  $v_{exp}$  afin de garder le même nombre de Reynolds que dans les conditions réelles.

Voici quelques observations montrant le comportement de certaines grandeurs dynamiques en fonction des paramètres extérieurs (angle d'attaque et vitesse d'écoulement) :

**Trainée en fonction de la vitesse à 15°**

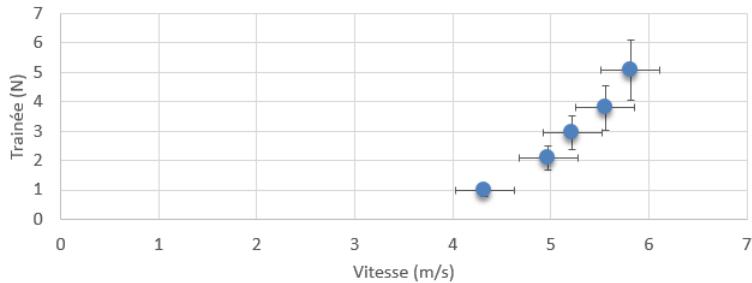


FIGURE 5.2 – Trainée en fonction de la vitesse à 15 degrés (angle d'attaque). Nous avons choisi de ne prendre des mesures qu'à partir de 4m/s car nous ne nous intéressons qu'à la phase de vol.

**Portance en fonction de l'angle à 18.2 m/s**

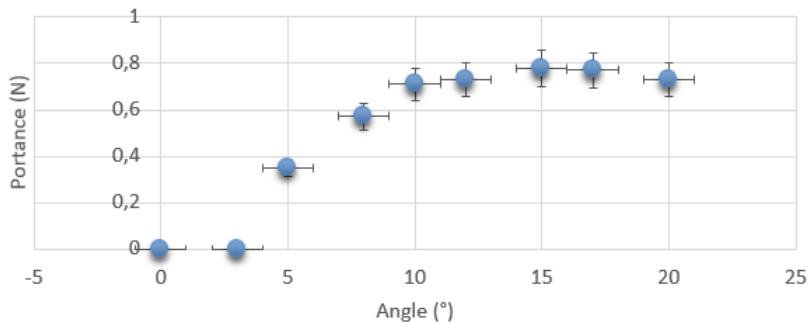


FIGURE 5.3 – Portance en fonction de l'angle d'incidence à 18,2 m/s (vitesse de l'écoulement) avec une marge d'erreur de 1 degré sur l'angle et de 10% sur la portance.

A partir de toutes nos données et des recherches bibliographiques, nous avons pu réaliser les courbes de portance en fonction de l'angle d'attaque, la trainée en fonction de la vitesse et la portance en fonction de la trainée.

5. Etude expérimentale : écoulement de l'air atour d'une aile du BEE-PLANE<sup>TM</sup>

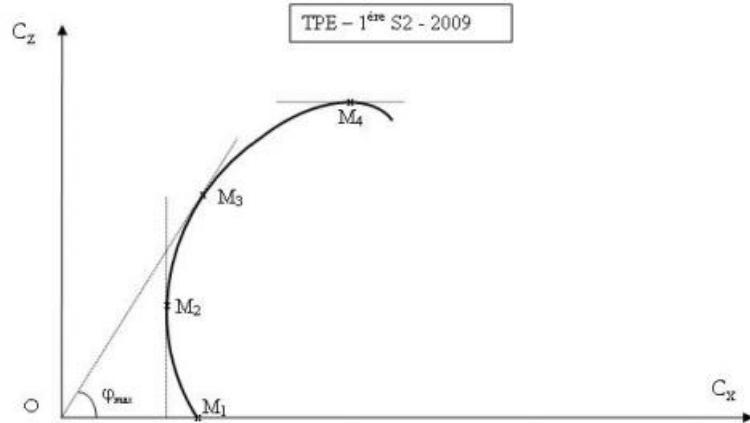


FIGURE 5.4 – Le point  $M_3$  correspond au point de finesse maximale, c'est lui que nous cherchons à obtenir.

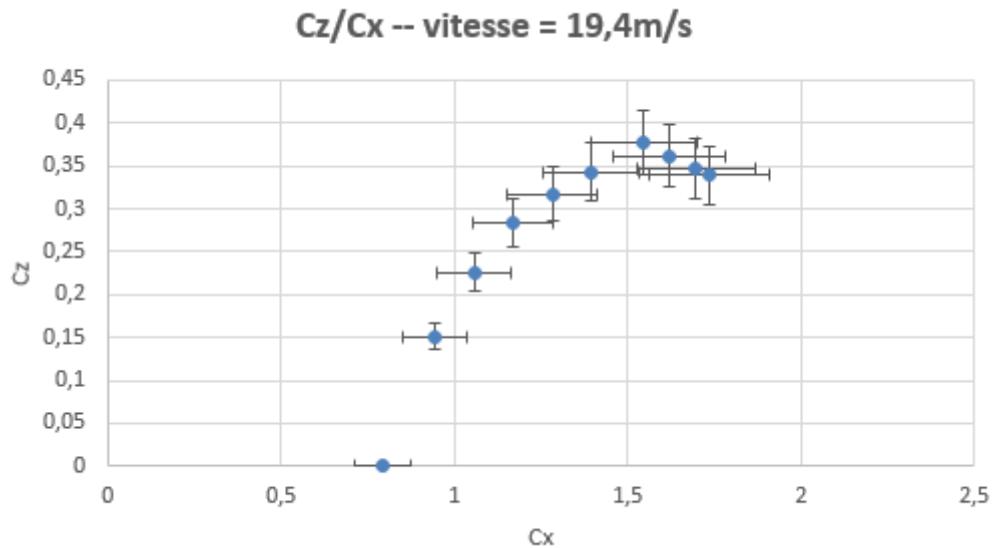


FIGURE 5.5 – Rapport portance/trainée à 19,4 m/s. La finesse maximale de l'aile vaut 14.

