RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ BATNA 2
FACULTE DE TECHNOLOGIE

Département de
GENIE MECANIQUE

MEMOIRE DE FIN D'ETUDES

PRESENTE POUR OBTENIR LE DIPLOME DE

MASTER

Filière : **Génie mécanique**

Spécialité : **Energétique**

PAR : BITAM ABED ERAOUF

THEME

Contribution au développement du logiciel « PompAx » : Conception 3D d'une pompe axiale.

Soutenu le 30 /06 /2018

Proposé et dirigé par: **Dr. Laïd MESSAOUDI**

Année Universitaire 2017-2018

Table des matières

LISTES DES FIGURES

REMERCIEMENT

On remercie, avant tout, le Bon Dieu de nous avoir donné la patience, le courage et de nous avoir facilité le chemin pour achever ce fruit de nos longues années d'études.

Au fur et à mesure que je tentais de me remémorer tous les gens à qui je désirais exprimer ma gratitude pour leurs appuis, leurs suggestions et tous les efforts qu'ils avaient déployés pour permettre à ce travail de voir le jour, la liste ne cessait de s'allonger.

Je voudrais tout d'abord remercier vivement Dr LAID MESSAOUDI de m'avoir accepté et dirigé dans cette thèse et de m'avoir accordé de son temps et de ses conseils très précieux, sans lesquels ce travail aurait été pénible voire même impossible à réaliser.

J'adresse mes remerciements respectueux aux les membres honorables du jury qui ont bien voulu expertiser mon travail.

Enfin je voue une gratitude particulière aux aux mes parents, ainsi qu'à toute personne ayant contribué, par son aide ou son soutient au bon déroulement de ce travail.

DEDICACE

JE dédie ce modeste travail à :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chères sœurs ……. pour leurs encouragements permanents, et leur soutien moral,

A mes chers frères, ….. pour leur appui et leur encouragement,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Pour mes très chers amis :

Djalal, Lazhar , Isaam, Idriss ,Rami ,Hicham

Hanan, Imen, sakina

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fuit de votre soutien infaillible,

Merci d'être toujours là pour moi

Introduction générale

Une pompe à flux axial, ou AFP, est un type commun de pompe qui consiste essentiellement en une hélice (roue axiale) dans une conduite, l'hélice peut être entraînée directement par un moteur scellé dans une cylindre ou par un moteur électrique ou des moteurs à essence / diesel montés sur un cylindre de l'extérieur ou par un arbre d'entraînement à angle droit qui perce le cylindre. Elle est constituée par un cylindre en fonte à l'intérieur duquel se trouvent un stator (fixe) et un rotor (mobile) en acier qui tourne tangentiellement au stator. Solidaires du rotor, les palettes peuvent coulisser et sont maintenues en contact avec les parois du stator.

Les pompes axiales, également appelées pompes à hélice, sont utilisées dans des applications qui font appel à de gros débit avec de faibles hauteurs. Ces pompes sont construites presque exclusivement à un seul étage, on n'a recours à deux étages que dans des cas particuliers, par exemple quand on veut obtenir une grande hauteur d'élévation.

Le principal avantage d'une pompe axiale est qu'elle permet de véhiculer un débit relativement élevé à une hauteur relativement faible. Elle peut pomper jusqu'à 3 fois plus d'eau et d'autres fluides à des hauteurs de moins de 4 mètres par rapport à la pompe à écoulement radial ou centrifuge la plus courante. Il peut également être facilement ajusté pour fonctionner avec une efficacité maximale à faible débit / haute pression et haut débit / basse pression en modifiant l'inclinaison de l'hélice en cours de mouvement (pour certains modèles uniquement).

Dans cette étude, il s'agit d'exploiter les résultats que donnent le logiciel de dimensionnement des pompes axiales « PompAx » développé par G. MEBARKI et qui se présentent sous forme de 16 fichiers donnant chacun des profils (NACA) constituant le rotor et le stator. Aussi, un second fichier donnant la géométrie de ces profils comprenant le rayon, l'angle de calage et la corde.

L’objectif principal de cette étude est d’automatiser l’exploitation de ces fichiers depuis la sortie de « PompAx » jusqu’à la conception finale de la pompe 3D afin de la préparer à la simulation numérique avec un logiciel de CFD. L’outil qui sera utilisé dans ce travail est le logiciel de conception 3D paramétrique « FreeCad » qui est libre d’utilisation et qui utilise le langage « Python » pour ses macro-commandes.

Chapitre 01

Généralités sur les pompes axiales

1.1. Introduction

Dans ce chapitre, nous allons présenter, d'une manière très brève, quelques définitions et critères caractérisant les pompes axiales, leur fonctionnement et leurs caractéristiques.

1.2. Pompes axiales

Les pompes axiales ou hélices sont des machines dont l'organe de travail est un propulseur (roue hélice) à pales en forme d'ailes portantes qui constituent une grille cylindrique à forte transmittance (rapport *pas/corde*). Etant donné que les trajectoires des particules liquides dans la roue se trouvent sur des surfaces cylindriques concentriques avec son axe, les pompes hélices sont qualifiées de pompes à écoulement axial et de ce fait appelées «*pompes axiales*».

L'énergie requise pour faire fonctionner ces machines dépend donc de nombreux facteurs rencontrés dans l'étude des écoulements dont:

a) les propriétés du fluide : masse volumique, viscosité, compressibilité ;

b) les caractéristiques de l'installation: longueur, diamètre, rugosité, singularités ... etc ;

c) les caractéristiques de l'écoulement: vitesse, débit, hauteur d'élévation, pression … etc.

1.3. Critères généraux de définition des pompes

Les critères les plus importants des pompes sont:

1.3.1. Vitesse de rotation

C'est le nombre de tours qu'effectue la pompe par unité de temps [1].

$$\omega = \frac{2 \cdot \pi \cdot N}{60} \qquad (1.1)$$

ω : vitesse angulaire de rotation (rd/s).

N : vitesse de rotation (tr/mn).

1.3.2. Débit d'une pompe

Le débit refoulé Q est le volume utile débité au refoulement de la pompe par unité de temps, en m³/s (unités également utilisées : l/s et m³/h). Il est proportionnel à la vitesse de rotation de la pompe. Le débit de fuite ainsi que les écoulements dans les jeux ne sont pas compris dans le débit refoulé [1].

1.3.3. Puissance

La puissance disponible au niveau de l'arbre d'entraînement de la roue de la pompe est la puissance absorbée par cette pompe. Cette puissance est exactement la puissance utile du moteur d'entraînement de la pompe [1].

La puissance transmise au fluide issue de la part de la pompe est appelée puissance hydraulique utile (P_u).

$$P_u = \rho \cdot g \cdot H \cdot Q_v \qquad (1.2)$$

P_u : puissance absorbée (W).

ρ: masse volumique du liquide pompé (kg/m^3).

g: accélération gravitaire (m/s^2).

Q: débit volumique fourni par la pompe (m^3/s).

H: hauteur produite par la pompe (m).

1.3.4. Rendement

Il est représenté par le rapport de la puissance utile à la puissance absorbée :

$$\eta = \frac{P_u}{P_a} \qquad (1.3)$$

1.3.5. Courbes caractéristiques des pompes

Les courbes principales qui caractérisent une pompe sont au nombre de trois. Elles sont établies par le constructeur [1]:

- courbe débit-hauteur
- courbe de rendement
- courbe de puissance

1.4. Principe de fonctionnement

Les pompes axiales sont des machines qui réalisent l'écoulement d'un fluide dans un réseau en utilisant une certaine quantité d'énergie fournie par un moteur. Les caractéristiques de débit et de pression de la pompe déterminent une courbe de fonctionnement de la pompe à comparer avec les pertes de charges et le dénivelé du réseau, la comparaison de la courbe de la pompe et du réseau permet de trouver un point de fonctionnement du système pompe réseau. Si le fluide est un liquide, il faudra éviter la cavitation du conduit d'aspiration ou de la pompe[2].

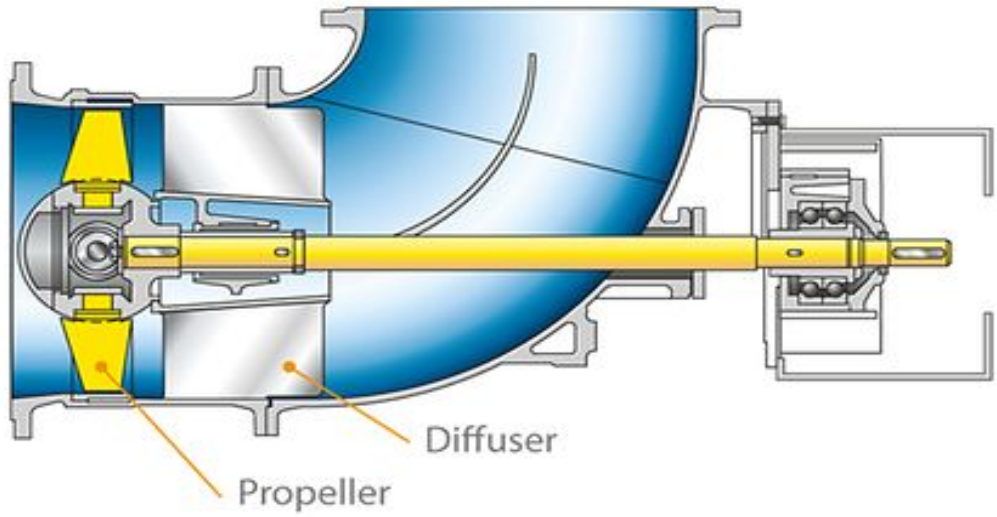


Figure 1.1: Pompe axiale.

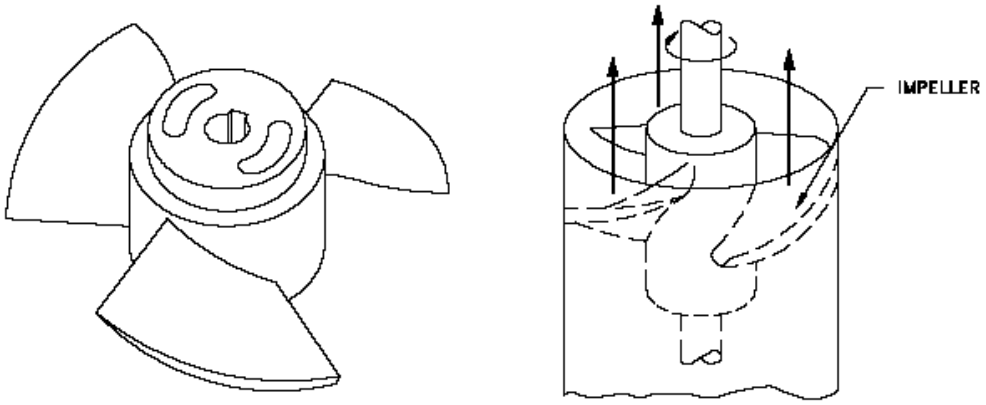


Figure 1.2: Rotor d'une pompe axiale avec aubes (impeller).

On peut décomposer le fonctionnement du pompe axiale en deux étapes:

- **Aspiration**

Le liquide est aspiré au centre du rotor par une ouverture appelée distributeur dont le rôle est de conduire le fluide depuis la conduite d'aspiration jusqu'à la section d'entrée du rotor. La pompe étant amorcée, c'est à dire pleine de liquide, la vitesse du fluide qui entre dans la roue augmente et par conséquent la pression dans l'ouïe diminue et engendre ainsi une aspiration et maintient l'amorçage [1].

- **Accélération**

Le rotor transforme l'énergie mécanique appliquée à l'arbre de la machine en énergie cinétique. A la sortie du rotor, le fluide se trouve projeté dans la volute dont le but est de collecter le fluide et de le ramener dans la section de sortie. La section offerte au liquide étant de plus en plus grande, son énergie cinétique se transforme en énergie de pression. La puissance hydraulique fournie par la pompe est donnée par la relation (1.2) dans laquelle la hauteur manométrique est la hauteur d'une colonne de liquide qui déterminerait une pression statique égale à la pression de refoulement et le terme Q_v H est souvent appelé *charge hydraulique [1]*.

La puissance mécanique à fournir à la machine est bien évidemment toujours supérieure à la puissance hydraulique fournie au liquide et on appelle rendement de la pompe le coefficient de proportionnalité qui lie ces deux paramètres[2].

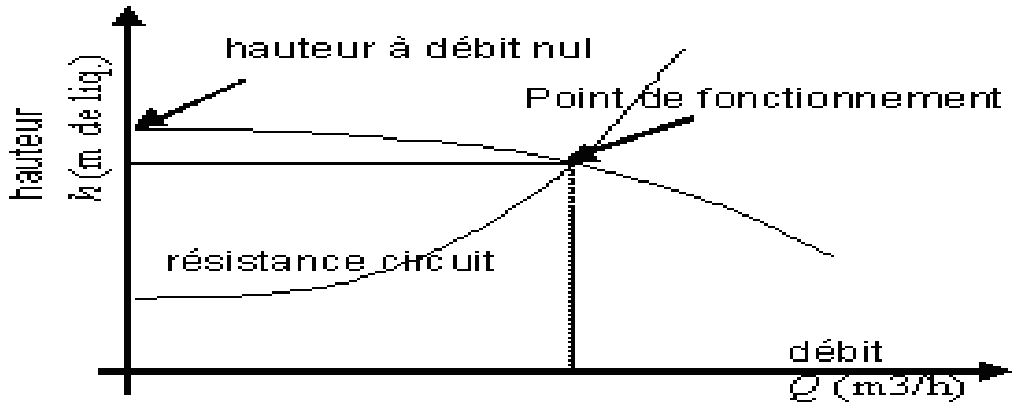


Figure 1.3: Point de fonctionnement.

Le rendement varie en fonction du point de fonctionnement et dépend également de la machine. Pour les machines usuelles, les catalogues de constructeurs indiquent qu'il se situe le plus souvent entre 70 % et 90 %.

Une pompe axiale ne délivre ni une quantité de liquide fixée, ni une pression déterminée : le point de fonctionnement est déterminé par la résistance du circuit connecté à la pompe. Elle augmente simultanément ces deux paramètres, en sorte que le débit obtenu dépend de la pression selon une certaine relation qui définit dans un graphique débit – pression une courbe qu'on appelle « courbe caractéristique de la pompe ».

Cette courbe caractéristique est le plus souvent décroissante : la pression diminue quand le débit augmente et affecte une forme grossièrement parabolique[2].

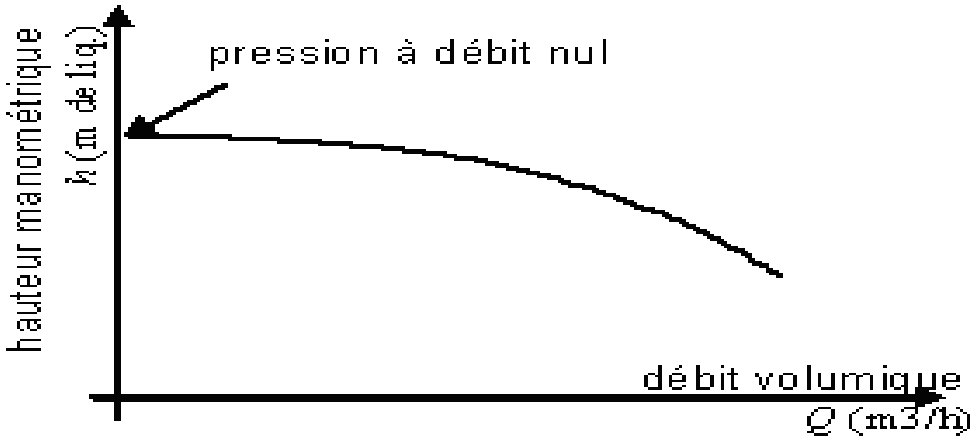


Figure 1.4: Courbe caractéristique.

En fonction des caractéristiques du circuit hydraulique de refoulement, les propriétés du liquide pompé vont varier tout en restant toujours situées sur cette courbe.

La pression obtenue lorsque la pompe fonctionne à débit nul est la pression maximale à laquelle le circuit aval puisse être soumis et constitue un paramètre de dimensionnement très important pour toute l'installation.

La courbe caractéristique d'une pompe dépend, pour un corps de pompe donné, de la dimension du diamètre extérieur de la roue. Les fournisseurs proposent en général des abaques définis dans le plan (Q_v, H), qui présentent les diverses courbes obtenues pour des diamètres variables de l'aube, ainsi que le rendement de la machine en ces points [1].

1.5. Principe et conception d'une hélice (aube)

1.5.1. Définition du l'aube

C'est un objet antisymétrique (symétrique par rapport à un point) qui, en rotation, décrit un disque rotor. La forme aérodynamique de ses pales produit alors entre les deux faces du disque une différence de pression perpendiculaire au disque et centrée sur son axe. Une pale d'hélice possède sa polaire = f (R_x, R_z) en rotation, comme pour une aile, on définit

sa trajectoire, son incidence, sa sustentation, sa traînée, sa charge par m^2, sa charge par cheval, sa corde, etc [1].

1.4.2. Critères généraux pour la construction d'une aube (hélice)

Le profil isolé est un obstacle profilé possédant une faible traînée (qui reste faible à condition que l'angle d'incidence ne soit pas trop élevé) mais une portance notable. La plus part de ces profils sont obtenus à partir d'un profil de base symétrique dont la ligne moyenne (ou squelette) est courbée d'une manière convenable[7].

Les principales définitions géométriques d'un profil sont présentées sur la figure ci-dessous :

- **Bord d'attaque :** point A situé sur l'arête d'entrée du profil.
- **Bord de fuite :** point B situé sur l'arête de sortie du profil.
- **Corde du profil (*l*):** ligne imaginaire joignant le bord d'attaque au bord de fuite et servant au repérage du profil par rapport à l'écoulement.
- **Ligne moyenne (squelette):** ligne courbée donnant la forme du profil.
- **Epaisseur du profil (e_{max}):** distance maximale entre l'extrados et l'intrados, mesurée perpendiculairement à la corde.
- **Epaisseur relative (e_{max}/ l):** rapport entre l'épaisseur du profil et la longueur de sa corde e_{max}/ l : 4 à 20% .
- **Flèche du profil (h_{max}):** distance maximale entre la ligne moyenne et la corde, mesurée perpendiculairement à cette dernière.
- **Cambrure géométrique (h_{max}/ l):** rapport entre la flèche du profil et la longueur de sa corde h_{max}/ l: 0 à 20%.
- **Envergure du profil (*b*):** longueur du profil mesurée perpendiculairement à son plan de symétrie.
- **Angle d'incidence (*i*):** angle entre la corde l et la vitesse d'entrée C_1, il permet le repérage du profil par rapport à l'axe de l'écoulement.
- **Allongement (λ):** rapport entre l'envergure et la longueur de la corde du profil $\lambda = b / l$
- **Intrados:** surface convexe du profil.
- **Extrados:** surface concave du profil.

- **Cambrure aérodynamique :** La cambrure aérodynamique présente une signification physique plus importante que celle liée à la relation précédente. $C_{z\infty O}$ peut en effet être relevée expérimentalement, c'est le coefficient de portance en aubage isolé mesuré sous incidence nulle ($i = 0$)

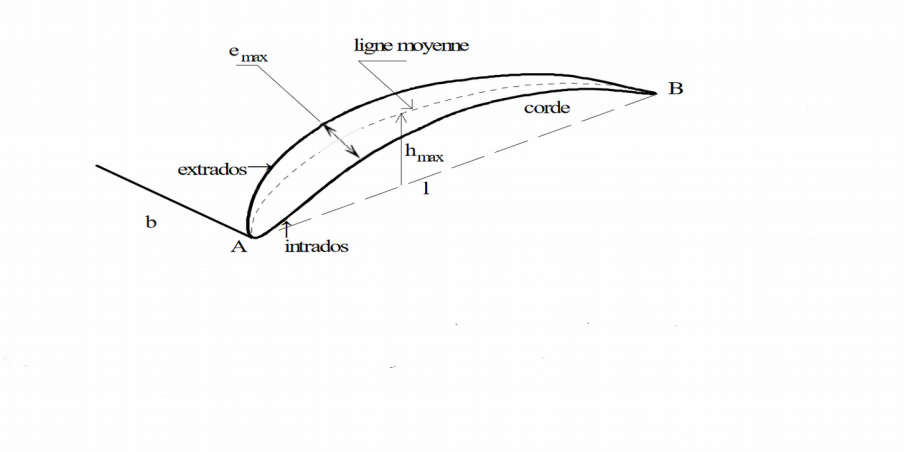


Figure 1.5: Définition géométrique de l'aube.

1.5. Conclusion

Nous avons passé brièvement, dans ce chapitre, quelques définitions concernant surtout les pompes axiales ainsi que les caractéristiques géométriques liées à la construction de l'aube qui est basée sur les profils qui la constituent. Nous avons cité uniquement les paramètres nécessaires à la compréhension du logiciel « PompAx ».

Le chapitre suivant sera consacré au logiciel « FreeCAD » que nous avons utilisé dans notre travail.

Chapitre 02

Généralité sur Freecad

2.1. Introduction

Dans ce chapitre, nous aborderons une présentation la plus complète sans rentrer dans les détails du logiciel que nous avons utilisé afin d'arriver à notre but dans ce projet. Nous allons donc, présenter les différents ateliers composant ce logiciel ainsi que la notion de macro-commande qui nous a permis d'écrire nos propres macros.

2.2. Présentation de *FreeCAD*

« *FreeCAD* » est un modeleur paramétrique 3D de CAO libre et open-source (sous licence LGPLv2) et un logiciel de modélisation d'information du bâtiment (BIM) avec support de la méthode des éléments finis (FEM). FreeCAD vise directement la conception de produits de génie mécanique, mais s'étend également à un plus large éventail d'utilisations autour de l'ingénierie, telles que l'architecture ou l'ingénierie électrique. FreeCAD peut être utilisé de manière interactive, ou sa fonctionnalité peut être accédée et étendue en utilisant le langage de programmation Python et est actuellement en phase de développement.