L'angle optimal est obtenu en traçant la tangente à la dernière courbe passant par l'origine. Cette tangente permet de trouver un angle de finesse optimal compris entre 10 et 12 degrés.

# 6 Nouvelles fonctions implémentées dans le code

Afin de respecter les contraintes du cahier des charges du code, fixé avec le client et ajusté en cours de projet, de nouvelles fonctions ont dû être implantées dans le code. Par ailleurs des fonctions déjà présentes dans le code, dont la modélisation n'avait pas été justifiée ou nous semblait erronée, ont été réécrites de toute pièce afin d'assurer la fiabilité du code dans son ensemble. Enfin notre travail sur la base de données nous a amené à écrire des fonctions XML pour en simplifier l'utilisation.

## 6.1 Fonctions rattachées à la Physique du code

Tout modèle étant par essence faux, nous avons trouvé important de rappeler l'ensemble des hypothèses réalisées dans la modélisation préalable au codage à proprement parler de la fonction correspondante. Cette phase est d'autant plus importante qu'elle permet de définir les limites d'usage du code, informations fondamentales pour utiliser les informations recueillies dans la conception à proprement parler du BEE-PLANE™. Ce travail n'aurait pas été possible sans la coopération avec le groupe TMO-01, qui nous a aidé dans l'énumération et la justification des hypothèses propres à chaque modèle.

La description que nous faisons par la suite de certaines de ces fonctions est pour nous un moyen pour nous de transmettre aux futurs équipes la démarche qui a été la nôtre au cours de ce projet, et d'expliquer aux futurs utilisateurs du code.

### 6.1.1 Conditions extérieures : la fonction `masse_vol_atm (T,alt)`

Cette fonction se base sur le modèle de l'atmosphère standard et tient compte de l'évolution de la température avec l'altitude, paramètre fondamental pour décrire les variations de masses volumiques observées par l'avion en fonction de sa phase de vol.

Les hypothèses retenues pour accéder à la masse volumique de l'air sont :

- Un air sec
- Le modèle des gaz parfaits :  
$$p = \rho \cdot r \cdot T \quad (r = 287 \text{ J.K}^{-1}.\text{kg}^{-1})$$
- Un modèle statique :  
$$dp = -\rho \cdot g_0 \cdot dz$$

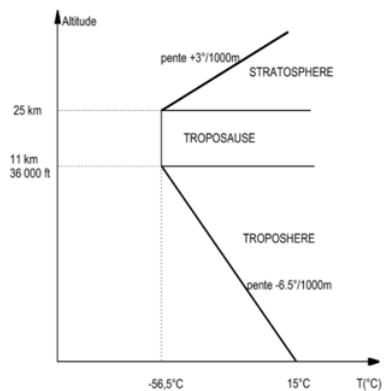


FIGURE 6.1 – Approche retenue pour l'évolution de la température avec l'altitude

### 6.1.2 Estimation de la consommation : la fonction consommation()

La fonction initialement présente dans le code pour évaluer la consommation de l'avion au cours d'un vol ne prenait pas en compte les phases de décollage et d'atterrissage dans son estimation. Or des estimations de Fermi réalisées en début d'année en collaboration avec le groupe TMO-01 nous ont permis de quantifier l'influence de ces phases dans la consommation totale d'un avion au cours d'un vol, et d'en souligner l'importance pour des vols court-courrier (tels ceux potentiellement réalisés par le BEE-PLANE<sup>TM</sup>).

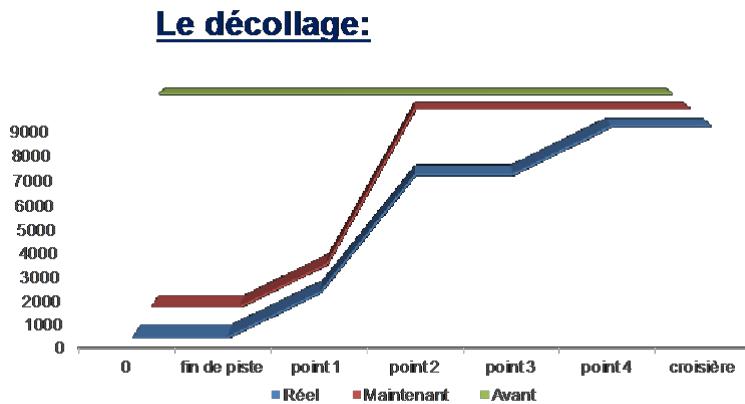
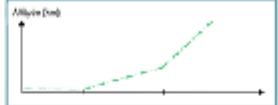


FIGURE 6.2 – Modèle retenu pour l'évolution de l'altitude de l'avion en fonction de la phase de vol

Nous utilisons ici les données moteurs et modélisations réalisées par les groupes de l'ESTACA 2014-2015 travaillant sur cet aspect du BEE-PLANE<sup>TM</sup>. Comme conseillé dans le rapport rédigé par l'ESTACA, nous avons décidé de modélisé la phase descente/atterrissage par une descente linéaire et non par un échelon comme c'était le cas dans le code antérieur.

Phase de vol	Modélisation	Allure de la trajectoire
Décollage/montée	1/ Consommation spécifique : $CS(M) = CS_0 \cdot (1 + M)$ 2/ Poussée du système de propulsion : $F$ 3/ Consommation horaire : $CH(M) = F \cdot CS(M)$ 4/ Consommation sur chaque phase de vol sachant : $d_{\text{phase}}(i)$ et $V_{\text{phase}}(i)$ , $i \in [11, 12, 2, 3]$	
Vol en croisière	$R = \eta_0 \frac{L}{D} \frac{Q_r}{g} \ln\left(1 + \frac{M_{\text{fuel}}}{M_{\text{finale}}}\right)$ Données : $\eta_0$ : Rendement global du système de propulsion $L, D$ : Portée et trainée de l'avion, en palier $Q_r$ : Capacité calorifique du kérósène $M_{\text{fuel}}, M_{\text{finale}}$ : Respectivement la masse de kérósène embarquée et la masse du BEE-PLANE <sup>TM</sup> à l'atterrissage	
Descente	Même implémentation que pour la phase de décollage, avec un taux de descente imposé	

### 6.1.3 Limitations par ces fonctions

Nous avons remarqué que cette fonction restreignait l'altitude maximale de vol en croisière en fonction de la la distance totale du vol, ce qui était prévisible puisque nous imposons le taux de descente de l'avion dans la phase de descente/atterrissage.

Nous avons donc dû modifier la fonction évaluant la consommation de l'avion au cours de son vol pour tenir compte de cette contrainte. Après modélisation, nous obtenons la courbe ci-joint, qui permet de relier l'altitude maximale de vol en croisière qui peut être demandée au code à la distance du vol que l'on souhaite étudier. Nous avons décidé de fixer le taux de descente au taux de descente moyen imposé par la réglementation en vigueur pur ne pas complexifier inutilement le code à ce stade, mais cette hypothèse pourrait être assouplie dans des versions futures du code.

Par ailleurs nous avons décidé de fixer l'altitude de croisière comme étant inchangée tout au long du vol. Cette hypothèse forte, et rarement validée dans la réalité, pourrait être retravaillée à l'avenir si l'on remarquait que cela pourrait influencer notablement l'évaluation de la consommation.

De même les couloirs aériens n'ont pas été pris en compte dans notre modélisation, la trajectoire entre l'aéroport de départ et l'aéroport d'arrivée étant assimilée à une droite. Cette sous-estimation de la distance totale du vol pourrait aussi être réévaluée selon les exigences en terme de précisons du code du client.

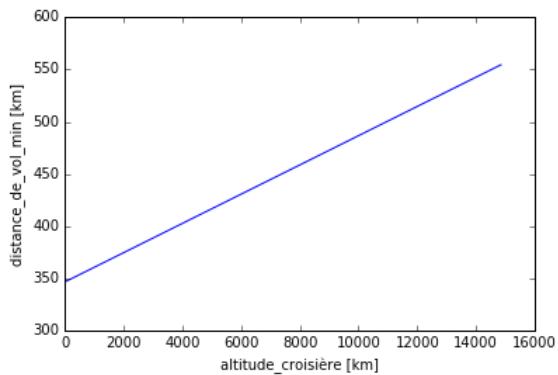


FIGURE 6.3 – Evaluation de la distance minimale de la phase de croisière



FIGURE 6.4 – Carte des couloirs aériens en France (source : DGAC)

## 6.2 Fonctions rattachées à la base de données

Le code qui nous a été fourni en début d'année était redondant et très difficile d'utilisation : ainsi, lorsque l'utilisateur voulait travailler sur un avion, il devait rentrer à la main toutes les informations concernant son avion (listées en partie 3.1.1.).

De plus, ces informations étaient stockées sous forme de lignes de codes en MatLab, ce qui nécessitait d'avoir une version du programme principal par avion étudié. Dans le code fourni, nous avions ainsi 4 versions du programme « Analyse\_Parametrique.m » :

- « Analyse\_Parametrique\_A320.m »
- « Analyse\_Parametrique\_A321.m »
- « Analyse\_Parametrique\_Bee\_ailes.m »
- « Analyse\_Parametrique\_Bee\_sans\_basket.m »

Ceci était en contradiction avec l'objectif de M. Dutertre d'avoir un code facilement utilisable pour tous types d'avion. C'est pourquoi il nous a semblé nécessaire d'implémenter, en collaboration,

avec le groupe TMO-01 une base de données d'avion, codée en XML.

Le groupe TMO-01 s'est occupé de la création et d'un premier remplissage de cette base de données. Celle-ci est un ensemble de 700 lignes de codes permettant d'accéder à toutes les données des avions ; c'est-à-dire les données concernant les moteurs, les ailes, le fuselage, les trains d'atterrissement. Chaque ligne est une branche de l'arbre, ou un nœud, et les branches les plus profondes sont des feuilles.