Le concept principal de l'interface de *FreeCAD* est qu'il est composé d'ateliers. Un atelier est une collection d'outils adaptés pour une tâche spécifique, comme travailler avec des maillages, faire du dessin 2D ou faire des esquisses contraintes. Vous pouvez changer l'atelier actuel avec le sélecteur d'ateliers (6). Vous pouvez personnaliser les outils inclus dans chaque atelier, ajouter des outils provenant d'autres ateliers ou même créer vos propres outils, que nous appelons Marco. Il y a aussi un atelier générique qui rassemble les outils les plus couramment utilisés appelé l'atelier **complete** (complet) [1].

2.3. **Bases de FreeCAD**

« *FreeCAD* » propose des outils similaires à *AutoDesk Revit*, *CATIA*, *CEO*, *AutoDesk Inventor*, *SolideWorks* ou *SolideEdge* et tombe donc dans la catégorie BIM (Building Info Modeling), MCAD (Conception assistée par PLM, CAx et CAE).

Il est destiné à être un modeleur paramétrique basé sur les caractéristiques avec une architecture logicielle modulaire, ce qui permet de fournir facilement des fonctionnalités supplémentaires sans modifier le système principal.

Comme de nombreux modeleurs de CAO 3D modernes, il aura un composant 2D afin d'extraire les détails de conception du modèle 3D pour créer des dessins de production 2D, mais pas le dessin 2D direct (comme *AutoCAD LT*), ni les formes animées ou organiques (comme *Blinder*, *Maya*, *3DS Max* ou *Cinéma 4D*), bien que, grâce à sa grande adaptabilité, *FreeCAD* pourrait devenir utile dans un domaine beaucoup plus large que son objectif actuel. Il est destiné à utiliser d'autres bibliothèques open-source du domaine de l'informatique scientifique. Parmi eux, *Open CASCADE* (un noyau CAD), *Coin3D* (une incarnation d'Open Inventor), le FrameWork *QtGUI* et *Python*, un langage de script populaire.

FreeCAD lui-même peut également être utilisé comme bibliothèque par d'autres programmes [4].

2.4. **Ateliers de FreeCAD**

FreeCAD, comme beaucoup d'applications de conception moderne, est basée sur le concept d'ateliers. Un atelier comporte un ensemble d'outils pour une tâche spécifique. Dans une usine de meubles, on peut trouver un atelier pour l'ébéniste, un autre pour celui qui travaille les métaux et, peut-être, un troisième pour l'assembleur. Le même principe s'applique dans *FreeCAD*. Les outils sont regroupés sous des ateliers, selon les tâches auxquelles ils sont destinés.

Quand vous basculez d'un atelier à un autre, les outils disponibles dans l'interface changent. Les barres d'outils, les barres de commande et sûrement d'autres parties de l'interface basculent vers le nouvel atelier, mais le contenu de votre scène ne change pas. Vous pouvez par exemple commencer à dessiner des formes 2D dans la planche à dessin, puis continuer à travailler sur ces objets dans l'atelier Pièce[4].

Remarquez que parfois, un Atelier est aussi appelé un "Module". Cependant Atelier et Module sont deux choses différentes. Un Module est une extension de *FreeCAD* alors qu'un Atelier est une configuration de l'interface graphique qui regroupe quelques outils et menus. Habituellement chaque Module contient son propre Atelier, c'est pour ça que les deux termes sont utilisés.

Les ateliers que nous avons utilisé dans notre travaille sont:

- Draft - Drawing - Sketcher - Part - Part Design

2.4.1. **L'atelier Draft**

La Planche à dessin (ou atelier Draft) est un module expérimental en cours de développement conçu pour ajouter des fonctionnalités basiques de dessin en deux dimensions à *FreeCAD*. Il est entièrement programmé en *Python*, et a également pour but de démontrer comment accroître les possibilités de *FreeCAD* entièrement en *Python*, sans même toucher au code source [4].

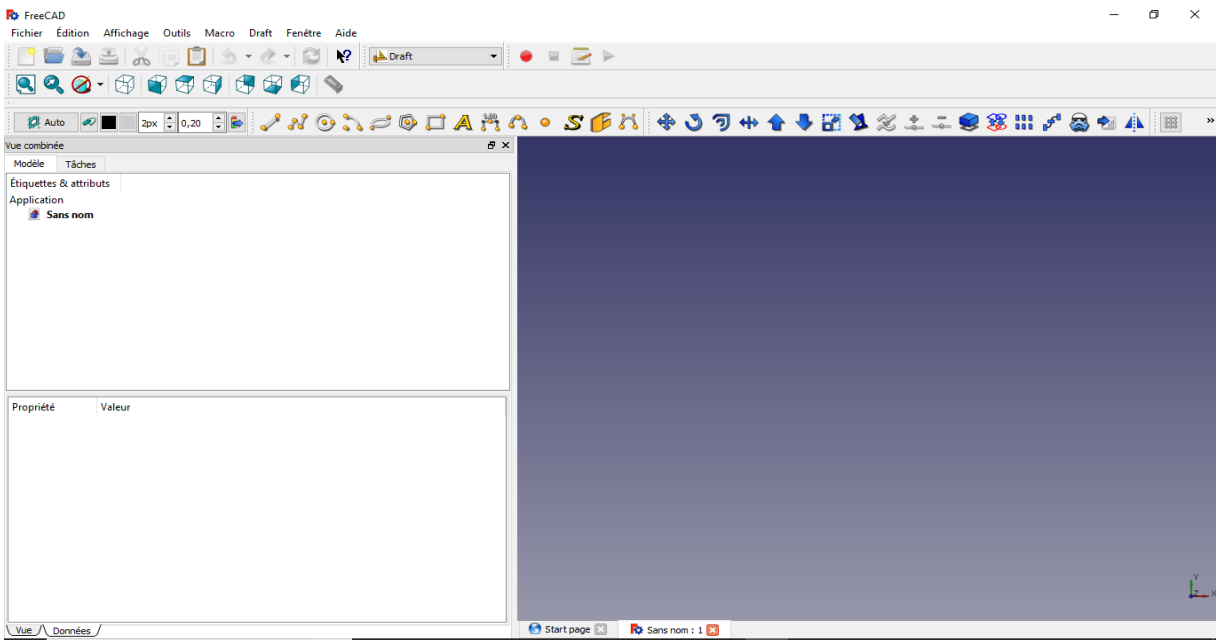


***Figure 2.1*: Atelier Draft.**

L'atelier Draft a son propre atelier de dessin qui place les objets du projet sur papier. Il a quelques fonctionnalités supplémentaires sur les outils de dessin standards et prend en charge les objets spécifiques tels que les dimensions (Annexe 01).

2.4.2. L'atelier Drawing

L'atelier mise en plan vous permet de mettre votre travail 3D sur papier (Annexe 02). C'est-à-dire, mettre des vues de vos modèles dans une fenêtre 2D et insérer cette fenêtre dans un dessin, par exemple une feuille avec une bordure, un titre et votre logo et enfin imprimer cette feuille [4].

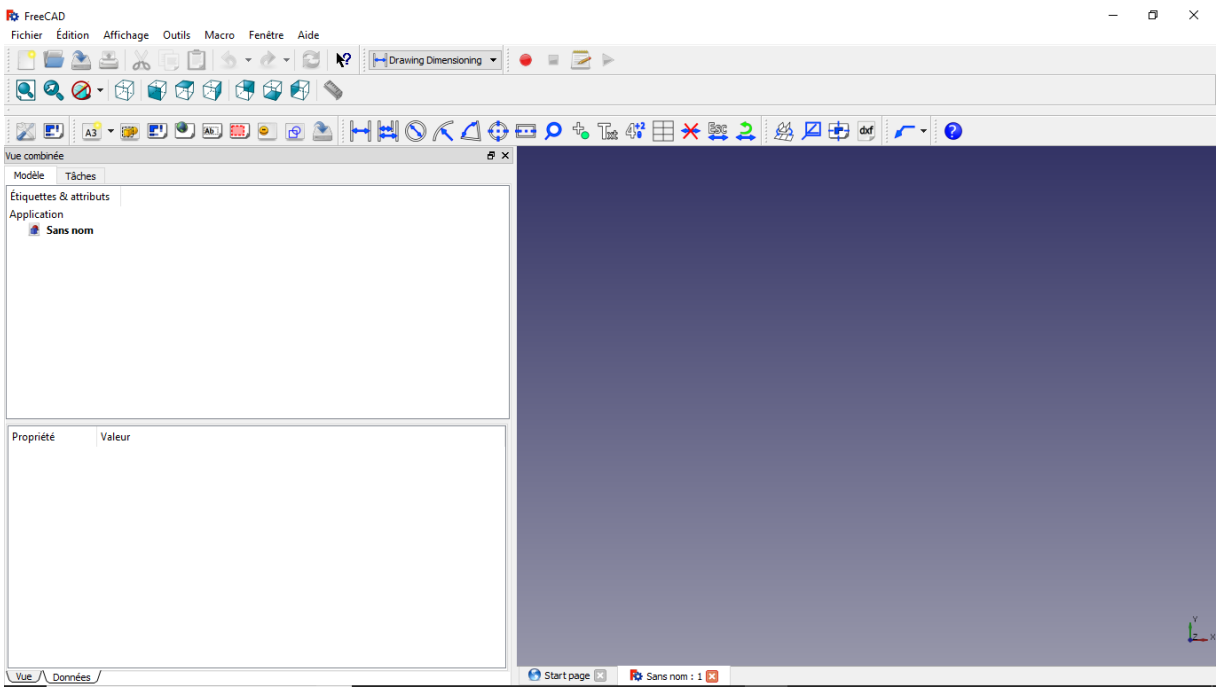


Figure 2.2: Atelier Drawing.

2.4.3. L'atelier Part

Les capacités CAO de *FreeCAD* sont basées sur le noyau *OpenCascade*. L'établissement de pièce de *FreeCAD* permet d'accéder et utiliser les objets et les fonctions *OpenCascade*. Ce dernier est un noyau CAO de niveau professionnel, qui contient des fonctions avancées de manipulation de géométrie 3D et d'objets. Les objets pièces,

contrairement aux objets maillages, sont beaucoup plus complexes, et permettent donc des opérations beaucoup plus avancées, telles que les opérations booléennes logiques, l'historique des modifications ou encore des comportements paramétriques (Annexe 03) [4].

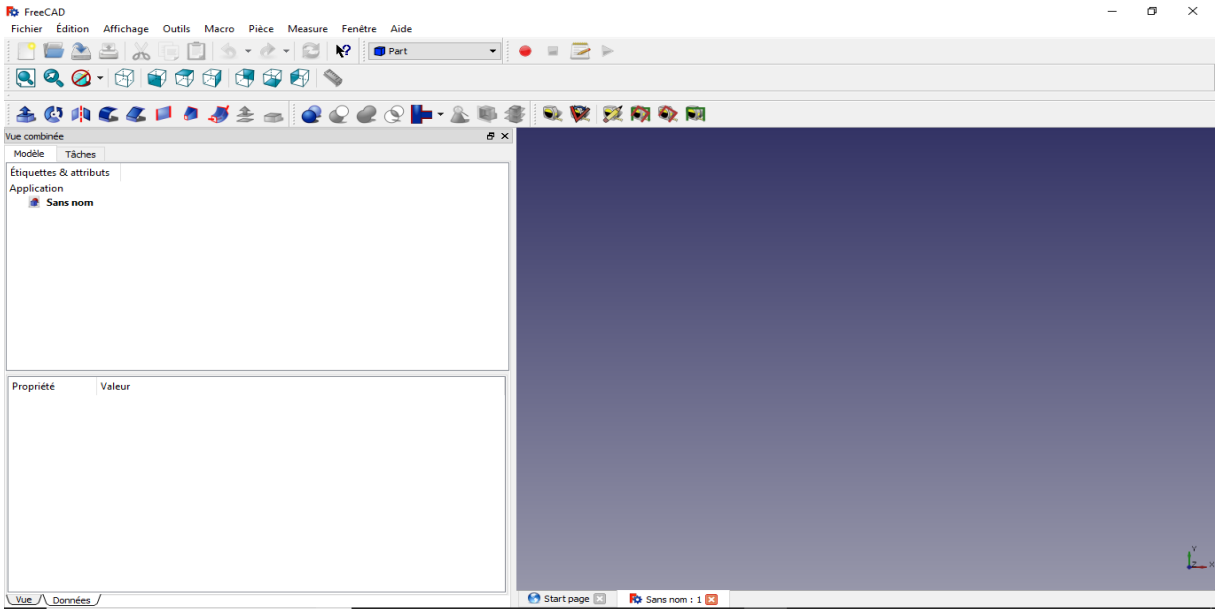


Figure 2.3: Atelier Part.

2.4.4. Atelier PartDesign

L'atelier PartDesign (conception des pièces) fournit des outils avancés pour la modélisation de pièces complexes et solides et est basé sur une méthodologie d'édition de fonctions (Annexe 04). Il est étroitement lié à l'atelier d'esquisse [1].

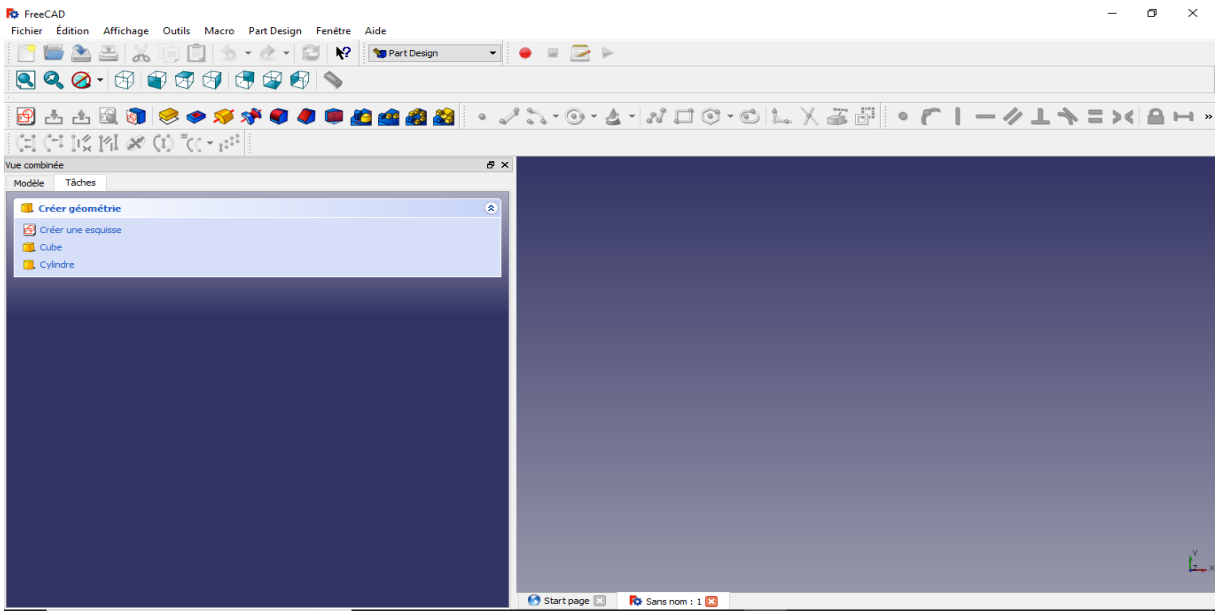


Figure 2.4: Atelier PartDsign.

- **Métrologie d'édition de fonctions**

Alors que l'atelier Part et d'autres ateliers *FreeCAD* construisent des modèles en combinant des formes, l'atelier PartDesign utilise des fonctions. Une fonction est une opération qui modifie la forme d'un modèle.

La première fonction est communément appelée la fonction de base. Au fur et à mesure que de nouvelles fonctions sont ajoutées au modèle, chaque fonction prends la forme de la fonction précédente et ajoute ou enlève de la matière, créant des dépendances directes d'une fonction à la suivante. Cette méthodologie imite un procédé de fabrication typique: un bloc est coupé sur un côté, puis sur un autre côté, des trous sont percés, puis des congés appliqués, etc.

Toutes les fonctions sont listées de façon séquentielle dans l'arborescence du modèle et peuvent être modifiées en tout temps, avec la dernière fonction représentant la pièce finale [1].

Les fonctions peuvent être classées selon différentes catégories:

- **Basée sur profil** : ces fonctions partent d'un profil pour définir la forme de la matière à ajouter ou enlever. Le profil consistera en une esquisse, une face plane de la géométrie existante (un profil sera extrait de ses arêtes), une forme liée ou un objet Draft qui aura préalablement été inclus dans le corps actif.
- **Additive** : ajoute de la matière au modèle existant. Les fonctions additives ont des icônes de couleur jaune.
- **Soustractive** : enlève de la matière du modèle existant. Les fonctions soustractives ont des icônes de couleurs rouge et bleu.
- **Basée sur primitive** : basée sur des primitives géométriques (cube, cylindre, cône, tore...). Elles peuvent être additives ou soustractives.
- **Fonctions de transformation** : elles appliquent une transformation à des fonctions existantes (symétrie, répétition linéaire ou circulaire, transformation multiple).
- **Habillage** : ces fonctions appliquent un traitement à des arêtes ou des faces, tels que des arrondis/congés, chanfreins ou dépouilles.
- **De procédure** : peut être dit de fonctions qui ne sont pas basées sur des profils, comme les fonctions de transformation et d'habillage.

2.4.5. Atelier Sketcher

L'atelier Sketcher permet de créer des géométries 2D nommées esquisses, qui seront principalement utilisées par l'Atelier PartDesign, mais potentiellement par d'autres ateliers tels que l'atelier Architectural. En général, l'esquisse est le point de départ de la plupart des modèles de CAO - une esquisse simple peut être 'extrudée' en une forme 3D, une autre esquisse peut être créée pour creuser une cavité à la surface de cette forme, ou encore générer une extrusion. Avec les opérations booléennes, l'atelier Sketcher est au cœur de la conception 3D solide (Annexe 05) [1].

L'atelier Sketcher met à l'avant les contraintes, qui permettent de définir des formes 2D selon des critères géométriques précis. Un solveur mathématique calcule le niveau de contrainte de l'esquisse et permet l'exploration interactive des degrés de liberté.

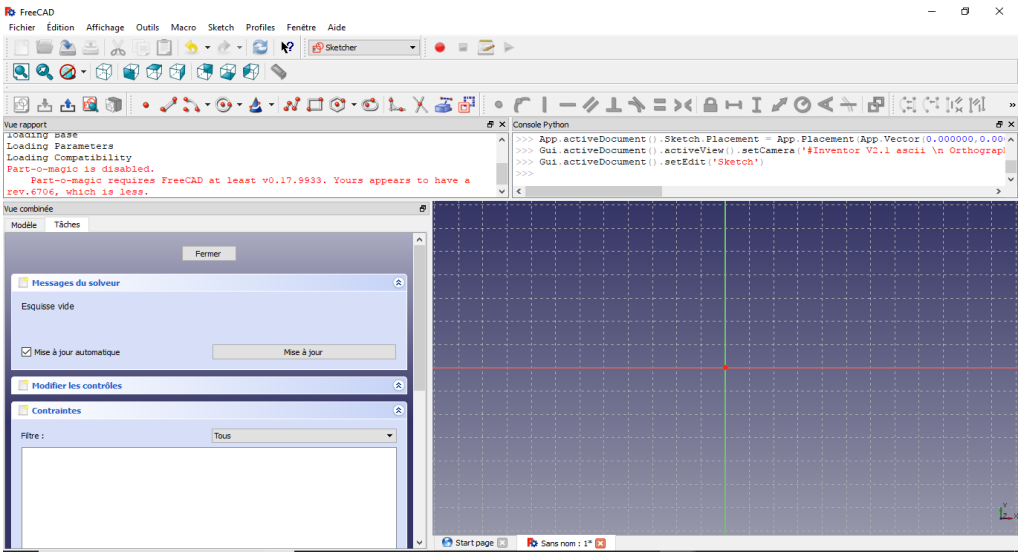


Figure 2.5: Atelier Sketcher.

2.3. Macros

Les macros sont un moyen pratique de créer des actions complexes dans *FreeCAD*. Vous enregistrez simplement les actions au fur et à mesure que vous les faites, puis enregistrez-les sous un nom et relisez-les quand vous le voulez. Puisque les macros sont en réalité une liste de commandes *Python*, vous pouvez aussi les éditer et créer des scripts très complexes. Il est très simple à utiliser: Appuyez sur le bouton d'enregistrement, il vous sera demandé de donner un nom à votre macro, puis effectuez quelques actions. Lorsque vous avez terminé, cliquez sur le bouton « Arrêter » l'enregistrement et vos actions seront sauvegardées. Vous pouvez maintenant accéder à la boîte de dialogue de la macro avec le bouton d'édition [5]:

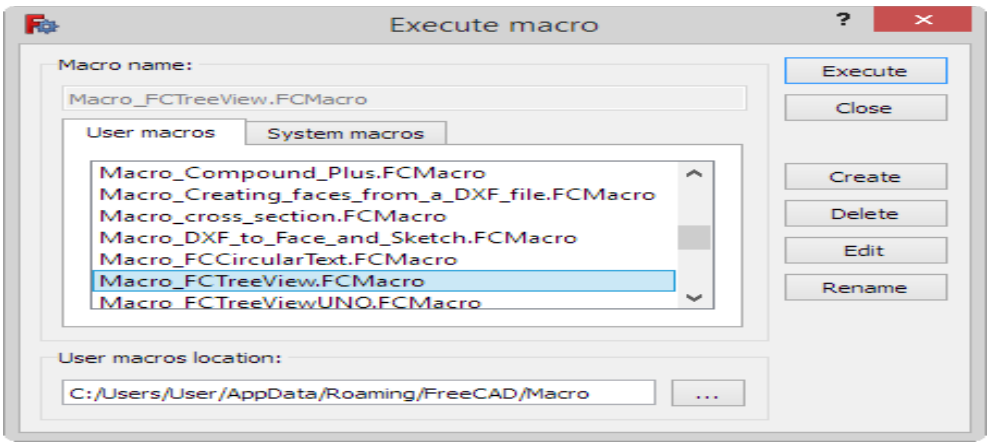


Figure 2.6: Addon manger.

2.5. Langage Python

Habituellement, lors de l'écriture d'un programme informatique, il suffit d'ouvrir votre environnement de programmation préféré qui est, dans la plupart des cas, un éditeur de texte avec plusieurs outils autour de lui, écrire votre programme, puis le compiler et l'exécuter. Lorsque vous avez fait des erreurs pendant l'écriture, votre programme ne fonctionnera pas! et vous obtiendrez un message d'erreur vous indiquant ce qu'il s'est passé. Ensuite, vous revenez à votre éditeur de texte, corrigez les erreurs, exécutez à nouveau, et ainsi de suite jusqu'à ce que votre programme fonctionne parfaitement.

En *Python*, tout ce processus, peut être exécuté de manière transparente dans l'interpréteur *Python*. C'est une fenêtre avec une invite de commande, vous pouvez simplement y taper votre code *Python*. Si vous l'installez sur votre ordinateur , vous aurez cet l'interpréteur dans votre menu de démarrage. Mais *FreeCAD* dispose également d'un interpréteur *Python* intégré, vous n'êtes donc pas obligé de l'installer, cet interpréteur est visible dans la fenêtre inférieure (Si vous ne voyez pas cette fenêtre, cliquez sur **Affichage->Vues->Console Python**). Tous ces exemples ont été relu à partir de l'interpréteur disponible dans *FreeCAD [6]*.

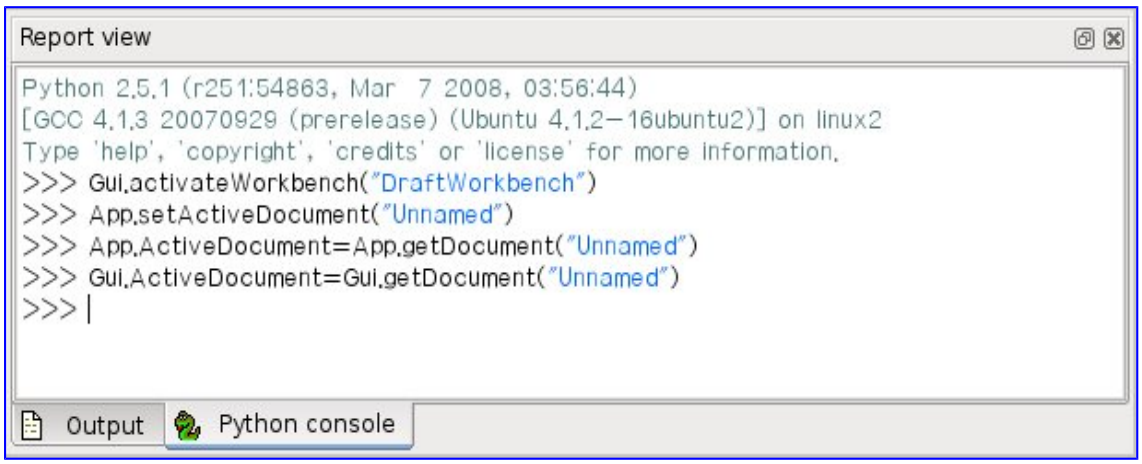


Figure 2.7: Console Python.

(SI vous ne l'avez pas, cliquez sur le menu Affichage --> Panneaux --> Console Python).

L'interpréteur affiche la version de *Python* installée, puis le symbole >>>, qui est l'invite de commande pour entrer votre code *Python*. L'écriture du code dans l'interpréteur est très simple: une ligne, est une instruction. Lorsque vous appuyez sur ENTREE, votre ligne de code est exécuté.

2.6. Conclusion

Le logiciel « *FreeCAD* » vise directement la conception de produits de génie mécanique basée sur le concept d'atelier ,chaque atelier contient des commandes spécialisées, sa fonctionnalité peut être accédée et étendue en utilisant le langage de programmation *Python* et est actuellement en phase de développement.

Après avoir passé en revue les détails qui nous intéresse dans *FreeCAD*, nous allons l'utiliser dans le prochain chapitre afin de modéliser notre pompe.

Chapitre 03

Conception 3D de la pompe axiale

3.1. Introduction

Dans ce chapitre, nous allons mettre en œuvre des procédures d'automatisation afin d'exploiter les fichiers de sortie de « *PompAx* » jusqu'à la conception finale de la pompe 3D afin de la préparer à la simulation numérique.

3.2. Méthodologie

Pour aboutir à la conception de la pompe axiale, il nous faut d'abord récupérer les fichiers des profils des aubes du rotor et du stator et corriger éventuellement les points hors domaine. Ensuite, on effectue une interpolation polynômial afin d'augmenter le nombre de points des profils qui est de 26 sur l'extrados et 26 sur l'intrados. Enfin, nous écrivons des macros afin de concevoir chaque composant de la pompe d'une manière automatique, à savoir :

- Conception de l'aube du rotor
- Conception de l'aube du stator
- Conception de l'entrée du rotor
- Conception du moyeu du rotor et du stator
- Conception de la ceinture
- Conception du rotor complet (aubes + entrée + moyeu)
- Conception du stator complet (aubes + moyeu)
- Conception de la pompe entière.

La clé à tous ce travail est la bonne lecture du fichier de géométrie « *.GEO ». La section suivante nous permettra de comprendre certains détails du logiciel *PompAx*.

3.3. **Logiciel "*PompAx*"**

PompAx est un logiciel réalisé par G. MEBARKI dans son travail de Magister sous la direction de Dr. L. MESSAOUDI en 1997. Il permet le dimensionnement, la prédiction et l'analyse des performances des pompes axiales. Il donne rapidement les dimensions globales de ces dernières seulement à partir du cahier des charges (*H, Q, N*) à la demande de l'utilisateur. Ce logiciel est le fruit d'une étude élaborée sur la base des essais systématiques longs et très coûteux entrepris par le NACA combinés à la théorie de la mécanique des grilles d'aubes établie par COMOLET. La méthode de calcul est mise au point sur quatre bases: l'utilisation des profils NACA 65, le rendement est optimal, les profils conservent la même cambrure du moyeu à la périphérie et ils sont vrillés afin de respecter en tout rayon un angle d'incidence égal à l'angle optimal défini par les essais du NACA [7].

PompAx a été écrit en premier lieu sous le système d'exploitation DOS et contient plus de 7500 lignes de code en TurboPascal. Ensuite, il a été porté sous Windows par deux ingénieurs informaticiens puis optimisé en utilisant la programmation orientée objet en 5300 lignes de code en langage Pascal Object.

Les fichiers relatifs aux profils des aubes obtenus par *PompAx* sont des fichier data qui ne se prètent pas aux tracés. De ce fait, la procédure de sortie de ces fichiers a été modifiée par Dr. L. MESSAOUDI en introduisant le codage spécifique afin d'avoir à la sortie uniquement des fichiers DXF. Ces derniers sont directement exploitable par n'importe quel logiciel de dessin (CAD).

Ci-dessous, quelques captures d'écrans du logiciel *PompAx* ainsi que du fichier de sortie relatif à la géométrie des aubes du rotor et du sator.

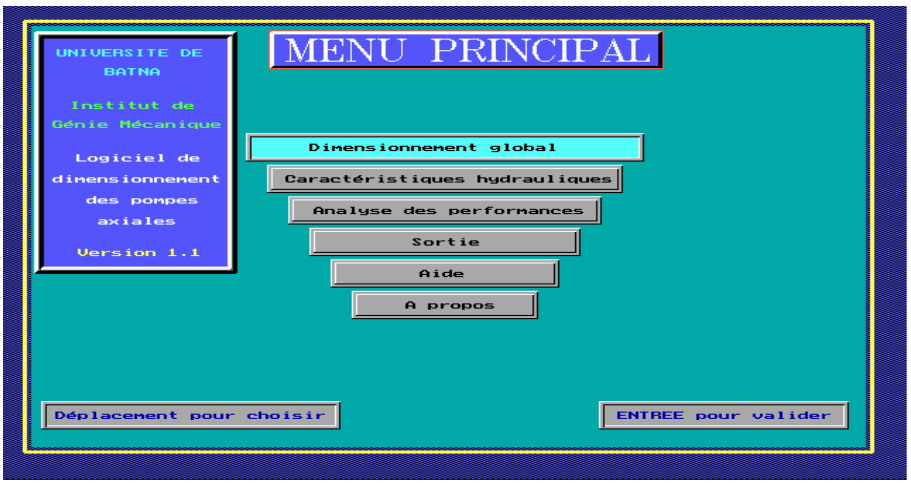
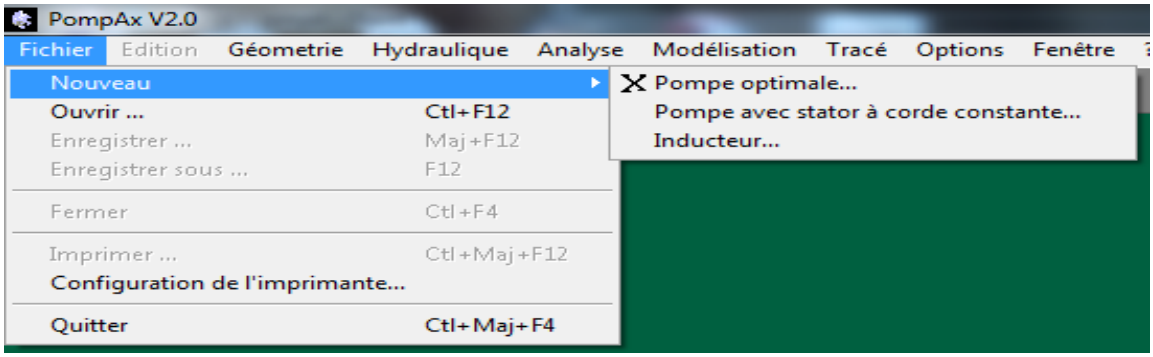


Figure 3.1: Ecran d'accueil du logiciel PompAx sous DOS.



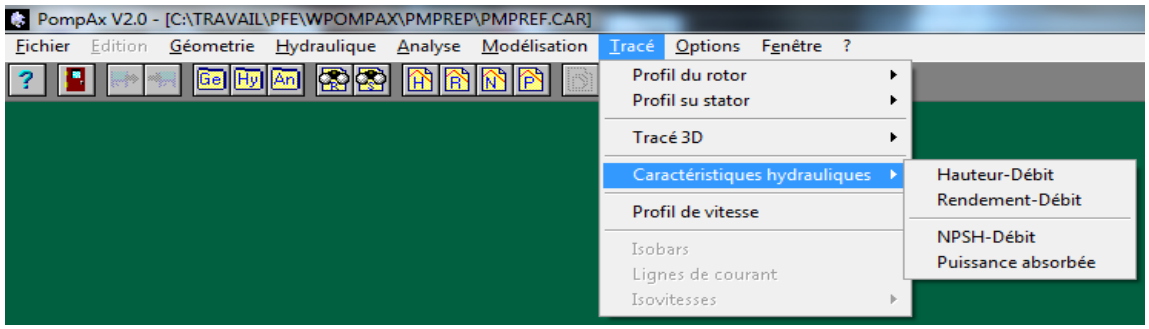


Figure 3.2: Logiciel PompAx sous Windows.

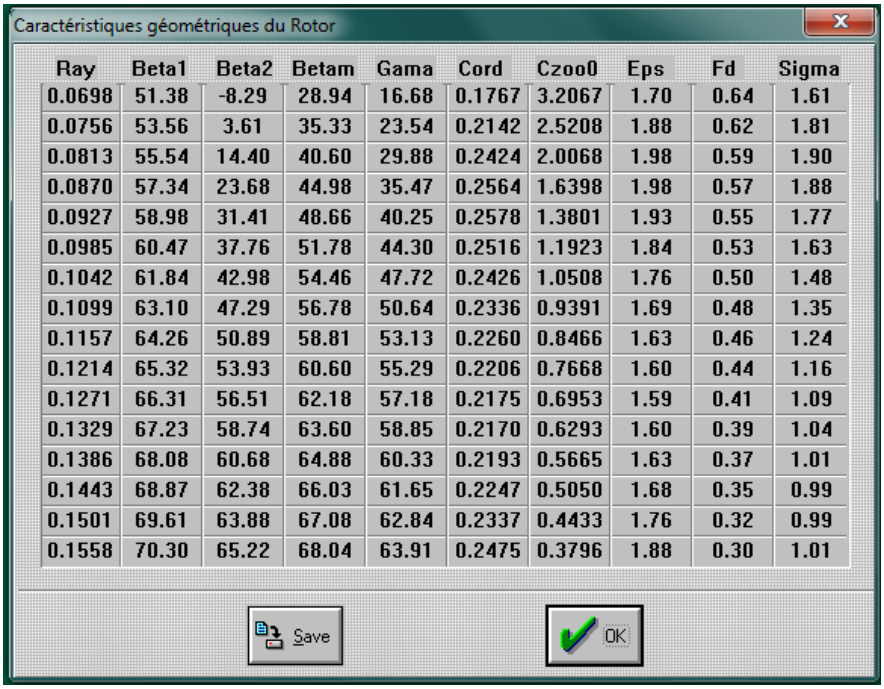


Figure 3.3: Caractéristiques géométriques du rotor – pompe PN2.

Plusieurs pompes ont été étudiées et validées par G. MEBARKI. Nous avons choisi une seule afin de mener à bien notre travail, c'est la pompe N2 dont les caractéristiques sont:

	Valeurs théoriques « POMPAX »				Valeurs expérimentales			
Pompes	Débit	Rend.	Hauteur	Puissance	Débit	Rend.	Hauteur	Puissance
N1	**0.104**	0.863	**2.99**	6.77	**0.173**	0.853	**3.20**	6.37
N2	0.085	0.843	1.605	1.513	0.083	0.85	1.76	2.3
N3	**0.098**	0.848	2.87	6.22	**0.165**	0.855	3.20	7.1
N6	0.164	**0.85**	3.73	5.94	0.14	**0.83**	4.32	7.00
N7	0.170	**0.845**	4.41	9.05	0.175	**0.837**	5.2	10.43

- Cavitation:

Pompes	NPSH calculé « POMPAX »	NPSH mesuré	Facteur de diffusion au rayon extérieur
N1	3.39	3.36	0.24
N2	1.286	1.96	0.166
N3	3.28	3.53	0.24
N6	2.89	2.36	0.18
N7	2.128	-	0.22

Figure 3.4: Pompes étudiées par G. MEBARKI.

Le fichier de sortie délivré par *PompAx* est un fichier texte de la forme :

```
15 1 0.0880 1.8090 1000  1.0000000000E-06 1.00 1.00 18.00 64.00 5.00 1000.0  4  5 9.81 0.600  0.198 0.580  0.400 1 1
0.0531  58.01  30.90  47.72  39.76  0.1018  1.7407  1.46  0.60  1.22
0.0565  59.60  37.32  50.97  43.85  0.1009  1.5089  1.41  0.57  1.14
0.0599  61.05  42.60  53.74  47.32  0.0992  1.3277  1.36  0.55  1.05
0.0633  62.38  46.95  56.15  50.27  0.0974  1.1811  1.33  0.52  0.98
0.0667  63.59  50.59  58.24  52.80  0.0962  1.0585  1.31  0.49  0.92
0.0702  64.72  53.65  60.09  54.99  0.0957  0.9526  1.30  0.47  0.87
0.0736  65.75  56.26  61.72  56.90  0.0961  0.8586  1.31  0.44  0.83
0.0770  66.71  58.51  63.18  58.59  0.0976  0.7727  1.33  0.41  0.81
0.0804  67.61  60.47  64.49  60.09  0.1004  0.6921  1.38  0.39  0.79
0.0838  68.43  62.18  65.68  61.43  0.1048  0.6141  1.45  0.36  0.80
0.0872  69.21  63.70  66.76  62.63  0.1080  0.5364  1.54  0.33  0.81
0.0907  69.93  65.05  67.74  63.72  0.1120  0.4565  1.68  0.31  0.85
0.0941  70.60  66.26  68.64  64.72  0.1160  0.3714  1.87  0.28  0.91
0.0975  71.23  67.36  69.47  65.63  0.1200  0.2776  2.14  0.25  1.01
0.1009  71.83  68.35  70.23  66.46  0.1240  0.1705  2.56  0.22  1.17
0.1043  72.38  69.25  70.94  67.23  0.1280  0.0436  3.28  0.20  1.48
0.0531  45.08  26.63  19.35  0.0825  2.6408  1.32  0.58  1.24
0.0565  43.29  25.22  18.72  0.0822  2.6046  1.24  0.57  1.16
0.0599  41.61  23.95  18.08  0.0823  2.5645  1.18  0.56  1.09
0.0633  40.04  22.79  17.45  0.0827  2.5214  1.13  0.54  1.04
0.0667  38.56  21.73  16.84  0.0833  2.4760  1.09  0.53  0.99
0.0702  37.17  20.77  16.24  0.0841  2.4289  1.06  0.52  0.95
0.0736  35.87  19.88  15.67  0.0851  2.3805  1.03  0.51  0.92
```

0.0770	34.64	19.06	15.12	0.0863	2.3311	1.01	0.50	0.89
0.0804	33.49	18.30	14.59	0.0877	2.2810	1.00	0.48	0.87
0.0838	32.40	17.60	14.08	0.0892	2.2305	0.98	0.47	0.85
0.0872	31.37	16.95	13.59	0.0909	2.1796	0.97	0.46	0.83
0.0907	30.40	16.35	13.13	0.0929	2.1285	0.97	0.45	0.81
0.0941	29.48	15.79	12.68	0.0949	2.0772	0.97	0.44	0.80
0.0975	28.62	15.26	12.26	0.0972	2.0259	0.96	0.42	0.79
0.1009	27.79	14.76	11.86	0.0997	1.9745	0.97	0.41	0.79
0.1043	27.01	14.30	11.47	0.1024	1.9230	0.97	0.40	0.78

***Figure 3.5*: Fichier PN2.GEO de la pompe N2.**

La correspondance avec les couleurs est :

- Rouge : Rayons des différents profils.

- Bleu : Calage de chaque profif.

- Vert : Corde de chaque profil.

Les 16 premières valeurs correspondent au rotor et les 16 dernières au stator.

3.4. Optimisation du tracé 3D

Par optimisation, nous voulons dire que notre objectif est de minimiser l'intervention de l'utilisateur. Pour cela nous avons utillisé le logiciel « *FreeCAD* » ainsi que le langage «*Python* » pour les macro-commandes.

Ci-dessous, un exemple de programme extrait de la console *Python* intégrée à *FreeCAD*.

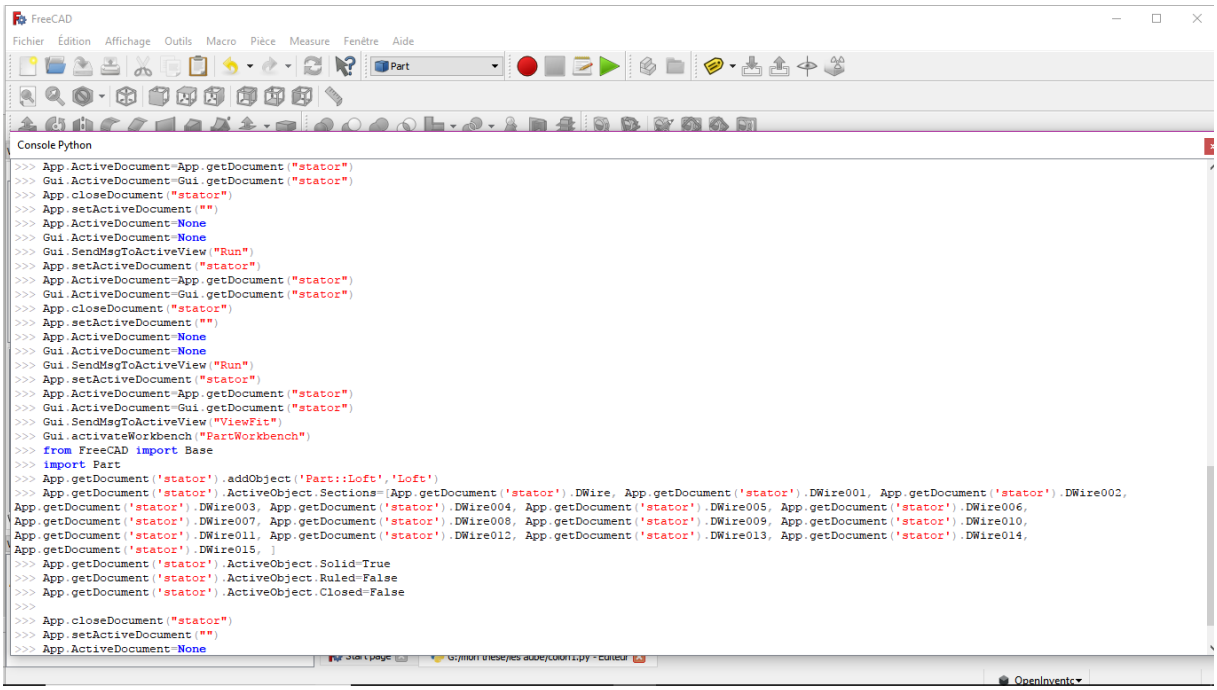


Figure 3.6: Console Python dans FreeCAD.

3.5. Composants de la pompe

La figure ci-dessous nous montre les différents composants de la pompe axiale.