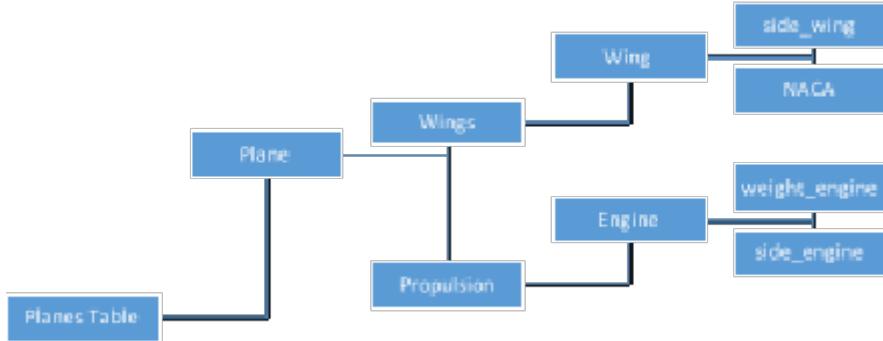


FIGURE 6.5 – Arbre d'accès aux informations contenues dans la base de donnée

Sa structure est détaillée plus en détail dans le rapport du groupe TMO-01.

Nous nous sommes occupés de la conversion des fonctions du code à la nouvelle structure, ce qui a demandé de changer toutes les entrées du code, qui apporte un changement radical de philosophie : l'utilisateur a maintenant accès à une base de données d'avions existantes indépendante du code. Il n'y a donc plus besoin d'avoir un programme spécifique par avion, les programmes peuvent faire appel à la base de données et l'utilisateur peut la modifier s'il le souhaite.

Pour faciliter le travail de l'utilisateur qui ne connaît pas forcément la syntaxe associée à l'utilisation d'une BDD, nous avons implémenté divers fonctions XML : Ceci nous a par exemple permis de réduire de 40 à 1-2 le nombre de lignes de code nécessaires pour rechercher un moteur.

Les requêtes pour accéder aux données de cette base nécessitent de connaître un langage très particulier. Ainsi pour ne pas avoir à l'apprendre, nous avons créé des fonctions qui réalisent ces actions. Ces fonctions sont au nombre de 5 :

- Attribut : permet de connaître la valeur d'un nœud de la base de données. La valeur donnée est une chaîne de caractères. On pourrait perfectionner la fonction en demandant le type de donnée que l'on veut recevoir (string ou number).
- Recherche\_moteur : permet de rechercher le nœud contenant les données d'un moteur. Elle est très pratique lorsque l'on étudie un avion et que l'on ne peut connaître que le nom générique du moteur.
- Recherche\_avion : permet de lancer le programme en recherchant dans la base de données le nœud correspondant à l'avion recherché.
- Ajout\_nœud : lorsque les programmes se lancent, certaines données temporaires sont stockées dans la base de données sous la forme de nœuds temporaires. Cette fonction permet de créer un nouveau nœud et de le placer dans l'arbre à l'endroit voulu.

- Modification\_noeud : permet de faire varier des paramètres d'un nœud pendant l'exécution du programme.

Ces fonctions sont suffisantes pour utiliser le code de manière confortable.

# 7 Planning - Organisation de l'équipe

## 7.1 Planning prévisionnel

Après avoir décomposé notre objectif principal en sous-objectifs, nous avons pu planifier notre travail sur l'année à l'aide d'un diagramme de Gantt. Celui-ci permet de rendre compte de façon incrémentale de l'avancée du projet : chaque tâche étant indiquée avec sa durée estimée, le diagramme permet de visualiser rapidement tout retard éventuel et donc de rester à tout moment réactif dans nos travaux.

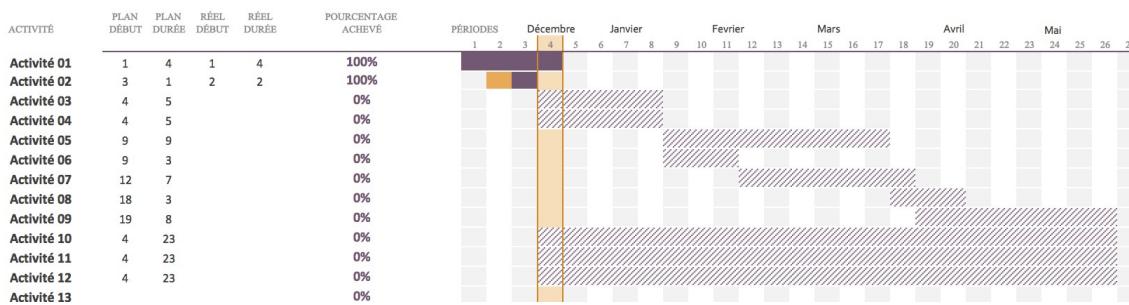


FIGURE 7.1 – Diagramme de Gantt du groupe TMO-02 (à la mi-décembre 2015)