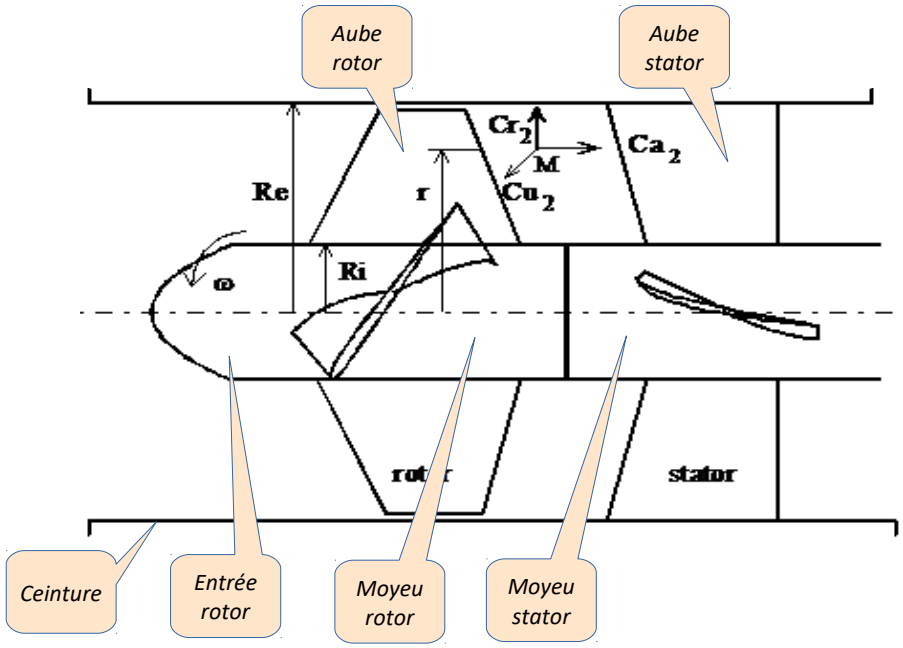


Figure 3.7: Pompe axiale avec ses composants.

3.6. Conception de l'aube du rotor

Les étapes suivis pour résoudre ce problème sont:

- Récupération des 16 fichiers DXF : COUPER1, COUPER2,…COUPER16.
- Récupération des valeurs : rayon, angle de calage et corde à partir de « PN2.GEO ».

Le premier programme que nous avons établi est donné en annexe 06. Les résultats sont :

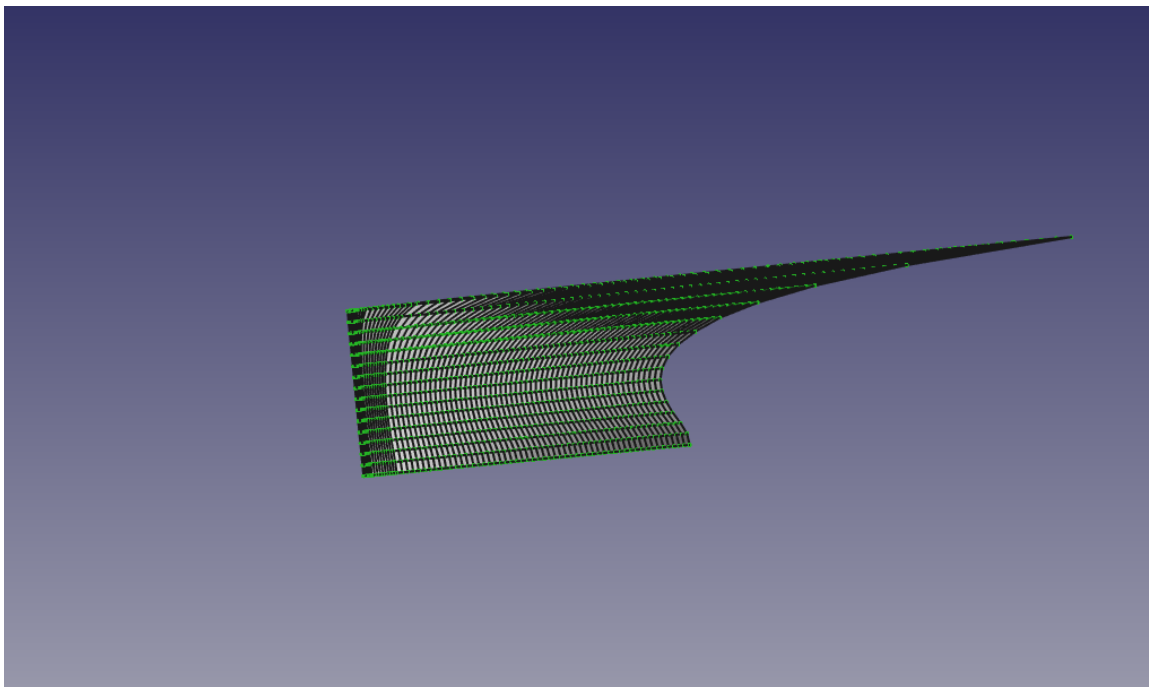


Figure 3.8: Forme de l'aube avec les valeurs initiales «PN2.GEO».

Remarques :

- Nous avons été dans l'obligation de changer certaines valeurs de la corde dans le fichier « PN2.GEO » afin d'obtenir une forme acceptable de l'aube.
- Nous avons aussi remarqué une déformation anormale de la forme de l'aube au niveau du rayon 8. De ce fait nous l'avons éliminé cette coupe en utilisant la commande appropriée.

- Au lieu d'utiliser l'interpolation polynômial afin d'augmenter le nombre de points du profil, FreeCAD nous permet de le faire tout simplement en agissant sur les propriétés du profil après importation en *Dwire*.

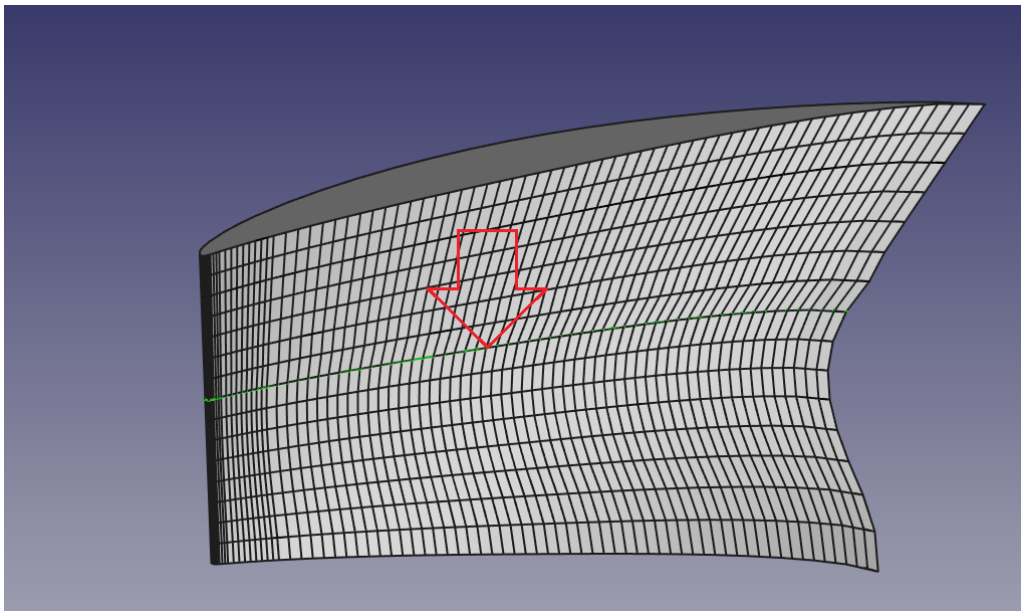


***Figure 3.9**: Aube du rotor avec la coupe 8.*

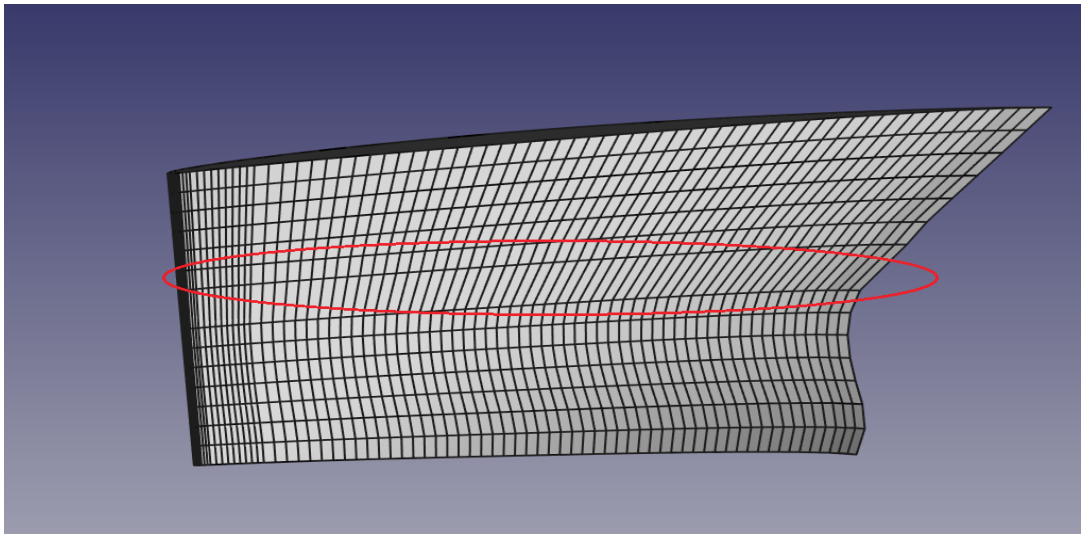


***Figure 3.10** : Aube du rotor sans la coupe 8.*

Problème:

En important le fichier DXF en *polyline,* on ne peut pas changer l'échelle.

Solution:

On importe le fichier DXF en *Dwire,* ce qui permet de changer l'échelle.

Le second programme que nous avons établi est donné en annexe 06. Les résultats sont :

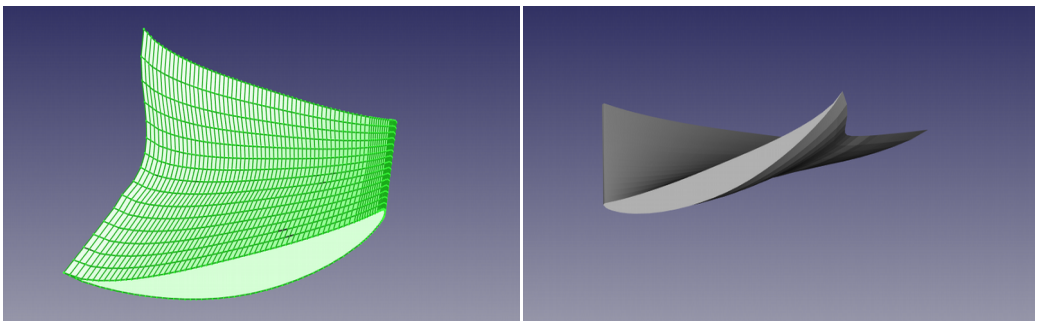


***Figure 3.11**: Forme de l'aube du rotor avec changement d'échelle.*

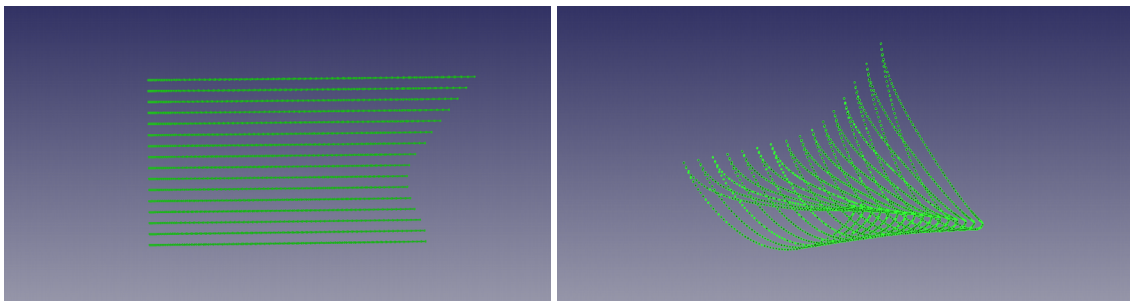


***Figure 3.12**: Coupes de l'aube du rotor.*

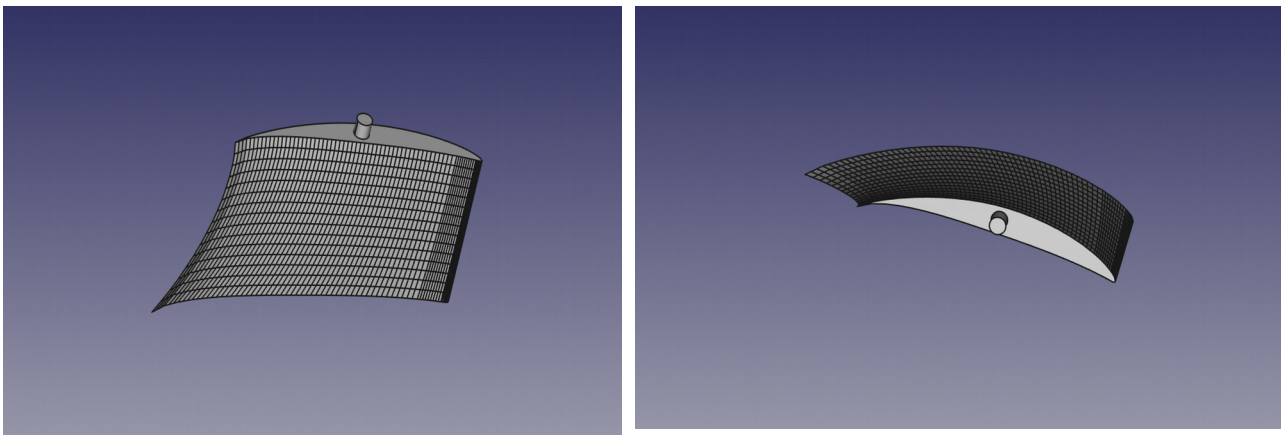


***Figure 3.13**: Conception de l'aube avec la goupille.*

3.7. **Conception de l'aube du stator**

Les étapes suivis pour résoudre ce problème sont:

- Récupération des fichier DXF : COUPES1, COUPES2,…COUPES16.
- Récupération des valeurs : rayon, angle de calage et corde à partir de « PN2.GEO ».

Le programme que nous avons établi est donné en annexe 07 et les résultats sont:

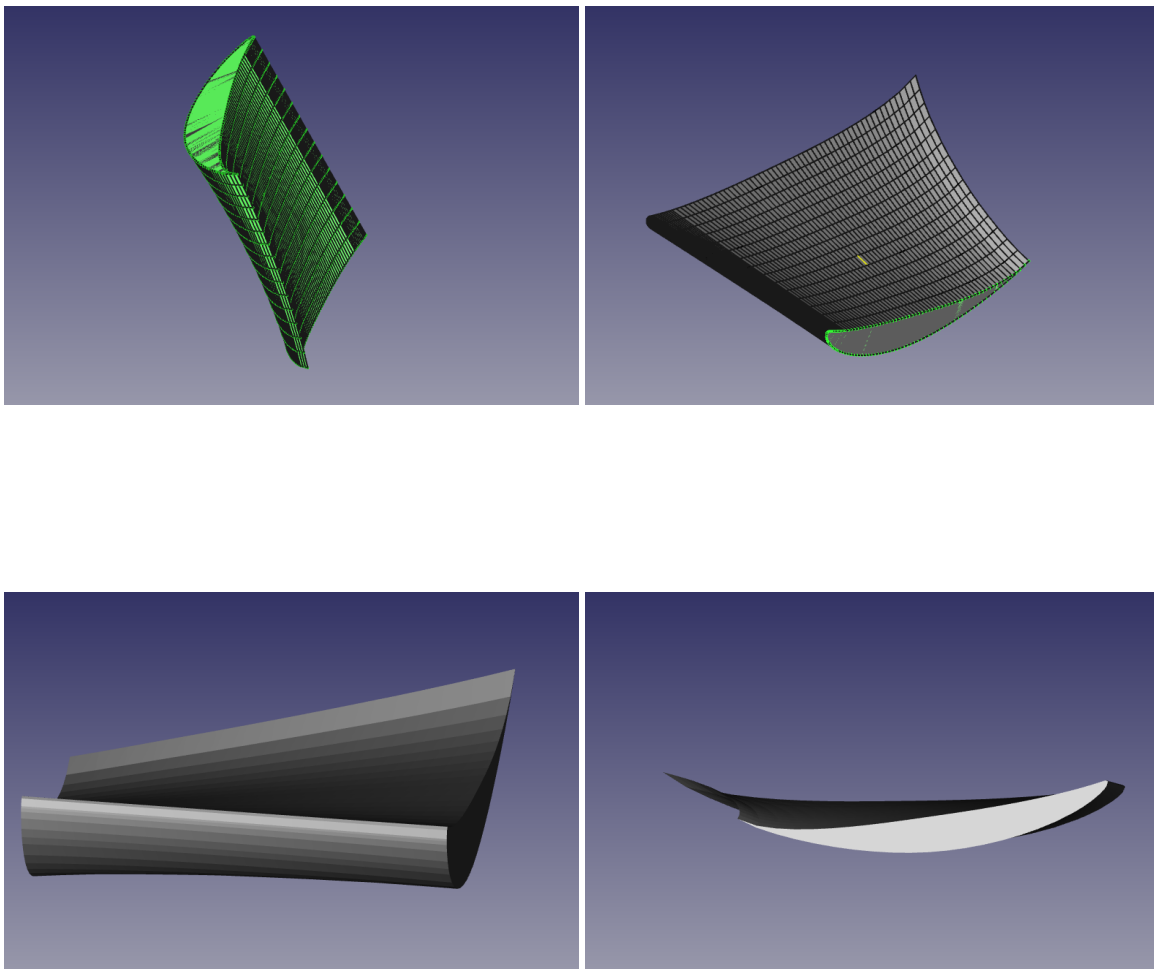


Figure 3.14: Forme de l'aube du stator avec changement de l'échelle.

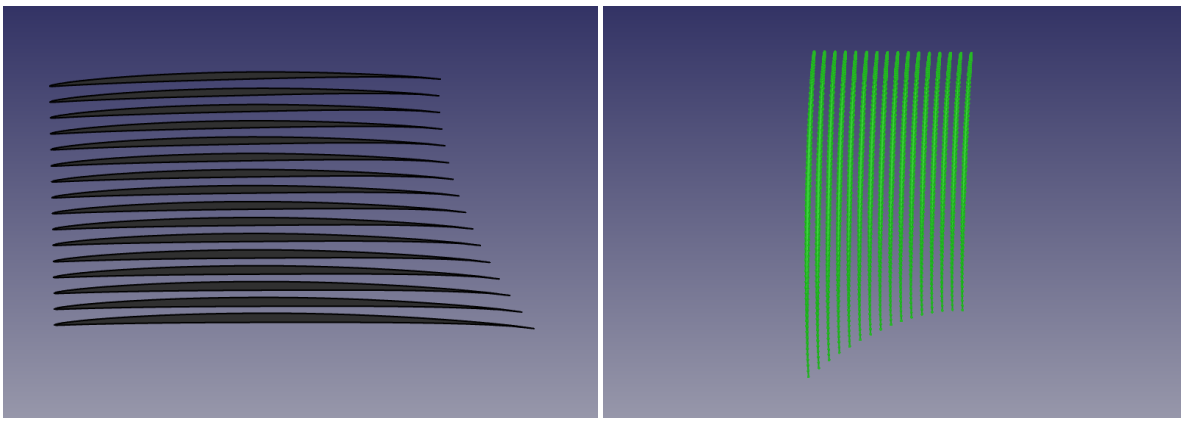


Figure 3.15: Coupes de l'aube du stator.

3.8. Conception de l'entrée du rotor

Le programme que nous avons établi est donné en annexe 08 et les résultats sont:

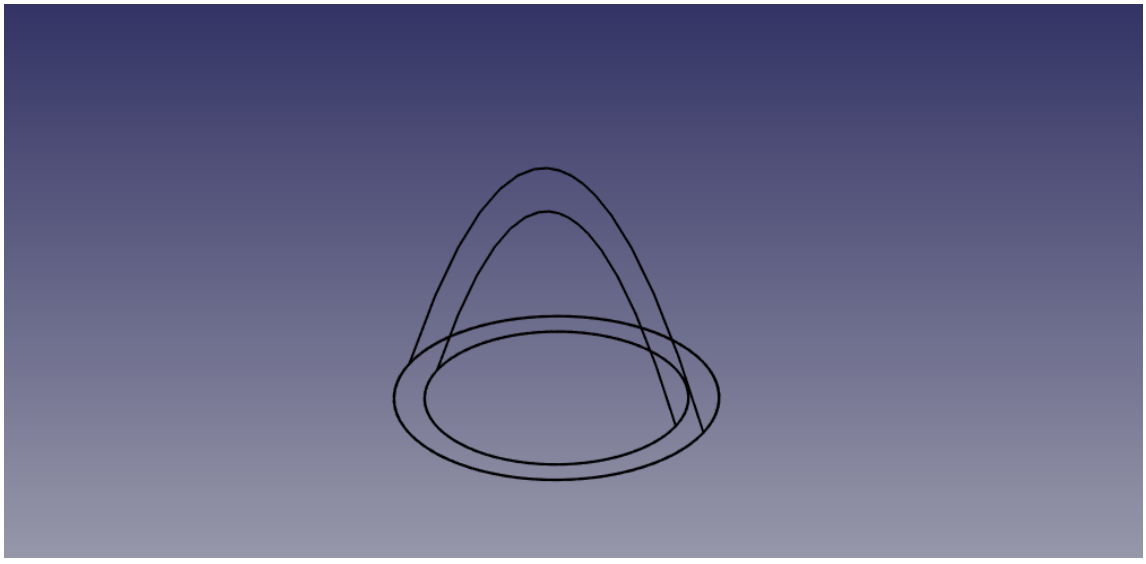


Figure 3.16: Dessin du l'entrée du rotor en 2D.

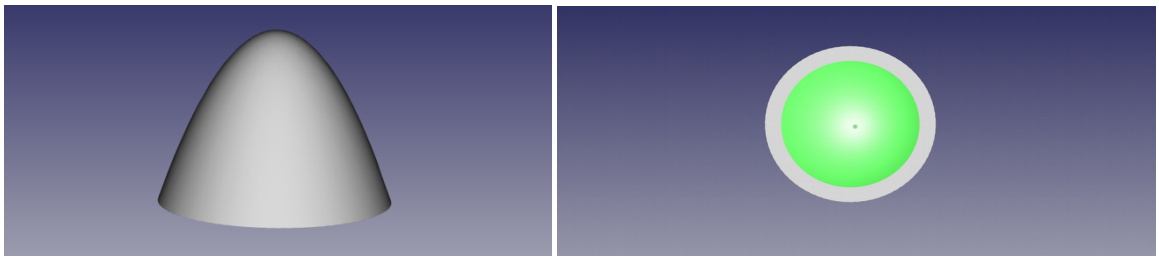


Figure 3.17: Forme de l'entrée du rotor (CF=1,5).

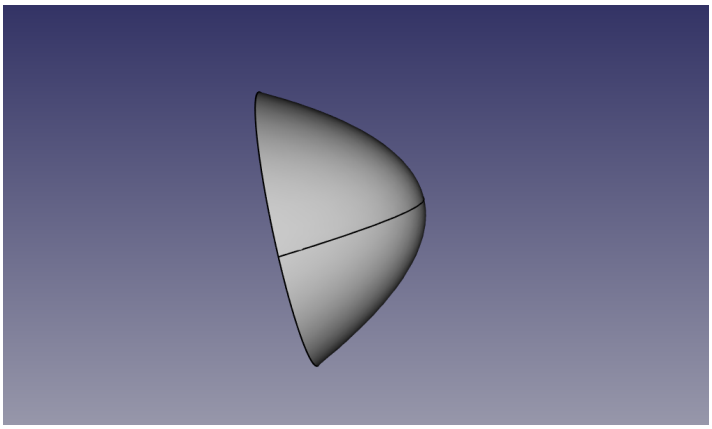


Figure 3.18: Forme du l'entrée du rotor (CF=1).

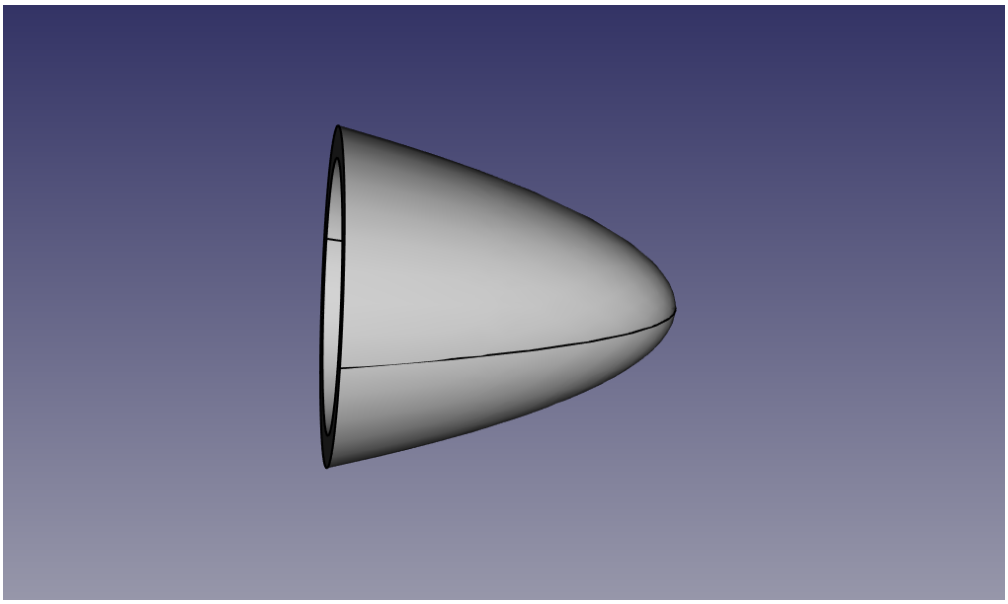


Figure 3.19: Forme du l'entrée du rotor (CF=2).

3.9. Conception du moyeu du rotor et du stator

- Le programme de conception du moyeu du rotor est en annexe 09.
- Le programme de conception du moyeu du stator est en annexe 10.

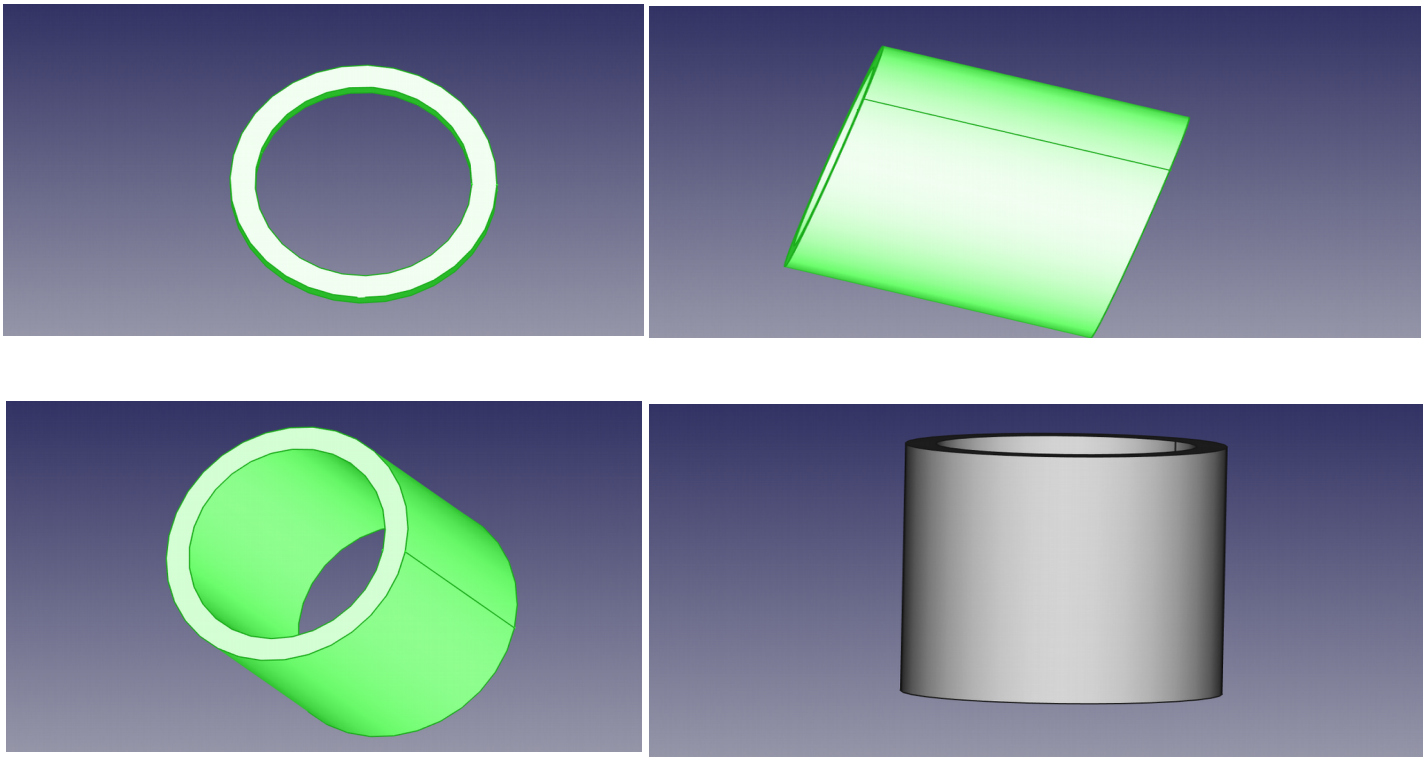
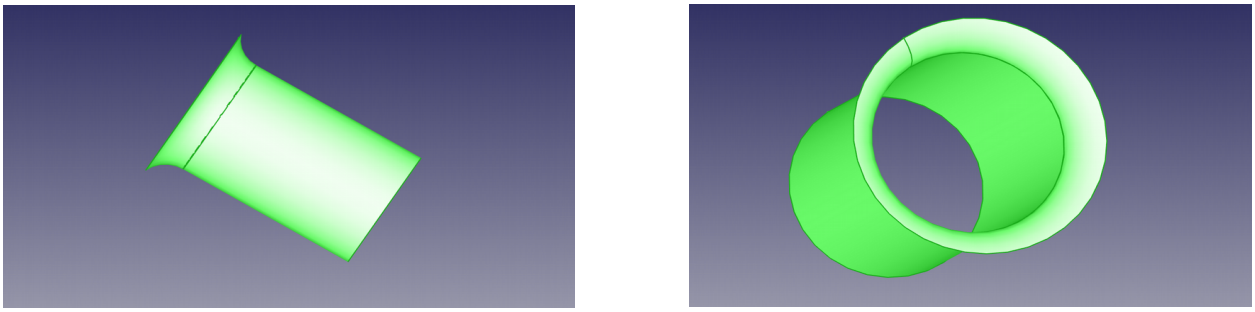


Figure 3.20: Conception du moyeu du rotor et du stator en 3D.

3.10. Conception de la ceinture

Le programme du conception du ceinture est donné en annexe 11.



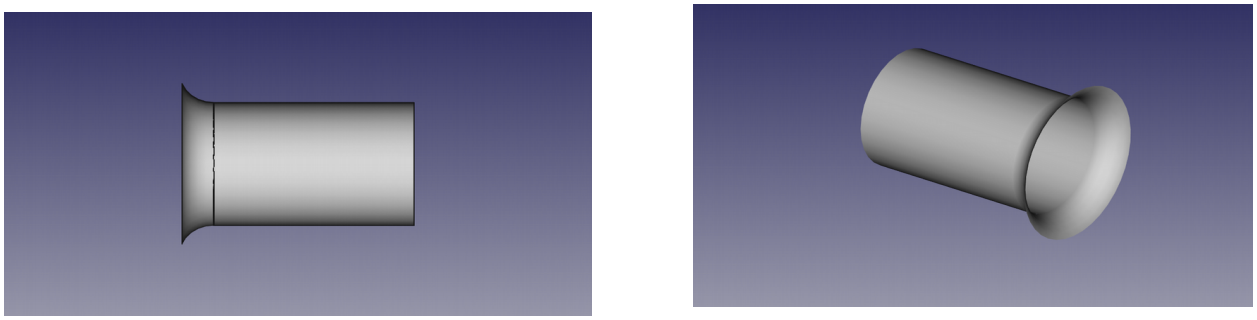


Figure 3.21: Conception de la ceinture en 3D.

3.11 Conception du rotor

Le programme de conception du rotor complet n'est pas encore achevé et fera l'objet de futures travaux. Nous avons essayé de faire le montage et les résultats sont présentés ci-dessous :

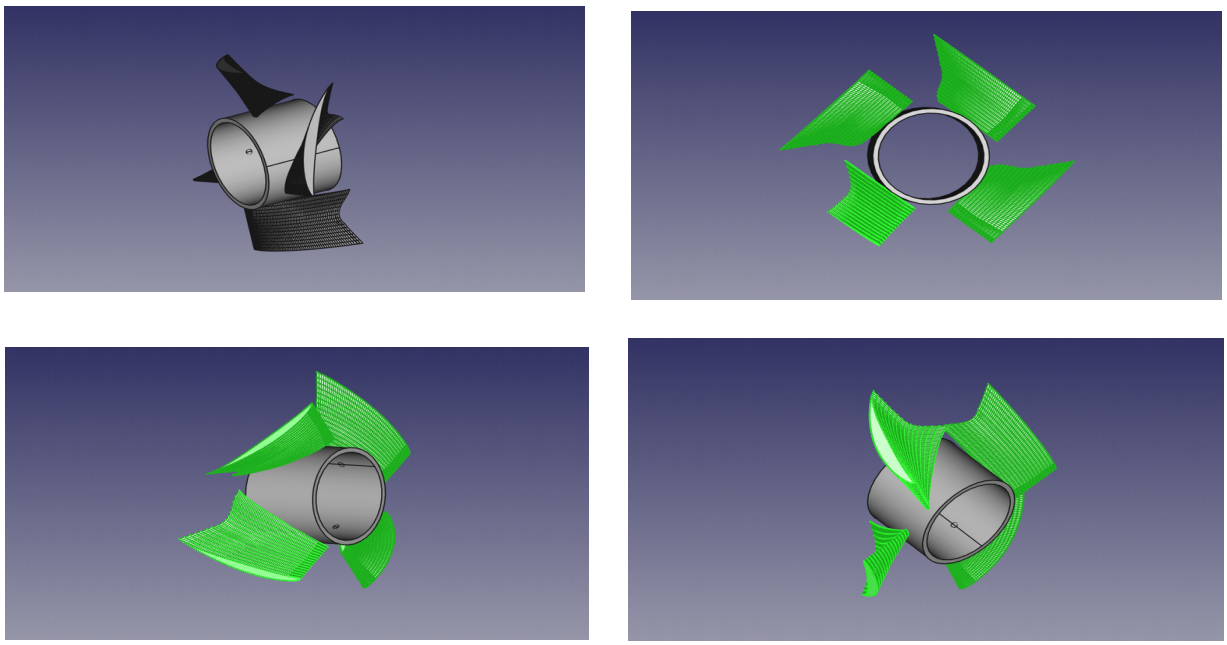
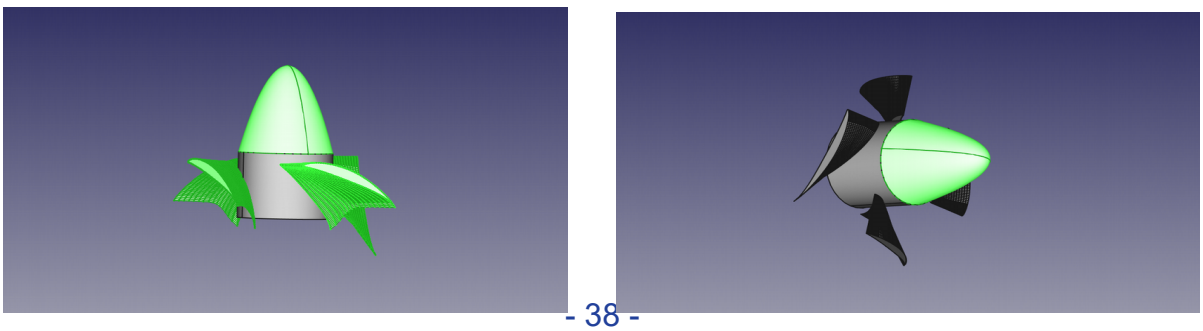


Figure 3.22: Conception du rotor en 3D.



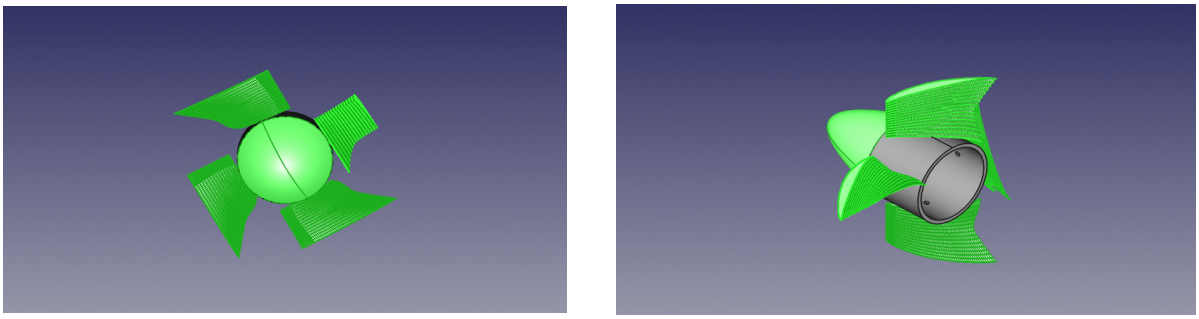


Figure 3.23: Conception du rotor complet 3D.

3.12 **Conception du stator**

Le programme de conception du stator complet n'est pas encore achevé et fera l'objet de futures travaux. Nous avons essayé de faire le montage et les résultats sont ci-dessous :

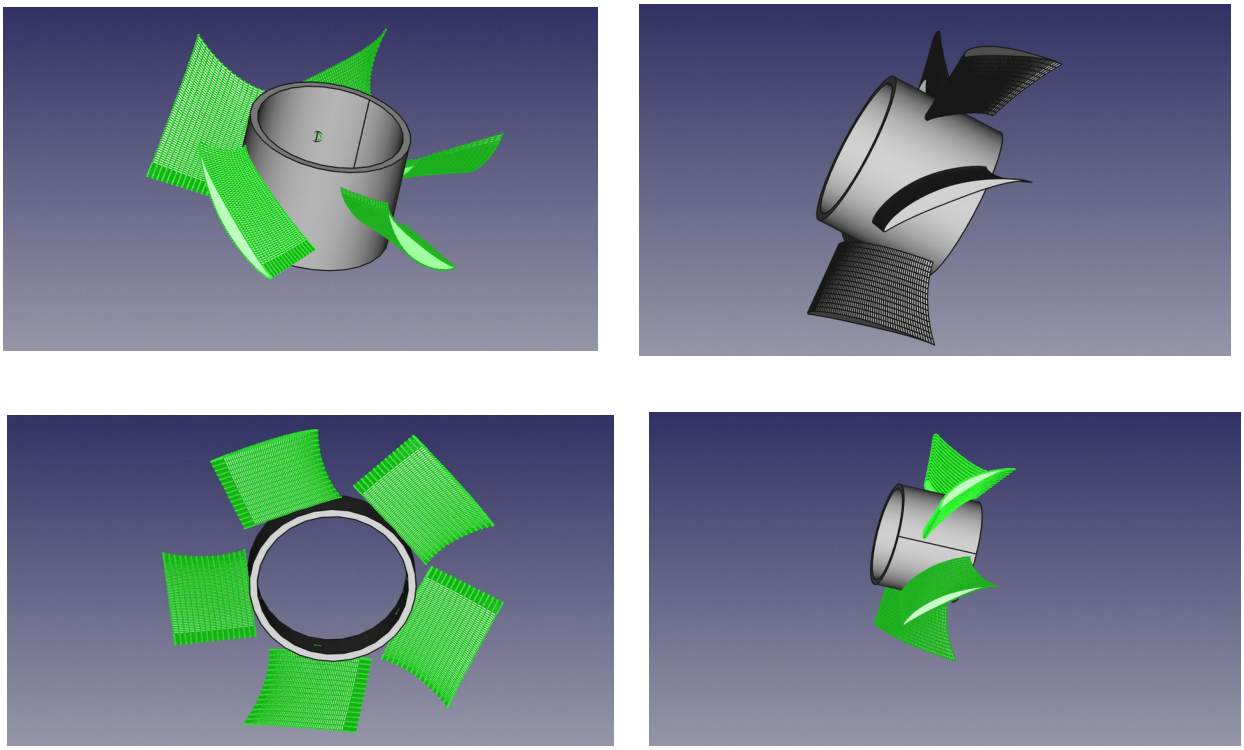


Figure 3.24: Conception du stator complet 3D.

3.13. **Montage de la pompe**

Les derniers essais que nous avons mené pour le montage de la pompe nous donnent les résultats suivants :

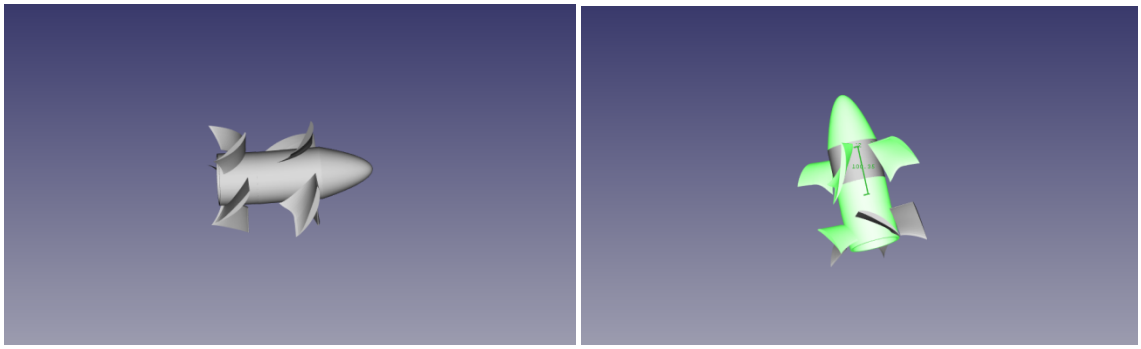


Figure 3.25: Montage du rotor et stator.

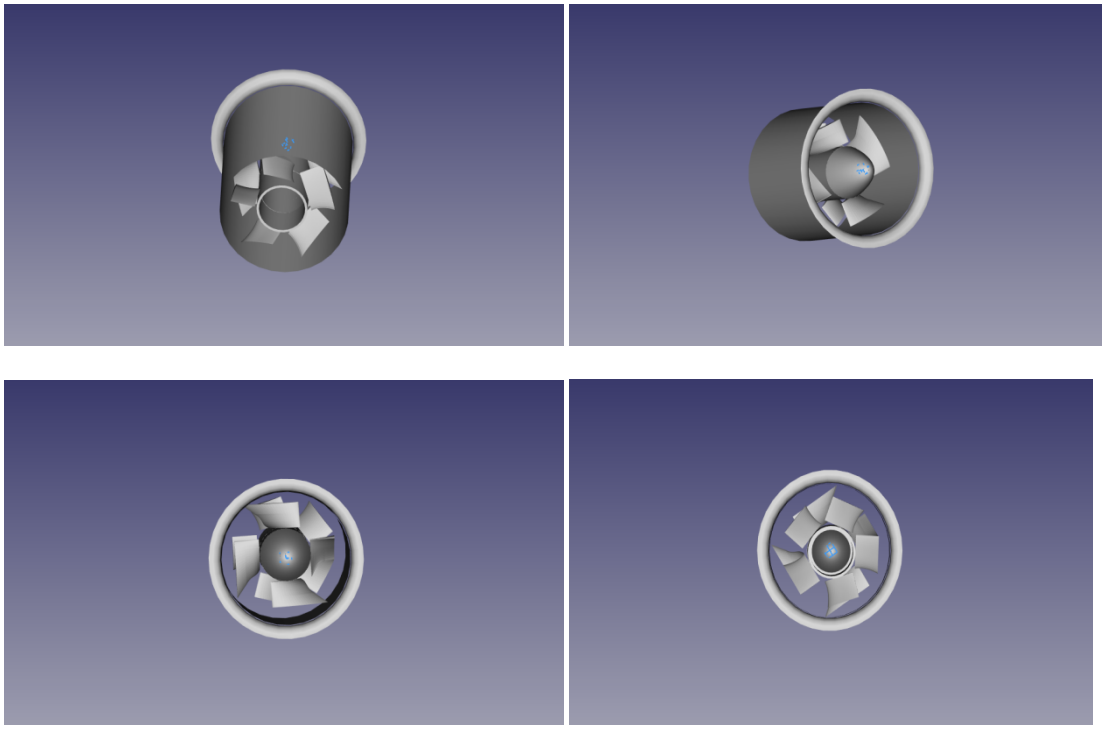


Figure 3.26: Montage de la pompe.