- Activité 1: Recherche bibliographique et présentation
- Activité 2: Inventaire des logiciels de modélisation d'avion
- Activité 3: Inventaire des inputs du code
- Activité 4: Liste des formules physiques et incertitudes sur les formules (code existant)
- Activité 5: Conception d'une base de données accessible
- Activité 6: Contact avec les laboratoires (soufflerie)
- Activité 7: Tests avec le code sur des modèles d'avion
- Activité 8: Création du dossier de tests
- Activité 9: Elaboration d'une interface code/utilisateur
- Activité 10: Création du guide d'utilisation
- Activité 11: Implémentation des formules physiques de TMO 01
- Activité 12: Travail sur le code

FIGURE 7.2 – Liste des tâches à réaliser

## 7.2 Organisation interne

Après avoir identifié les tâches à réaliser, nous les avons réparties entre les différents membres de l'équipe, selon les affinités de chacun. Pour cela, nous avions dressé le profil des membres de l'équipe grâce à la méthode Belbin : outil présenté lors de l'APP2 qui permet notamment d'identifier les qualités naturelles de chaque membre d'une équipe, afin d'augmenter l'efficacité de l'ensemble de l'équipe.

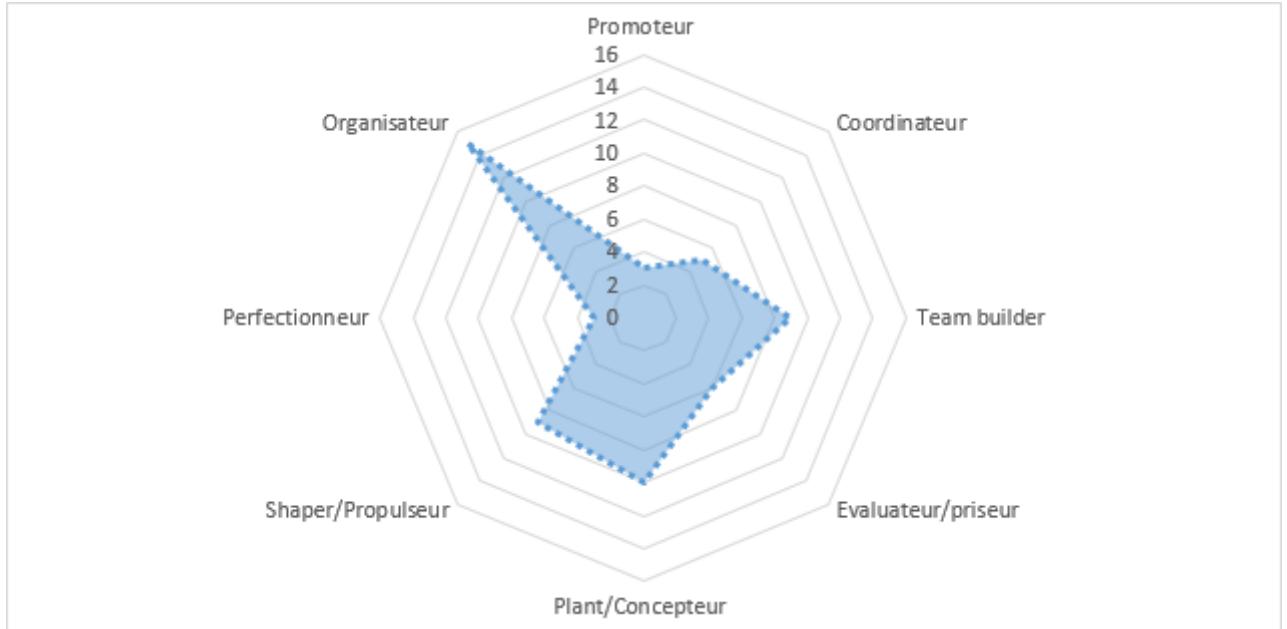


FIGURE 7.3 – Diagramme Belbin de l'équipe TMO-02

Il en ressort une forte capacité au sein du groupe à s'organiser, à gérer une équipe et à concevoir les tenants et les aboutissants de notre projet. A contrario, il faudra être attentif à la coordination au sein de l'équipe, et à notre capacité à rendre un travail fini précis, soigné et pertinent vis à vis de notre problématique, puisque c'est dans ces domaines que nous semblons avoir le plus de lacunes. Ce découpage permet ensuite à chacun de travailler en autonomie ou en binômes, selon ses disponibilités, dans un souci d'efficacité.

Mais il est important de rester en contacts réguliers pour s'assurer de l'avancée de chacun et de rester en cohérence avec les objectifs fixés lors de l'élaboration de notre diagramme de Gantt. Pour ce faire, nous avons créé des outils de partage :

- Un groupe WhatsApp : pour communiquer et s'entraider si besoin
- Un Google doc : pour fixer par écrit nos tâches et leur avancement
- Un Google Drive : pour partager nos travaux et nos études

Enfin, nous effectuons une réunion interne avec notre référent pédagogique Cédric Mehl et ce au moins une fois toutes les deux semaines. Ces réunions sont l'occasion de faire la synthèse de nos avancées, synthèse prenant la forme physique d'un CR, ainsi que de tenir notre client, Xavier Dutertre, informé de l'état de nos travaux.

## 7.3 Interaction avec les membres du projet extérieurs au groupe

L'organisation externe pour le projet Bee-Plane est primordiale, et ce à deux niveaux :

### 7.3.1 Au niveau de l'école Centrale Paris

Le projet BEE-PLANE<sup>TM</sup> est un projet enjeu partagé par 2 groupes : TMO-01 et TMO-02 aux objectifs complémentaires ; d'une part la recherche d'un modèle physique du vol adéquat, d'autre part l'implémentation de ce modèle en MATLAB afin d'être capable de déterminer les paramètres nécessaires à l'optimisation de la configuration aérodynamique de l'avion.