Il nous reste, bien sûr, à automatiser cette tache mais après avoir mis en place une méthodologie de montage pas-à-pas de toute la pompe pour pouvoir mettre en oeuvre le programme *Python* correspondant.

3.14. **Conclusion**

Nous avons essayé d'écrire des programmes ou « macros » en *Python* afin de minimiser l'intervention de l'utilisateur et de faciliter la conception de la pompe axiale qui est un modèle géométriquement très complexe. Les programmes que nous avons écrit sont mentionnés en annexes et sont bien commentés afin de faciliter leur compréhension.

Une fois exécutés dans l'interpréteur *Python* de *FreeCAD* ou bien à travers les macro-commandes, ces programmes font la conception des différentes parties de la pompe en les dessinant en 3D, en sauvegardant leurs images au format « *.png » et en effectuant leurs dessins techniques sur format A4 puis en le sauvegardant au format « *.pdf » et tout cela, sans aucune intervention de l'utilisateur.

CONCLUSION GENARALE ET PERSPECTIVES

Dans cette étude, nous avons exploité les résultats que donnent le logiciel de dimensionnement des pompes axiales « *PompAx* » et qui se présentent sous forme de 16 fichiers donnant chacun des profils (NACA) constituant le rotor et le stator ainsi qu'un second fichier donnant la géométrie de ces profils comprenant le rayon indiquant la position de chaque profil, l'angle de calage et la corde de ce profil.

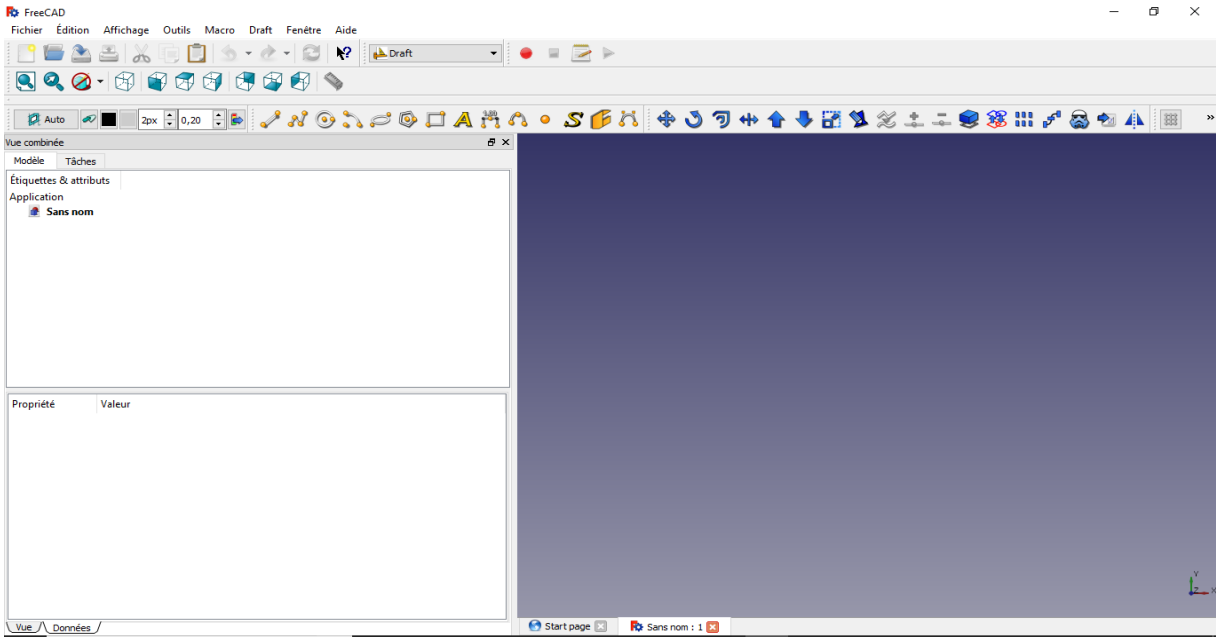
Nous avons atteint une grande partie de notre objectif principal qui consiste à automatiser l'exploitation de ces fichiers depuis la sortie de « *PompAx* » jusqu'à la conception finale de la pompe 3D afin de la préparer à la simulation numérique avec un logiciel de CFD. En effet, l'outil que nous avons utilisé dans ce travail est le logiciel de conception 3D paramétrique «*FreeCAD*» combiné au langage de programmation «*Python*» ; ceci nous a permis d'écrire nos différents programmes ou macros.

Nous avons établis environs 07 programmes ou « macros » qui permettent d'automatiser totalement la conception des différentes parties constituant la pompe axiale qui est géométriquement très complexe à réaliser. Une fois exécutés dans l'interpréteur *Python* de *FreeCAD* ou bien à travers les macro-commandes, ces programmes font la conception des différentes parties de la pompe en les dessinant en 3D, en sauvegardant leurs images au format « *.png » et en effectuant leurs dessins techniques sur format A4 puis en le sauvegardant au format « *.pdf » et tout cela, sans aucune intervention de l'utilisateur.

L'objectif final de ce travail n'est pas encore atteint mais le plus important a été fait et nous avons donné tous les détails nécessaires à la poursuite de ce travail afin d'aboutir au résultat attendu de se projet. Une interface graphique permettant de choisir les différentes parties de la pompe avec tout ce qui est nécessaire fera l'objet principal de la continuité de ce travail. Cette interface pourra être directement intégrée à *FreeCAD* et sera écrite en utilisant *QtDesigner* et *Python*.

Annexes

Annexe 01: Commandes de l'atelier Draft



➢ Dessin d'objets:

Les outils de création d'objets sont :

-  **Ligne**: Trace un segment de ligne à partir de 2 points
-  **Filaire**: Trace une ligne composée de plusieurs segments de lignes
-  **Cercle**: Trace un cercle à partir du centre et du rayon
-  **Arc**: Trace un segment d'arc à partir du centre, rayon, angle de départ et angle d'arrivée
-  **Ellipse**: Dessine une ellipse à partir de deux points opposés (coins)
-  **Polygone**: Dessine un polygone régulier à partir du centre et du rayon

-  **Rectangle**: Trace un rectangle à partir de 2 points opposés
-  **Texte**: Dessine une note en texte multiligne
-  **Cote**: Trace une cote
-  **B-Spline**: Dessine une courbe B-Spline à partir d'une série de points
-  **Point**: Insère un objet point
-  **ShapeString**: L'outil ShapeString insère une forme composée, qui représente une chaîne de texte
-  **Contraindre des faces:** Crée un nouvel objet sur la face de l'objet sélectionné
-  **Courbes de Bezier:** Dessine la courbe de Bézier à partir d'une série de points
-  **Label:** Place une étiquette avec une flèche pointant vers un élément sélectionné.

➢ Édition d'objets:

Ces outils permettent de modifier des objets existants. Ils fonctionnent sur les objets sélectionnés, s'il n'y a aucune sélection, vous serez invité à en faire une.

-  **Déplacer:** Déplace l'objet (ou les objets) d'un emplacement à un autre
-  **Rotation**: Pivote l'objet (ou les objets) d'un angle de départ à un angle d'arrivée
-  **Décalage**: Déplace les segments d'un objet à une certaine distance

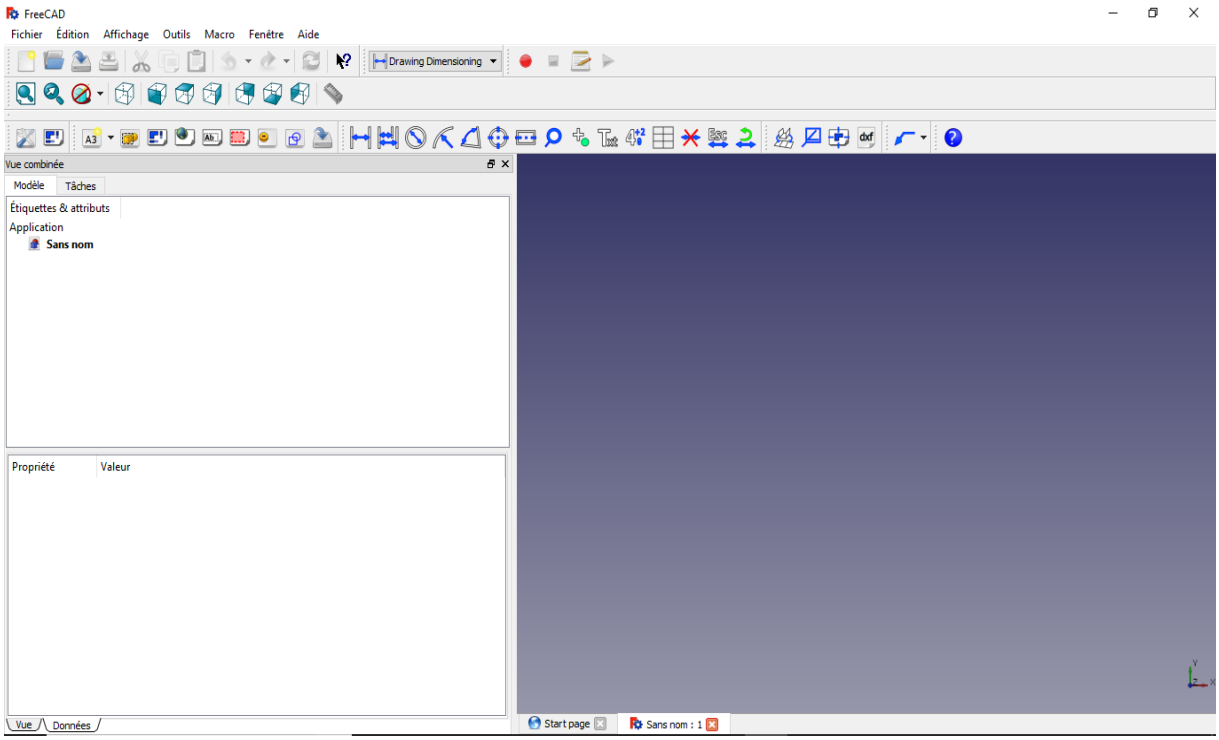
-  **Découper/prolonger (Trimex)**: Découpe ou prolonge un objet

-  **Mettre à niveau:** Joint des objets en un objet de plus haut niveau

-  **Rétrograder:** Explose des objets en objets de niveau inférieur

-  **Redimensionner:** Redimensionne l'objet (ou les objets) sélectionné(s) à partir d'un point de base

Annexe 02: Commandes de l'atelier Drawing



➢ OUTILS GRAPHIQUES:

Ces outils permettent de créer, configurer et exporter des mises en plan 2D:

Ouvrir un fichier vectoriel SVG: Ouvre une feuille de dessin précédemment sauvegardée au format de fichier SVG

Nouvelle feuille A3 paysage: Créé une nouvelle feuille à partir du gabarit A3 par défaut de FreeCAD

Insérer une vue: Insère une vue de l'objet sélectionné dans la feuille active

Annotation: Ajoute une annotation dans la feuille de dessin courante.

Clip: Ajoute un groupe de clip dans la feuille de dessin courante

 **Ouverture du navigateur internet:** Ouvre un aperçu de la feuille courante dans le navigateur.

 **Vue Orthogonale:** Crée automatiquement des vues orthogonales d'un objet sur la feuille de dessin courante.

 **Symbol:** Ajoute-le contenu d'un fichier SVG en tant que symbole dans la feuille de dessin en cours.

 **Draft View:** Insère une vue Brouillon spécial de l'objet sélectionné dans la feuille de dessin en cours.

 **Spreadsheet View:** Insère une vue d'une feuille de calcul sélectionnée dans la feuille de dessin en cours.

 **Exporter la feuille**: Exporte la feuille dans un fichier au format SVG

Project Shape: Crée une projection de l'objet sélectionné (Source) dans la vue 3D.

Annexe 03: Commandes de l'atelier Part

- **Modifier les objets :**

Voici les outils permettant de modifier les objets existants. Ils vous permettront de choisir quels objets modifier.

 **Opérations booléennes**: Réalise les principales opérations booléennes sur des objets.

 **Union:** Fusionne (additionne) deux objets.

 **Intersection:** Extrait la partie commune (intersection) de deux solides.

 **Soustraction:** Soustrait un objet à un autre.

 **Join features:** Crée un raccord et joint des objets (ex., tubes) (v0.16)

 **Connect:** Crée un raccord avec connexion interne de objets (v0.16)

 **Embed:** Intègre un objet clos dans un autre objet avec paroi (v0.16)

 **Cutout:** Crée une découpe dans la paroi d'un objet pour ajouter un autre objet avec paroi (v0.16)

- **Coupes et séparation d'objets: (v0.17)**

 **Boolean fragments:** Crée un objet avec chaque découpe booléenne créée entre divers pièces (v0.17)

 **Slice:** Découpe et sépare un objet à chaque intersection d'un autre objet (v0.17)

 **XOR**: Supprime l'espace partagé par un nombre pair d'objets (version symétrique de Cut) (v0.17)

 **Extrusion:** Extrude les faces planes d'un objet.

 **Congé:** Réalise un congé (arrondi) sur les arêtes d'un objet.

 **Révolution:** Crée un objet par révolution d'un autre objet autour d'un axe.

 **Coupe:** Crée une coupe par l'intersection d'un objet avec un plan de coupe.

 Plusieurs sections de coupes ...:

 **Chanfrein:** Chanfreine les arêtes d'un objet.

 **Miroir:** Réalise la symétrie axiale de l'objet sélectionné autour d'un axe donné.

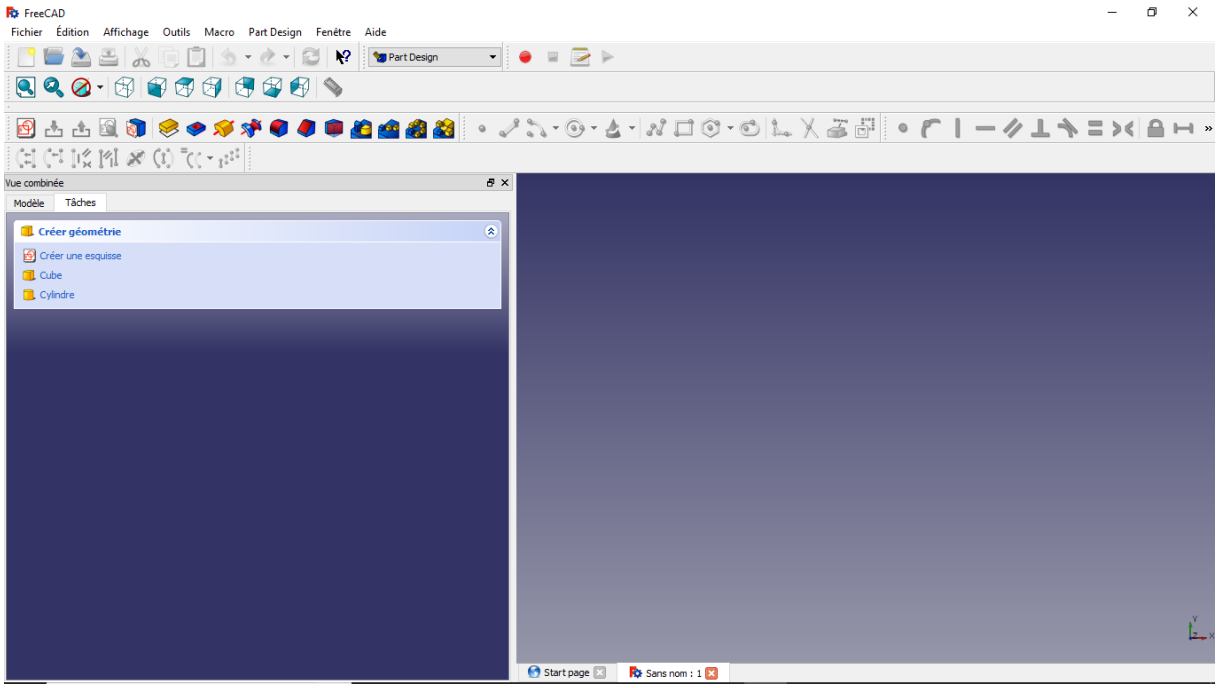
 **Balayage:** Balayage d'une ou plusieurs sections le long d'un chemin (tracé).

 **Loft**: Crée un lissage d'un profil jusqu'un autre profil

 **Offset:** Crée une copie à l'échelle de l'objet sélectionné.

 **Lissage:** Lissage d'une série de profils.

Annexe 04: Commandes de l'atelier PartDesign



Les outils:

Les outils Part Design sont situés dans le menu Part Design qui apparaît lorsque l'atelier Part Design est chargé.

Outils Structure:

Il s'agit d'outils pour organiser l'arborescence Modèle:

Pièce: ajoute un conteneur Pièce dans le document actif et le rend actif.

Outils d'assistance Part Design:

Créer un corps: crée un corps dans le document actif et le rend actif.

 **Créer une esquisse:** crée une nouvelle esquisse sur un plan ou une face sélectionnée. Si rien n'est sélectionné, l'utilisateur est invité à sélectionner un plan dans le panneau Tâches. L'interface bascule ensuite vers l'atelier Sketcher en mode d'édition d'esquisse.

 **Éditer l'esquisse:** édite l'esquisse sélectionnée.

 **Appliquer une esquisse:** applique une esquisse sur une face ou un plan sélectionné du corps actif.

Outils de référence:

 **Créer un point de référence:** crée un point de référence dans le corps actif.

 **Créer une ligne de référence:** crée une ligne de référence (droite) dans le corps actif.

 **Créer un plan de référence:** crée un plan de référence dans le corps actif.

 **Créer une forme liée:** crée une forme liée dans le corps actif.

 **Créer un clone**: crée un clone dans le corps actif.

Outils additifs:

Ces outils permettent de créer des fonctions de base ou d'ajouter de la matière à un corps solide existant.

 **Protrusion:** extrude un objet solide à partir de l'esquisse sélectionnée.

 **Révolution**: crée un solide par révolution d'une esquisse autour d'un axe. L'esquisse doit former un profil fermé.

 **Lissage additif:** crée un solide en réalisant une transition entre au moins deux esquisses.

 **Balayage additif:** crée un solide en balayant une ou plusieurs esquisse(s) le long d'un chemin ouvert ou fermé.

 **Cube additif:** crée un cube additif.

 **Cône additif:** crée un cône additif.

 **Cylindre additif:** crée un cylindre additif.

 **Ellipsoïde additif:** crée un ellipsoïde additif.

 **Prisme additif:** crée un prisme additif.

 **Sphère additive:** crée une sphère additive.

 **Tore additif:** crée un tore additif.

 **Cale additif:** crée une cale additive.

Outils soustractifs:

Ces outils permettent d'enlever de la matière à un corps solide existant :

 **Cavité:** crée une cavité à partir de l'esquisse sélectionnée.

 **Perçage:** crée une fonction perçage à partir de l'esquisse sélectionnée. L'esquisse doit contenir un ou plusieurs cercles.

 **Enlèvement de matière par révolution:** crée une rainure par révolution d'une esquisse sur un axe.

 **Lissage soustractif:** crée un solide en réalisant une transition entre au moins deux esquisses puis la soustrait du corps actif.

 **Balayage soustractif:** crée un solide en balayant une ou plusieurs esquisse(s) le long d'un chemin ouvert ou fermé puis le soustrait du corps actif.

 **Cube soustractif:** crée un cube soustractif.

 **Cône soustractif:** crée un cône soustractif.

 **Cylindre soustractif:** crée un cylindre soustractif.

 **Ellipsoïde soustractif:** crée un ellipsoïde soustractif.

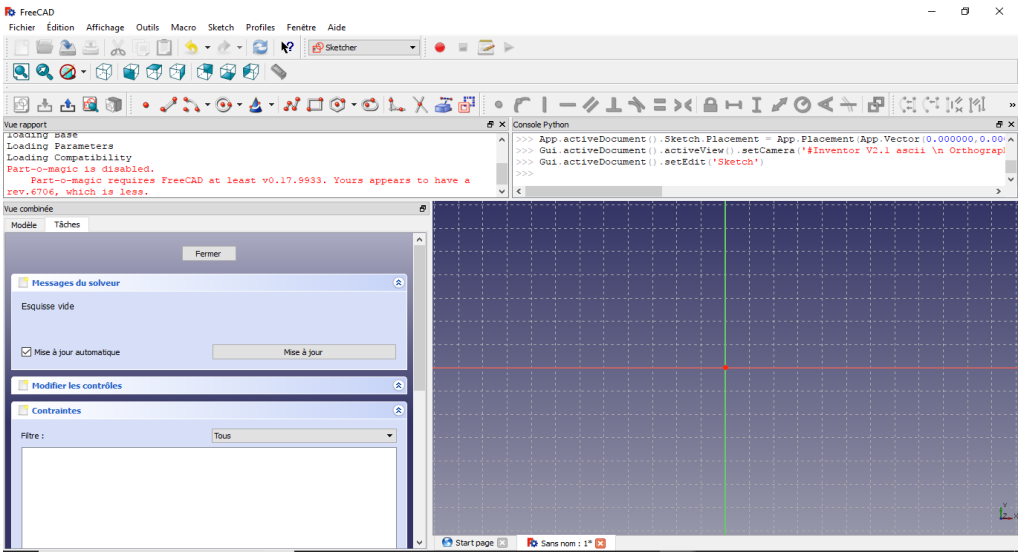
 **Prisme soustractif:** crée un prisme soustractif.

 **Sphère soustractive:** crée une sphère soustractive.

 **Tore soustractif:** crée un tore soustractif.

 **Cale soustractive:** crée une cale soustractive

Annexe 05: Commandes de l'atelier Sketcher



Les outils:

Les outils Atelier Esquisse sont tous situés dans le menu Sketch qui s'affiche lorsque vous chargez votre plan de travail.