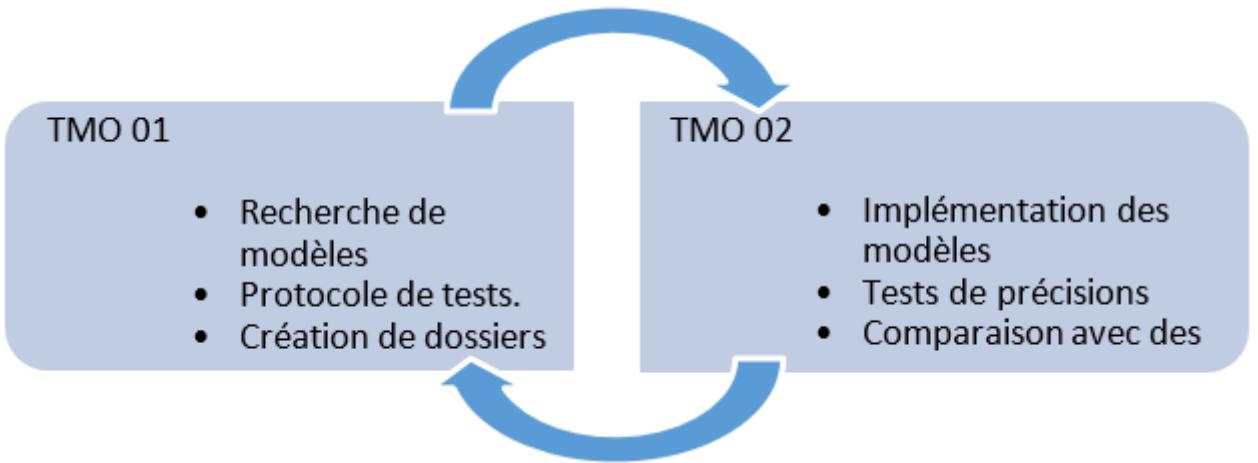


FIGURE 7.4 – Schéma synthétique des interactions entre les 2 groupes du projet BEE-PLANE™

Il est donc nécessaire de mener ce projet avec une vision double : conserver cette dynamique de travail en interaction, tout en s'assurant d'une certaine indépendance afin de ne pas être bloqué si le groupe TMO-01 ne renvoie plus de résultats.

Pour ce faire, nous avons également créé un deuxième groupe WhatsApp, pour pouvoir échanger rapidement et planifier nos travaux en interaction. De plus, nos appels au client se font en commun, pour présenter une vision cohérente et synthétique de l'avancée de notre projet.

### 7.3.2 Au niveau universitaire

Le projet BEE-PLANE™ est un projet collaboratif entre la société de Xavier Dutertre, Technoplane SAS, et des universités et écoles d'ingénieurs en charge de l'aspect de R&D. Ainsi par exemple :

- L'école Centrale Paris est en charge de la partie aérodynamique
- SupMéca s'occupe du train d'atterissage
- L'ESTACA s'occupe de la propulsion

Il est ainsi indispensable d'intégrer les travaux des autres écoles, dans l'implémentation de notre code, afin de s'assurer que nous livrerons au client une partie cohérente d'un tout, et non un greffon indépendant et inutilisable.

Ainsi par exemple, nous avons prévu de réaliser une base de données permettant notamment de prendre en compte les paramètres moteurs envoyés par SupMéca, données auxquelles notre code Matlab aura accès. Nos échanges de données se font par l'intermédiaire du serveur de Xavier Dutertre : Technoplane.

## 8 Les difficultés rencontrées par le groupe TMO-02 jusqu'à présent

Tout au long de notre projet, nous avons dû faire face à divers problèmes pour continuer d'avancer dans notre démarche d'amélioration de l'aérodynamique du BEE-PLANE™. Si ces périodes de réflexion et de remise en question se sont montrées très chronophages, ce sont sûrement elles qui nous ont le plus fait progresser, que ce soit au sein du projet ou dans la démarche plus globale d'appréhension de ce qu'est le travail de l'ingénieur au sein d'une équipe.

Cette section vise aussi à aider les équipes futures dans leur démarche, afin qu'elles ne perdent pas de temps à refaire les mêmes erreurs que nous, mais aussi qu'elles réalisent que notre travail final a été précédé de phases d'incertitudes et de remise en question, nécessaires et constructives.

Le premier problème auquel nous avons eu à faire face, et que nous n'avons toujours pas entièrement réglé au moment de l'écriture de ce rapport intermédiaire, est la compréhension du code qui nous a été fournis. En effet, nous avons débuté notre projet avec un code qui avait été réalisé puis complété par déjà plusieurs équipes avant nous. N'ayant aucun guide à disposition pour nous expliquer la démarche, la logique, qui se cachait derrière chacune des lignes de code, nous avons du lire l'ensemble du code à plusieurs reprises avant de comprendre ce qu'il réalisait. Cette phase nous a aussi permis de cerner les zones de codes qui seraient à revoir ou du moins à tester de nouveau. C'est dans ce contexte que nous nous sommes mis d'accord sur l'élaboration d'un guide d'utilisateur pour faciliter le travail de compréhension du code pour les équipes futures.