Géométries d'esquisse:

Ces outils permettent de créer des objets.

Point: dessine un point.

Ligne: dessine une ligne entre 2 points.

 **Arc**: dessine un segment d'arc à partir du centre, rayon, angle de départ et angle d'arrivée.

 **Arc par 3 points:** dessine un arc de cercle sur deux points et un troisième point pour la circonférence.

 **Cercle:** dessine un cercle à partir de son centre et du rayon.

 **Cercle par 3 points:** dessine un cercle à partir de trois points.

 Sections coniques:

 **Ellipse:** dessine une ellipse à partir du centre, d'un point sur le grand rayon et d'un point sur le petit rayon. (v0.15)

 **Ellipse par 3 points:** dessine une ellipse à partir du grand diamètre (2 points) et d'un point sur le petit rayon. (v0.15)

 **Arc d'ellipse:** dessine une ellipse à partir du centre, d'un point sur le grand rayon, avec un point de départ et un point d'arrivée. (v0.15)

 **Arc d'hyperbole**: dessine un arc d'hyperbole. (v0.17)

 **Arc de parabole**: dessine un arc de parabole. (v0.17)

 **Polyligne:** dessine une ligne composée de plusieurs segments connectés entre eux.

 **Rectangle:** dessine un rectangle à partir de 2 points opposés.

 **Triangle:** dessine un triangle équilatéral inscrit dans un cercle. (v0.15)

 **Carré:** dessine un carré inscrit dans un cercle. (v0.15)

 **Pentagone:** dessine un pentagone régulier inscrit dans un cercle. (v0.15)

 **Hexagone**: dessine un hexagone régulier inscrit dans un cercle. (v0.15)

 **Heptagone:** dessine un heptagone régulier inscrit dans un cercle. (v0.15)

 **Octogone**: dessine un octogone régulier inscrit dans un cercle. (v0.15)

 **Clavette:** dessine une clavette de type A en entrant le centre du demi-cercle, le point pour le rayon et le point final du deuxième demi-cercle. (v0.15)

 **Congé:** crée un congé entre deux lignes connectées en un point. Sélectionnez les deux lignes, ou cliquez sur le sommet commun, puis activez l'outil.

 **Ajuster:** ajuste une ligne, un cercle ou un arc par rapport à l'emplacement du clic.

 **Géométrie externe:** crée une arête liée à une géométrie externe.

 **Mode Construction**: bascule un élément vers / depuis le mode construction. Les éléments de construction (en bleu) sont ignorés lors d'une opération de géométrie 3D (obsolète, version 0.15).

 **Mode Construction:** bascule les éléments vers / depuis le mode Construction. Dans FreeCAD v0.16, il a été ajouté la possibilité de créer une géométrie directement dans le Mode Construction, et l'icone a été changée. Le fait de sélectionner une géométrie existante et de cliquer sur cet outil bascule la géométrie en mode Construction. Dans FreeCAD v0.16, quand on clique sur cet outil alors qu'aucune géométrie n'est sélectionnée, cela change le mode pour les prochains objets à créer (Normal / Construction).

 **Coïncident:** crée une contrainte coïncidente (point sur point) entre deux sommets sélectionnés.

 **Point sur objet:** crée une contrainte point-sur-objet sur les éléments sélectionnés. L'un des éléments doit être un sommet, l'autre une ligne, un arc ou un cercle.

 **Vertical:** crée une contrainte de verticalité sur les lignes ou segments de polylignes sélectionnés. Plus d'un élément peut être sélectionné.

 Horizontal: crée une contrainte d'horizontalité sur les lignes ou segments de polylignes sélectionnés. Plus d'un élément peut être sélectionné.

 **Parallèle:** crée une contrainte de parallélisme entre deux lignes sélectionnées.

 **Perpendiculaire:** crée une contrainte de perpendicularité entre deux lignes sélectionnées.

 **Tangente:** crée une contrainte de tangence entre deux éléments sélectionnés, ou de colinéarité entre deux lignes.

 **Égalité**: crée une contrainte d'égalité entre au moins deux éléments sélectionnés. Contraindra la longueur pour des lignes et le rayon pour des cercles et des arcs.

 **Symétrie**: crée une contrainte symétrique entre deux points par rapport à une ligne.

Associées à des données numériques

Pour ces contraintes vous pouvez utiliser une expression. Les données peuvent prendre la forme d'une feuille de calcul de l'atelier Spreadsheet.

 **Fixe**: crée une contrainte fixe sur le sommet sélectionné en ajoutant des dimensions horizontale et verticale relatives à l'origine (les dimensions peuvent être éditées par la suite).

 **Distance horizontale:** fixe la distance horizontale entre deux sommets ou extrémités de ligne. Si un seul élément est sélectionné, la distance sera relative à l'origine.

 **Distance verticale:** fixe la distance verticale entre deux sommets ou extrémités de ligne. Si un seul élément est sélectionné, la distance sera relative à l'origine.

 **Longueur:** fixe la longueur d'une ligne sélectionnée, ou la distance entre une ligne et un point. La distance sera perpendiculaire à la ligne.

 **Rayon**: crée une contrainte radiale sur un arc ou un cercle sélectionné en ajoutant un rayon. Cette dimension pourra être éditée par la suite.

 **Angle Interne**: crée une contrainte d'angle interne entre deux lignes sélectionnées.

 **Loi de Snell:** contraint deux lignes à respecter une loi de réfraction simulant la trajectoire de la lumière à travers une interface. (v 0.15)

 **Alignement Interne:** aligne les éléments selectionnés à la forme sélectionnée (par exemple, contraint une ligne à devenir le grand axe d'une ellipse).

 **Basculement de Contrainte:** bascule la barre d'outils ou les contraintes sélectionnées vers / depuis le Mode Référence. v0.16

Autres :

 **Nouvelle esquisse:** crée une nouvelle esquisse sur un plan ou une face sélectionnée. Si rien n'est sélectionné, le plan XY sera utilisé par défaut.

 **Éditer l'esquisse:** édite l'esquisse sélectionnée.

Annexe 06: Programme de conception de l'aube du rotor

```
# -*- coding: utf-8 -*-
#importer les base du freecad:
import importDXF
import Draft
import FreeCAD
import Part
from FreeCAD import Base

rR=[] # Rayon de la coupe
GR=[] # Calage du profile
cR=[] # Corde du profile
# Changer l'option d'importation des fichiers DXF (True ==> DWire et False ==> Polyline)
```

```
FreeCAD.ParamGet("User parameter:BaseApp/Preferences/Mod/Draft").SetBool("dxfCreateDraft",True)

# Nombre de divisions de chaque segment formant le profile:
NBD=2
i=0
j=0
g = lambda x: [float(x[0])*1000]
h=lambda x: [float(x[4])]
s=lambda x: [float(x[5]),4]
# Ouvre le fichier des caractéristiques géométriques en lecture:
REP=u"C:/Users/Master/Desktop/n/N2/"
# nom du document:
nomDoc = "Profils_Rotor"
# Nombre de coupes de l'aube:
Ncoupes = 16
```

```
# nom de l'aube:
nomAube = "Aube_Rotor"
# Boucles sur toutes les lignes jusqu'à la fin et extraction des colonnes:
# 1, 5 et 6 pour le Rotor qui correspondent aux colonnes 0, 4 et 5.
# Les valeurs multipliées par 1000 sont converties en mm
#lire les valeurs des reyons du rotor:
with open(REP+"PN2.GEO") as f:
    next = f.readline()
   reyon = f.readlines()
   reyon =[g(e.split()) for e in reyon ]
#mettre les valeurs dans une liste:
   for i in range(Ncoupes):
        rR[j:]=reyon[i]
       j=i+1
#lire l'angle GAMA du rotor:
with open(REP+"PN2.GEO") as f:
     next=f.readline()
    GAMA_values=f.readlines()
    GAMA_values = [h(i.split()) for i in GAMA_values]
#mettre les valeurs dans une liste:
    for i in range(Ncoupes):
        GR[j:]=GAMA_values [i]
         j=1+i
with open(REP+"PN2.GEO") as f:
      next=f.readline()
     CORD=f.readlines()
     CORD = [s(e.split()) for e in CORD]
#mettre les valeurs dans une liste:
     for i in range(Ncoupes):
        cR[j:]=CORD[i]
        j=i+1
for i in range(Ncoupes):
# Nom du fichier:
     nomCOUPE="COUPER"+str(i+1)
# Importation des profiles:
    importDXF.insert(REP+nomCOUPE+".DXF",nomDoc)
```

```
# changement du nombre de divisions:
FreeCAD.getDocument(nomDoc).getObject("DWire").Subdivisions = NBD
# Changement d'échelle:
Draft.scale([FreeCAD.ActiveDocument.DWire],delta=FreeCAD.Vector(cR[0],cR[0],0.0),center=FreeCAD.Vect
or(0.0,0.0,rR[0]),copy=False)
# Déplacement et rotation du profile:
FreeCAD.getDocument(nomDoc).getObject("Scale").Placement =
App.Placement(App.Vector(0,0,rR[0]),App.Rotation(App.Vector(0,0,1),GR[0]))
for i in range(Ncoupes):
    if i<9:
# Changement du nombre de divisions:
        FreeCAD.getDocument(nomDoc).getObject("DWire00"+str(i+1)).Subdivisions = NBD
# Changement d'échelle:
  Draft.scale([FreeCAD.ActiveDocument.DWire],delta=FreeCAD.Vector(cR[i+1],cR[i+1],0.0),center=FreeCAD
Vector(0.0,0.0,rR[i+1]),copy=False)
# Déplacement et rotation du profile:
         FreeCAD.getDocument(nomDoc).getObject("Scale00"+str(i+1)).Placement =
App.Placement(App.Vector(0,0,rR[i+1]),App.Rotation(App.Vector(0,0,1),GR[i+1]))
     if i>9:
# changement du nombre de divisions:
         FreeCAD.getDocument(nomDoc).getObject("DWire0"+str(i)).Subdivisions =NBD
# Changement d'échelle:
          Draft.scale([FreeCAD.ActiveDocument.DWire],delta=FreeCAD.Vector(cR[i],cR[i],0.0),center=
Free.CAD.Vector(0.0,0.0,rR[i]),copy=False)
```

Annexe 07: Programme de conception de l'aube du stator

```
# -*- coding: utf-8 -*-
#importer les base du freecad:
import importDXF
import Draft
import FreeCAD
import Part
from FreeCAD import Base

rS=[] # Rayon de la coupe
GS=[] # Calage du profile
cS=[] # Corde du profile
# Changer l'option d'importation des fichiers DXF (True ==> DWire et False ==> Polyline)
FreeCAD.ParamGet("User parameter:BaseApp/Preferences/Mod/Draft").SetBool("dxfCreateDraft",True)
# Nombre de divisions de chaque segment formant le profile:
NBD=2
i=0
j=0
g = lambda x: [float(x[0])*1000]
n=lambda x: [float(x[3])]
h=lambda x: [float(x[4])]
# Ouvre le fichier des caractéristiques géométriques en lecture:
REP=u"C:/Users/Master/Desktop/n/N2/"
# nom du document:
nomDoc = "Profils_Stator"
# Nombre de coupes de l'aube:
Ncoupes = 16
# nom de l'aube:
nomAube = "Aube_Rotor"
# Boucles sur toutes les lignes jusqu'à la fin et extraction des colonnes:
# 1, 4 et 5 pour le Stator qui correspondent aux colonnes 0, 3 et 4.
# Les valeurs multipliées par 1000 sont converties en mm
#lire les valeurs des reyons du rotor:
with open(REP+"PN2.GEO") as f:
    next= f.readline()
    rayon = f.readlines()
```

```
    rayon =[g(e.split()) for e in rayon ]
#mettre les valeurs dans une liste:
    for j in range(16):
        j=16+j
        rS [i:]=rayon[j]
       i=i+1
#lire l'angle GAMA du rotor:
with open(REP+"PN2.GEO") as f:
      next = f.readline()
     GAMA=f.readlines()
     GAMA = [n(e.split()) for e in GAMA]
#mettre les valeurs dans une liste:
     for j in range(16):
          j=16+j
          GS [i:]=GAMA[j]
          i=i+1
#Lire les valeurs de la  Corde:
with open(REP+"PN2.GEO") as f:
       next=f.readline()
      CORD_S=f.readlines()
      CORD_S = [h(e.split()) for e in CORD_S]
#mettre les valeurs dans une liste:
     for j in range(16):
          j=16+j
         cS[i:]=CORD_S[j]
         i=i+1
for i in range(Ncoupes):
# Nom du fichier:
      nomCOUPE="COUPER"+str(i+1)
# Importation des profiles:
     importDXF.insert(REP+nomCOUPE+".DXF",nomDoc)
# Changement du nombre de divisions:
FreeCAD.getDocument(nomDoc).getObject("DWire").Subdivisions = NBD
# Changement d'échelle:
Draft.scale([FreeCAD.ActiveDocument.DWire],delta=FreeCAD.Vector(cS[0],cS[0],0.0),
center=FreeCAD.Vector(0.0,0.0,rS[0]),copy=False)
```

```
# Déplacement et rotation du profile:
FreeCAD.getDocument(nomDoc).getObject("Scale").Placement =
App.Placement(App.Vector(0,0,rS[0]),App.Rotation(App.Vector(0,0,1),GS[0]))
for i in range(Ncoupes):
    if i<9:
# Changement du nombre de divisions:
        FreeCAD.getDocument(nomDoc).getObject("DWire00"+str(i+1)).Subdivisions = NBD
# Changement d'échelle:
        Draft.scale([FreeCAD.ActiveDocument.DWire],delta=FreeCAD.Vector(cS[i+1],cS[i+1],0.0),
center=FreeCAD.Vector(0.0,0.0,rS[i+1]),copy=False)
# Déplacement et rotation du profile:
        FreeCAD.getDocument(nomDoc).getObject("Scale00"+str(i+1)).Placement =
App.Placement(App.Vector(0,0,rS[i+1]),App.Rotation(App.Vector(0,0,1),GS[i+1]))
    if i>9:
# Changement du nombre de divisions:
        FreeCAD.getDocument(nomDoc).getObject("DWire0"+str(i)).Subdivisions =NBD
# Changement d'échelle:
        Draft.scale([FreeCAD.ActiveDocument.DWire],delta=FreeCAD.Vector(cS[i],cS[i],0.0),
center=FreeCAD.Vector(0.0,0.0,rS[i]),copy=False)

# Déplacement et rotation du profile:
        FreeCAD.getDocument(nomDoc).getObject("Scale0"+str(i)).Placement =
App.Placement(App.Vector(0,0,rS[i]),App.Rotation(App.Vector(0,0,1),GS[i]))
# Raffraichir la vue:
App.activeDocument().recompute()
# Effacer ce qui est inutile:
App.getDocument(nomDoc).removeObject("DWire")
for i in range(16):
if i<9:
    App.getDocument(nomDoc).removeObject("DWire00"+str(i+1))
if i>9:
    App.getDocument(nomDoc).removeObject("DWire0"+str(i))
for i in range(17):
if i<9:
    App.getDocument(nomDoc).removeObject("COUPER"+str(i+1))
    App.getDocument(nomDoc).removeObject("Coupe"+str(i+1))
```

```
if i>9:
    App.getDocument(nomDoc).removeObject("COUPER"+str(i))
    App.getDocument(nomDoc).removeObject("Coupe"+str(i))
# Raffraichir la vue:
App.activeDocument().recompute()
# Construction de l'aube par extrusion suivant les 16 profiles:
App.getDocument(nomDoc).addObject('Part::Loft','Loft')
App.getDocument(nomDoc).ActiveObject.Sections=[App.getDocument(nomDoc).Scale,
App.getDocument(nomDoc).Scale001, App.getDocument(nomDoc).Scale002,
App.getDocument(nomDoc).Scale003, App.getDocument(nomDoc).Scale004,
App.getDocument(nomDoc).Scale005, App.getDocument(nomDoc).Scale006,
App.getDocument(nomDoc).Scale007, App.getDocument(nomDoc).Scale008,
App.getDocument(nomDoc).Scale009, App.getDocument(nomDoc).Scale010,
App.getDocument(nomDoc).Scale011, App.getDocument(nomDoc).Scale012,
App.getDocument(nomDoc).Scale013, App.getDocument(nomDoc).Scale014,
App.getDocument(nomDoc).Scale015, ]
App.getDocument(nomDoc).ActiveObject.Solid=True
App.getDocument(nomDoc).ActiveObject.Ruled=True
App.getDocument(nomDoc).ActiveObject.Closed=False
# Raffraichir la vue:
FreeCAD.ActiveDocument.recompute()
# Afficher toute l'aube à l'écran:
Gui.SendMsgToActiveView("ViewFit")
```

Annexe 08: Programme de conception de l'entrée du rotor

```
# -*- coding: utf-8 -*-
import Draft
import FreeCAD
import Part
from FreeCAD import Base

g = lambda x: [float(x[0])*1000,4]
#lire les valeurs des reyons du rotor:
with open(REP+"PN2.GEO") as f:
    next= f.readline()
    reyon = f.readlines()
    reyon =[g(e.split()) for e in reyon ]
#Mettre les valeurs dans une liste:
   for i in range(Ncoupes):
        rR[j:]=reyon[i]
        j=i+1
Ri=rR[0] #Le reyon exterieur
e=5 #L'épiaseur
Re=Ri-e #Le reyon interieur
Hi=1.5*Ri #L'hauteur de cylindre
ep=Hi/2 #La position du l'emprainte
epp=1 #Le profond du l'emprainte
Lp=35 #longeaur de l'emprainte
nomdocs="support_rotor" #Le nom de fichier
App.newDocument(nomdocs)
#Ouvrire une nouveau fichier
App.setActiveDocument(nomdocs)
App.ActiveDocument=App.getDocument(nomdocs)
Gui.ActiveDocument=Gui.getDocument(nomdocs)
#Appler la base Sktecher et choisir le plan XY pour designer:
App.activeDocument().addObject('PartDesign::Body','Body')
Gui.activeView().setActiveObject('pdbody', App.activeDocument().Body)
App.activeDocument().Body.newObject('Sketcher::SketchObject','Sketch')
App.activeDocument().Sketch.Support = (App.activeDocument().XY_Plane, [''])
App.activeDocument().Sketch.MapMode = 'FlatFace'
```

```
App.ActiveDocument.recompute()
Gui.activeDocument().setEdit('Sketch')
Gui.activateWorkbench('SketcherWorkbench')
# Lancer l'atelier Part Designe:
import PartDesignGui
#Disigner les 2 cercles:
ActiveSketch = App.ActiveDocument.getObject('Sketch')
tv = Show.TempoVis(App.ActiveDocument)
if ActiveSketch.ViewObject.HideDependent:
objs = tv.get_all_dependent(ActiveSketch)
objs = filter(lambda x: not x.TypeId.startswith("TechDraw::"), objs)
objs = filter(lambda x: not x.TypeId.startswith("Drawing::"), objs)
tv.hide(objs)
if ActiveSketch.ViewObject.ShowSupport:
tv.show([ref[0] for ref in ActiveSketch.Support if not ref[0].isDerivedFrom("PartDesign::Plane")])
if ActiveSketch.ViewObject.ShowLinks:
tv.show([ref[0] for ref in ActiveSketch.ExternalGeometry])
tv.hide(ActiveSketch)
ActiveSketch.ViewObject.TempoVis = tv
del(tv)
ActiveSketch = App.ActiveDocument.getObject('Sketch')
if ActiveSketch.ViewObject.RestoreCamera:
ActiveSketch.ViewObject.TempoVis.saveCamera()
FreeCAD.ActiveDocument.recompute()
App.ActiveDocument.Sketch.addGeometry(Part.Circle(App.Vector(0.000000,0.000000,0),App.Vector(0,0,1),Ri
),False)
App.ActiveDocument.Sketch.addConstraint(Sketcher.Constraint('Coincident',0,3,-1,1))
App.ActiveDocument.recompute()
App.ActiveDocument.Sketch.addGeometry(Part.Circle(App.Vector(0.000000,0.000000,0),App.Vector(0,0,1),Re
),False)
App.ActiveDocument.Sketch.addConstraint(Sketcher.Constraint('Coincident',1,3,-1,1))
App.ActiveDocument.recompute()
#Fermeture de l'Atelier Sketcher et lance l'Atelier Part Designe:
Gui.getDocument(nomdocs).resetEdit()
ActiveSketch = App.ActiveDocument.getObject('Sketch')
tv = ActiveSketch.ViewObject.TempoVis
```

```
if tv:
tv.restore()
ActiveSketch.ViewObject.TempoVis = None
del(tv)
Gui.activateWorkbench('PartDesignWorkbench')
App.getDocument(nomdocs).recompute()
Gui.getDocument(nomdocs).getObject("Sketch").Visibility=True
#Construire le cylinder:
App.activeDocument().Body.newObject("PartDesign::Pad","Pad")
App.activeDocument().Pad.Profile = App.activeDocument().Sketch
App.activeDocument().Pad.Length = 10.0
App.ActiveDocument.recompute()
Gui.activeDocument().hide("Sketch")
App.ActiveDocument.recompute()
Gui.activeDocument().setEdit('Pad', 0)
Gui.Selection.clearSelection()
Gui.ActiveDocument.Pad.ShapeColor=Gui.ActiveDocument.Body.ShapeColor
Gui.ActiveDocument.Pad.LineColor=Gui.ActiveDocument.Body.LineColor
Gui.ActiveDocument.Pad.PointColor=Gui.ActiveDocument.Body.PointColor
Gui.ActiveDocument.Pad.Transparency=Gui.ActiveDocument.Body.Transparency
Gui.ActiveDocument.Pad.DisplayMode=Gui.ActiveDocument.Body.DisplayMode
Gui.activeDocument().hide("Sketch")
App.ActiveDocument.Pad.Length = Hi
App.ActiveDocument.Pad.Length2 = 100.000000
App.ActiveDocument.Pad.Type = 0
App.ActiveDocument.Pad.UpToFace = None
App.ActiveDocument.Pad.Reversed = 0
App.ActiveDocument.Pad.Midplane = 0
App.ActiveDocument.Pad.Offset = 0.000000
App.ActiveDocument.recompute()
Gui.activeDocument().resetEdit()
# Raffraichir la vue:
App.ActiveDocument.recompute()
App.getDocument(nomdocs).recompute()
# Afficher toute l'aube à l'écran:
Gui.SendMsgToActiveView("ViewFit")
```