Un autre problème qui s'est posé à nous a été la répartition du travail entre les deux équipes. Ce problème découle en réalité du 1<sup>er</sup> problème expliqué ci-dessus, puisqu'en fait c'est l'appréhension de chacun vis-à-vis du code qui a bloqué tout échange entre les deux équipes, chacune se renvoyant la charge d'avoir à travailler sur le code. Ce problème a été résolu, suite à de longues discussions, grâce à des concessions de la part de chacune des deux équipes, notre groupe acceptant de travailler sur le code à condition d'avoir aussi des tâches en lien plus direct avec les domaines de l'aéronautique et de la physique. C'est en fait notre appréhension vis-à-vis du caractère intangible et complexe du codage qui nous a surtout dissuadé de travailler le code plus en profondeur, dans un premier temps. Il serait donc intéressant que les équipes enjeux à venir prennent le temps de comprendre ce qui est vraiment attendu d'elles au niveau du code, qui d'autre part est loin d'être l'unique élément à travailler dans ce projet même si tout s'y rapporte.

Enfin, le dernier problème auquel nous avons eu à faire face jusqu'à présent a été de voir comment évaluer quels algorithmes de calculs des coefficients aérodynamiques déjà implantés dans le code étaient pertinents et donc à conserver, et lesquels seraient à revoir, que ce soit pour une précision insuffisante ou une implémentation hasardeuse. Il s'agit donc ici d'un problème directement lié au travail que nous avons mené tout au long de l'année. Cette question, qui s'est posée très vite dans notre phase préalable de compréhension du code, n'a toujours pas été résolue et devra donc être abordée plus sérieusement dans les semaines à venir.

## 9 Bibliographie

Groupe TMO 01, « Rapport intermédiaire TMO 01 », 2015

Groupe TMO 02, « Rapport intermédiaire TMO 02 », 2015

« Les statistiques mondiales en temps réel ». <http://www.planetoscope.com/transport/Avion>, 2012

« Objectifs de l'avion Bee-Plane ». <http://www.bee-plane.fr/#/objectifs/3994087>, 2014

« Quand une Airbus roule à l'électricité », [pluse.edf.com](http://pluse.edf.com), 2016

Ateliers du développement Professionnel & Leadership. « Trace » p16-17 et p70-71. Dimitri Dagot & Serge Dell-Vedove

« Aircraft range : the Breguet range equation » MIT.

<http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node98.html>

Sciences. « Un avion à structure innovante : le Bee-Plane », 2012

Sciences et Avenir. « Bee-Plane, l'incroyable projet d'avion... détachable ».

<http://www.sciencesetavenir.fr/high-tech/20150618.OBS1057/bee-plane-l-incroyable-projet-d-avion-detachable.html>, 2015

« Bien définir un problème pour mieux le résoudre ».

<http://www.techniques-ingenieur.fr/fiche-pratique/genie-industriel-th6/piloter-et-animer-la-qualite-dt34/bien-definir-un-probleme-pour-mieux-le-resoudre-0571/>, 2015

« ADS - Aircraft Design Software ». <http://www.pca2000.com/fr/software/ads-generalites.php>, 2012

Présentation « La méthode du Ludion », Paul-Hubert des MESNARDS, 2015

AIRBUS INDUSTRIES. Airbus, [www.airbus.com](http://www.airbus.com), 2015

# A Présentation non-exhaustive des logiciels existants

## A.1 Le logiciel CATIA

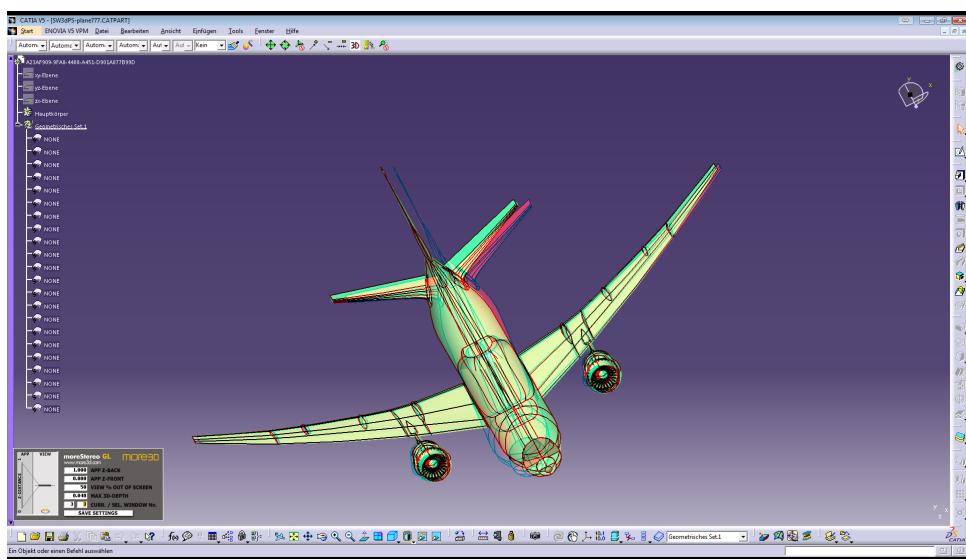


FIGURE A.1 – Un autre logiciel très utilisé dans l'industrie est CATIA : Conception Assistée tridimensionnelle Interactive Appliquée

Produit par DASSAULT SYSTEMES, le logiciel CATIA couvre un très grand nombre de domaines dont la conception mécanique, la conception de systèmes électriques, ainsi que pour le design et la CAO. Son grand atout est sa capacité à modéliser le comportement de l'appareil en contexte, ce qui permet d'affiner encore plus l'étude en fonction des différentes phases du vols ce qui est fondamental en aéronautique (de par les variation de pression et de température qu'il peut y avoir selon que l'avion est en vol ou sur le tarmac). Ce logiciel serait plus un complémentaire qu'un substitut au logiciel que nous construisons puisqu'il ne décrit pas l'aérodynamique du système.

## A.2 Le logiciel MapleSim

Développé par Maple Soft, MapleSim est un outil de modélisation et de simulation de phénomènes physiques. Celui-ci génère des équations de modèle et permet la simulation et l'analyse de phénomènes en utilisant le moteur de Maple. Ce logiciel peut plus particulièrement être appliqué à l'aéronautique aux travers d'un module spécifique. Toutefois ce logiciel n'est pas expressément conçu pour modéliser l'aérodynamique d'un avion mais plutôt pour concevoir un pilotage automatique par calcul formel pour un avion donné.

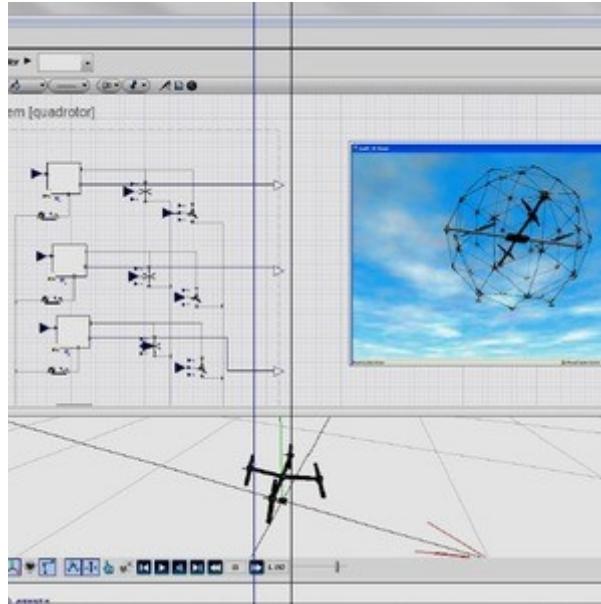


FIGURE A.2 – Capture d’écran du logiciel Maplesim avec un schéma logique de pilote automatique pour drone

### A.3 Le logiciel AutoCad

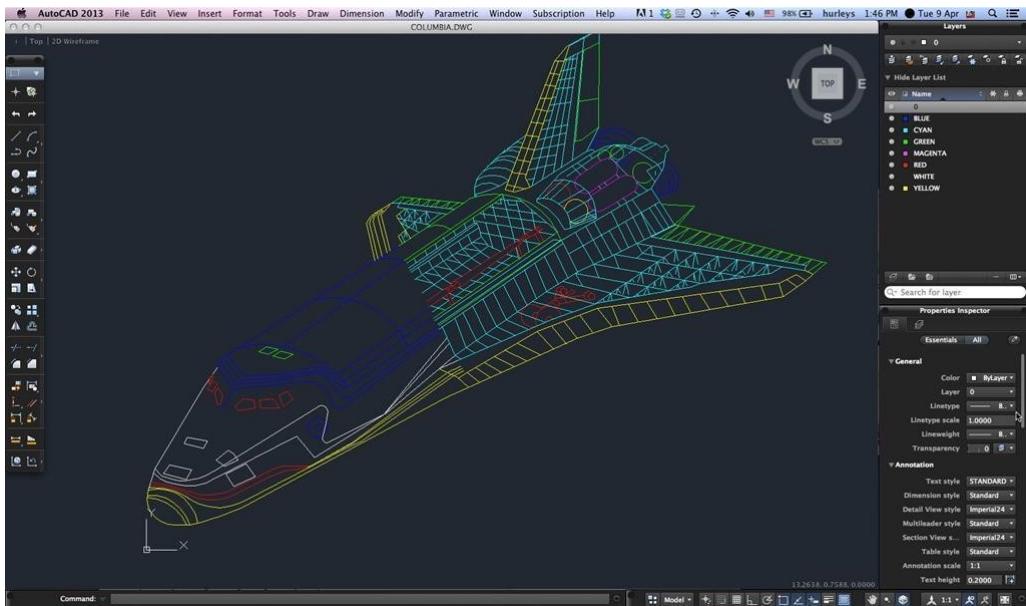


FIGURE A.3 – Capture d’écran d’un plan 3D de navette spatiale réalisé sous AutoCad

AutoCAD est un logiciel de dessin assisté par ordinateur (DAO). Celui-ci permet de modéliser la forme 3D d’un objet quelconque. Ce logiciel présente des caractéristiques similaires au logiciel CATIA présenté en (i). Il n’est donc pas adapté à la modélisation aérodynamique d’un avion mais permet l’établissement de plans 2D et 3D nécessaire à la conception d’un appareil.