Annexe 09: Programme de conception du moyeu du rotor

```
# -*- coding: utf-8 -*-
import Draft
import FreeCAD
import Part
from FreeCAD import Base

g = lambda x: [float(x[0])*1000,4]
#lire les valeurs des rayons:
with open(REP+"PN2.GEO") as f:
    next= f.readline()
    reyon = f.readlines()
    reyon =[g(e.split()) for e in reyon ]
#Mettre les valeurs dans une liste:
   for i in range(Ncoupes):
        rR[j:]=reyon[i]
        j=i+1
Ri=rR[0] #Le reyon extérieur
e=5             #L'épiaseur
Re=Ri-e      #Le rayon intérieur
Hi=1*Ri      #Hauteur de cylindre
ep=Hi/2      #La position du l'empreinte
epp=1        #Le profond du l'empreinte
Lp=35        #Longeaur de l'emprainte
nomdocs="moyeu_rotor" #Le nom de fichier
App.newDocument(nomdocs)
#Ouvrire une nouveau fichier
App.setActiveDocument(nomdocs)
App.ActiveDocument=App.getDocument(nomdocs)
Gui.ActiveDocument=Gui.getDocument(nomdocs)
#Appler la base Sktecher et choisir le plan XY pour designer:
App.activeDocument().addObject('PartDesign::Body','Body')
Gui.activeView().setActiveObject('pdbody', App.activeDocument().Body)
App.activeDocument().Body.newObject('Sketcher::SketchObject','Sketch')
App.activeDocument().Sketch.Support = (App.activeDocument().XY_Plane, [''])
App.activeDocument().Sketch.MapMode = 'FlatFace'
```

```
App.ActiveDocument.recompute()
Gui.activeDocument().setEdit('Sketch')
Gui.activateWorkbench('SketcherWorkbench')
# Lancer l'atelier Part Designe:
import PartDesignGui
#Disigner les 2 cercles:
ActiveSketch = App.ActiveDocument.getObject('Sketch')
tv = Show.TempoVis(App.ActiveDocument)
if ActiveSketch.ViewObject.HideDependent:
objs = tv.get_all_dependent(ActiveSketch)
objs = filter(lambda x: not x.TypeId.startswith("TechDraw::"), objs)
objs = filter(lambda x: not x.TypeId.startswith("Drawing::"), objs)
tv.hide(objs)
if ActiveSketch.ViewObject.ShowSupport:
tv.show([ref[0] for ref in ActiveSketch.Support if not ref[0].isDerivedFrom("PartDesign::Plane")])
if ActiveSketch.ViewObject.ShowLinks:
tv.show([ref[0] for ref in ActiveSketch.ExternalGeometry])
tv.hide(ActiveSketch)
ActiveSketch.ViewObject.TempoVis = tv
del(tv)
ActiveSketch = App.ActiveDocument.getObject('Sketch')
if ActiveSketch.ViewObject.RestoreCamera:
ActiveSketch.ViewObject.TempoVis.saveCamera()
FreeCAD.ActiveDocument.recompute()
App.ActiveDocument.Sketch.addGeometry(Part.Circle(App.Vector(0.000000,0.000000,0),App.Vector(0,0,1),Ri
),False)
App.ActiveDocument.Sketch.addConstraint(Sketcher.Constraint('Coincident',0,3,-1,1))
App.ActiveDocument.recompute()
App.ActiveDocument.Sketch.addGeometry(Part.Circle(App.Vector(0.000000,0.000000,0),App.Vector(0,0,1),Re
),False)
App.ActiveDocument.Sketch.addConstraint(Sketcher.Constraint('Coincident',1,3,-1,1))
App.ActiveDocument.recompute()
#Fermeture de l'Atelier Sketcher et lance l'Atelier Part Designe:
Gui.getDocument(nomdocs).resetEdit()
ActiveSketch = App.ActiveDocument.getObject('Sketch')
tv = ActiveSketch.ViewObject.TempoVis
```

```
if tv:
tv.restore()
ActiveSketch.ViewObject.TempoVis = None
del(tv)
Gui.activateWorkbench('PartDesignWorkbench')
App.getDocument(nomdocs).recompute()
Gui.getDocument(nomdocs).getObject("Sketch").Visibility=True
#Construire le cylinder:
App.activeDocument().Body.newObject("PartDesign::Pad","Pad")
App.activeDocument().Pad.Profile = App.activeDocument().Sketch
App.activeDocument().Pad.Length = 10.0
App.ActiveDocument.recompute()
Gui.activeDocument().hide("Sketch")
App.ActiveDocument.recompute()
Gui.activeDocument().setEdit('Pad', 0)
Gui.Selection.clearSelection()
Gui.ActiveDocument.Pad.ShapeColor=Gui.ActiveDocument.Body.ShapeColor
Gui.ActiveDocument.Pad.LineColor=Gui.ActiveDocument.Body.LineColor
Gui.ActiveDocument.Pad.PointColor=Gui.ActiveDocument.Body.PointColor
Gui.ActiveDocument.Pad.Transparency=Gui.ActiveDocument.Body.Transparency
Gui.ActiveDocument.Pad.DisplayMode=Gui.ActiveDocument.Body.DisplayMode
Gui.activeDocument().hide("Sketch")
App.ActiveDocument.Pad.Length = Hi
App.ActiveDocument.Pad.Length2 = 100.000000
App.ActiveDocument.Pad.Type = 0
App.ActiveDocument.Pad.UpToFace = None
App.ActiveDocument.Pad.Reversed = 0
App.ActiveDocument.Pad.Midplane = 0
App.ActiveDocument.Pad.Offset = 0.000000
App.ActiveDocument.recompute()
Gui.activeDocument().resetEdit()
# Raffraichir la vue:
App.getDocument(nomdocs).recompute()
# Afficher toute l'aube à l'écran:
Gui.SendMsgToActiveView("ViewFit")
```

Annexe 10: Programme de conception du moyeu du stator

```
# -*- coding: utf-8 -*-
import Draft
import FreeCAD
import Part
from FreeCAD import Base
g = lambda x: [float(x[0])*1000,4]
#lire les valeurs des rayons:
with open(REP+"PN2.GEO") as f:
    next= f.readline()
    reyon = f.readlines()
    reyon =[g(e.split()) for e in reyon ]
#Mettre les valeurs dans une liste:
   for i in range(Ncoupes):
        rR[j:]=reyon[i]
         j=i+1
Ri=rR[0] #Le reyon extérieur
e=5             #L'épiaseur
Re=Ri-e      #Le rayon intérieur
Hi=1*Ri      #Hauteur de cylindre
ep=Hi/2       #La position du l'empreinte
epp=1         #Le profond du l'empreinte
Lp=35         #Longeaur de l'emprainte
nomdocs="moyeu_stator" #Le nom de fichier
App.newDocument(nomdocs)
#Ouvrire une nouveau fichier
App.setActiveDocument(nomdocs)
App.ActiveDocument=App.getDocument(nomdocs)
Gui.ActiveDocument=Gui.getDocument(nomdocs)
#Appler la base Sktecher et choisir le plan XY pour designer:
App.activeDocument().addObject('PartDesign::Body','Body')
Gui.activeView().setActiveObject('pdbody', App.activeDocument().Body)
App.activeDocument().Body.newObject('Sketcher::SketchObject','Sketch')
App.activeDocument().Sketch.Support = (App.activeDocument().XY_Plane, [''])
App.activeDocument().Sketch.MapMode = 'FlatFace'
App.ActiveDocument.recompute()
```

```
Gui.activeDocument().setEdit('Sketch')
Gui.activateWorkbench('SketcherWorkbench')
# Lancer l'atelier Part Designe:
import PartDesignGui
#Designer les 2 cercles:
ActiveSketch = App.ActiveDocument.getObject('Sketch')
tv = Show.TempoVis(App.ActiveDocument)
if ActiveSketch.ViewObject.HideDependent:
objs = tv.get_all_dependent(ActiveSketch)
objs = filter(lambda x: not x.TypeId.startswith("TechDraw::"), objs)
objs = filter(lambda x: not x.TypeId.startswith("Drawing::"), objs)
tv.hide(objs)
if ActiveSketch.ViewObject.ShowSupport:
tv.show([ref[0] for ref in ActiveSketch.Support if not ref[0].isDerivedFrom("PartDesign::Plane")])
if ActiveSketch.ViewObject.ShowLinks:
tv.show([ref[0] for ref in ActiveSketch.ExternalGeometry])
tv.hide(ActiveSketch)
ActiveSketch.ViewObject.TempoVis = tv
del(tv)
ActiveSketch = App.ActiveDocument.getObject('Sketch')
if ActiveSketch.ViewObject.RestoreCamera:
ActiveSketch.ViewObject.TempoVis.saveCamera()
FreeCAD.ActiveDocument.recompute()
App.ActiveDocument.Sketch.addGeometry(Part.Circle(App.Vector(0.000000,0.000000,0),App.Vector(0,0,1),Ri
),False)
App.ActiveDocument.Sketch.addConstraint(Sketcher.Constraint('Coincident',0,3,-1,1))
App.ActiveDocument.recompute()
App.ActiveDocument.Sketch.addGeometry(Part.Circle(App.Vector(0.000000,0.000000,0),App.Vector(0,0,1),Re
),False)
App.ActiveDocument.Sketch.addConstraint(Sketcher.Constraint('Coincident',1,3,-1,1))
App.ActiveDocument.recompute()
#Fermeture de l'Atelier Sketcher et lance l'Atelier Part Designe:
Gui.getDocument(nomdocs).resetEdit()
ActiveSketch = App.ActiveDocument.getObject('Sketch')
tv = ActiveSketch.ViewObject.TempoVis
if tv:
```

```
tv.restore()
ActiveSketch.ViewObject.TempoVis = None
del(tv)
Gui.activateWorkbench('PartDesignWorkbench')
App.getDocument(nomdocs).recompute()
Gui.getDocument(nomdocs).getObject("Sketch").Visibility=True
#Construire le cylinder:
App.activeDocument().Body.newObject("PartDesign::Pad","Pad")
App.activeDocument().Pad.Profile = App.activeDocument().Sketch
App.activeDocument().Pad.Length = 10.0
App.ActiveDocument.recompute()
Gui.activeDocument().hide("Sketch")
App.ActiveDocument.recompute()
Gui.activeDocument().setEdit('Pad', 0)
Gui.Selection.clearSelection()
Gui.ActiveDocument.Pad.ShapeColor=Gui.ActiveDocument.Body.ShapeColor
Gui.ActiveDocument.Pad.LineColor=Gui.ActiveDocument.Body.LineColor
Gui.ActiveDocument.Pad.PointColor=Gui.ActiveDocument.Body.PointColor
Gui.ActiveDocument.Pad.Transparency=Gui.ActiveDocument.Body.Transparency
Gui.ActiveDocument.Pad.DisplayMode=Gui.ActiveDocument.Body.DisplayMode
Gui.activeDocument().hide("Sketch")
App.ActiveDocument.Pad.Length = Hi
App.ActiveDocument.Pad.Length2 = 100.000000
App.ActiveDocument.Pad.Type = 0
App.ActiveDocument.Pad.UpToFace = None
App.ActiveDocument.Pad.Reversed = 0
App.ActiveDocument.Pad.Midplane = 0
App.ActiveDocument.Pad.Offset = 0.000000
App.ActiveDocument.recompute()
Gui.activeDocument().resetEdit()
# Rafraîchir la vue:
App.getDocument(nomdocs).recompute()
# Afficher toute l'aube à l'écran:
Gui.SendMsgToActiveView("ViewFit")
```

Annexe 11: Programme de conception de la ceinture

```
# -*- coding: utf-8 -*-

import Draft
import FreeCAD
import Part
from FreeCAD import Base


g = lambda x: [float(x[0])*1000]
# Nombre de coupes de l'aube:
Ncoupes = 16
i=0
j=0
rR=[] # Rayon de la coupe
# Changer l'option d'importation des fichiers DXF (True ==> DWire et False ==> Polyline)
FreeCAD.ParamGet("User parameter:BaseApp/Preferences/Mod/Draft").SetBool("dxfCreateDraft",True)
#le nom du fichier
nomdoc='ceinture'
# Ouvre le fichier des caractéristiques géométriques en lecture:
REP=u"C:/Users/Master/Desktop/n/N2/"
App.newDocument(nomdoc)
App.setActiveDocument(nomdoc)
App.ActiveDocument=App.getDocument(nomdoc)
Gui.ActiveDocument=Gui.getDocument(nomdoc)
#lire les valeurs des reyons du rotor:
with open(REP+"PN2.GEO") as f:
    next = f.readline()
    reyon = f.readlines()
    reyon =[g(e.split()) for e in reyon ]
#mettre les valeurs dans une liste:
    for i in range(Ncoupes):
        rR[j:]=reyon[i]
        j=i+1
rC=rR[15]+1.7 #rayon du cienture
H='300' #heatur de cylindre
p=156 #le rayon du l'arc
```

```
#Dessin du ceinture en 2D
points=[FreeCAD.Vector(-
2.01555466652,0.986426174641,0.0),FreeCAD.Vector(2.04971671104,0.977885723114,0.0)]
line = Draft.makeWire(points,closed=False,face=True,support=None)
Draft.autogroup(line)
pl=FreeCAD.Placement()
pl.Rotation.Q=(0.0,-0.0,-0.0,1.0)
pl.Base=FreeCAD.Vector(-2.03263568878,1.83193409443,0.0)
circle = Draft.makeCircle(radius=50,placement=pl,face=True,startangle=-159.44396474,endangle=-
88.8426632615,support=None)
Draft.autogroup(circle)
#Creation du ceinture (3D)
Gui.activateWorkbench("PartWorkbench")
FreeCAD.ActiveDocument.addObject("Part::Revolution","Revolve")
FreeCAD.ActiveDocument.Revolve.Source = FreeCAD.ActiveDocument.Arc
FreeCAD.ActiveDocument.Revolve.Axis = (1.000000000000000,0.000000000000000,0.000000000000000)
FreeCAD.ActiveDocument.Revolve.Base = (0.000000000000000,0.000000000000000,0.000000000000000)
FreeCAD.ActiveDocument.Revolve.Angle = 360.000000000000000
FreeCAD.ActiveDocument.Revolve.Solid = False
FreeCAD.ActiveDocument.Revolve.AxisLink = None
FreeCAD.ActiveDocument.Revolve.Symmetric = False
FreeCADGui.ActiveDocument.Arc.Visibility = False
Gui.ActiveDocument.Revolve.ShapeColor=Gui.ActiveDocument.Arc.ShapeColor
Gui.ActiveDocument.Revolve.LineColor=Gui.ActiveDocument.Arc.LineColor
Gui.ActiveDocument.Revolve.PointColor=Gui.ActiveDocument.Arc.PointColor
FreeCAD.ActiveDocument.addObject("Part::Revolution","Revolve001")
FreeCAD.ActiveDocument.Revolve001.Source = FreeCAD.ActiveDocument.Line
FreeCAD.ActiveDocument.Revolve001.Axis = (1.000000000000000,0.000000000000000,0.000000000000000)
FreeCAD.ActiveDocument.Revolve001.Base = (0.000000000000000,0.000000000000000,0.000000000000000)
FreeCAD.ActiveDocument.Revolve001.Angle = 360.000000000000000
FreeCAD.ActiveDocument.Revolve001.Solid = False
FreeCAD.ActiveDocument.Revolve001.AxisLink = None
FreeCAD.ActiveDocument.Revolve001.Symmetric = False
FreeCADGui.ActiveDocument.Line.Visibility = False
Gui.ActiveDocument.Revolve001.ShapeColor=Gui.ActiveDocument.Line.ShapeColor
Gui.ActiveDocument.Revolve001.LineColor=Gui.ActiveDocument.Line.LineColor
Gui.ActiveDocument.Revolve001.PointColor=Gui.ActiveDocument.Line.PointColor
```

```
FreeCAD.getDocument(nomdoc).getObject("Arc").Placement = App.Placement(App.Vector(-
2.03264,p,0),App.Rotation(App.Vector(0,0,1),0))
FreeCAD.getDocument(nomdoc).getObject("Line").End = (2.04972, rC, 0)
FreeCAD.getDocument(nomdoc).getObject("Line").Start = (-2.01555, rC, 0)
FreeCAD.getDocument(nomdoc).getObject("Line").Length = H
#Rafrachaire la vu:
App.ActiveDocument.recompute()
#Affiche la ceinture dans toute l'écran
Gui.SendMsgToActiveView("ViewFit")
```

BIBLIOGRAPHIE

1- LASSAAD-Mazouzi

Etude de défiance d'une pompe a eau centrifuge de type g'uinard HP, 2015-2016.(p 12.13.14)

2- ***wikipedia*** : https://fr.wikipedia.org/wiki/Pompe_centrifuge

3- GILBERT-PERNOT

Chapiter 2 :les étapes de construction d'une hélice ,principe et conception d'une hélice, http://www.gilbert-pernot.fr/helice_calcul_fabrication.html#helice_trajectoire_developpement

4- FREECAD USER GUIDE

https://www.freecadweb.org/wiki/Workbenches/fr

5- FREECAD USER GUIDE

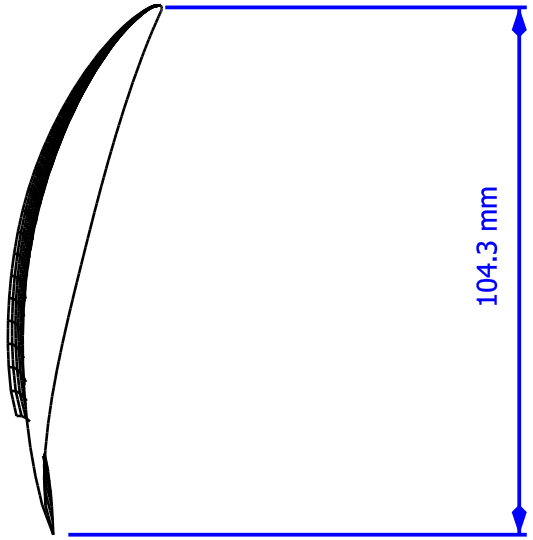
https://www.freecadweb.org/wiki/index.php?title=Macros

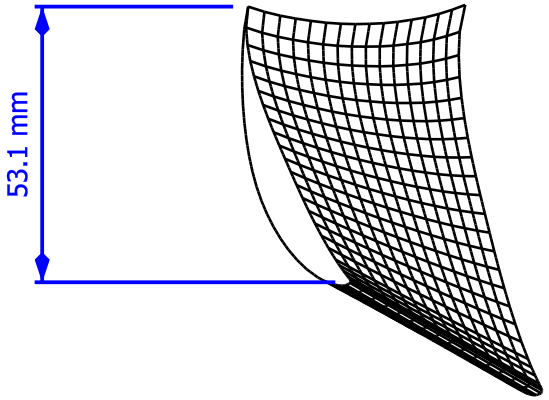
6- FREECAD USER GUIDE

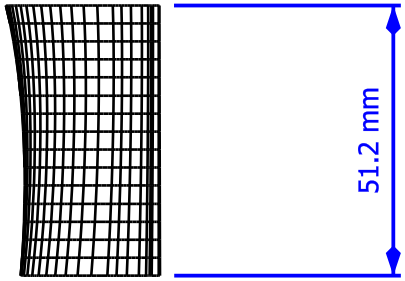
https://www.freecadweb.org/wiki/Introduction_to_Python

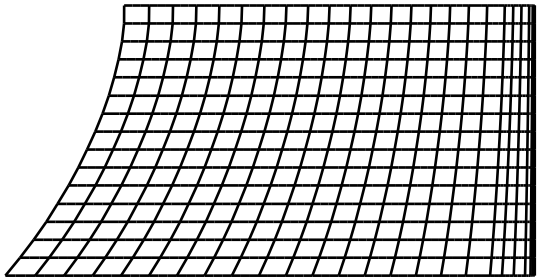
7- ***G.Mebarki,***

Mimoire-mebarki, Logicielle POMP.AX, 1998, (p 5,6)

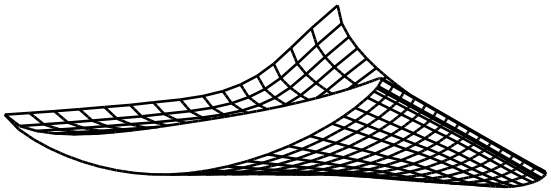


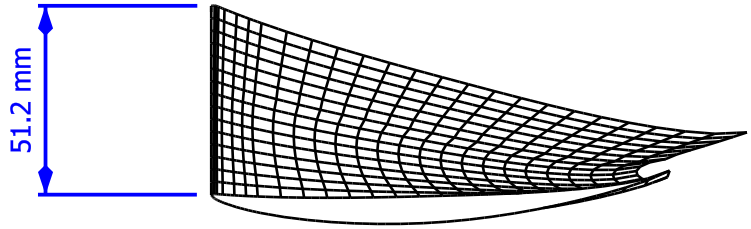


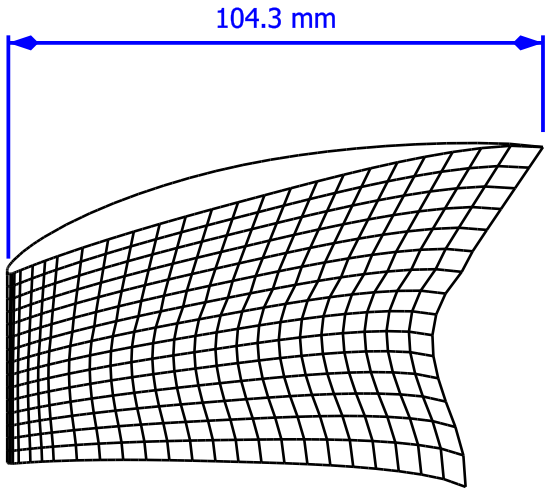


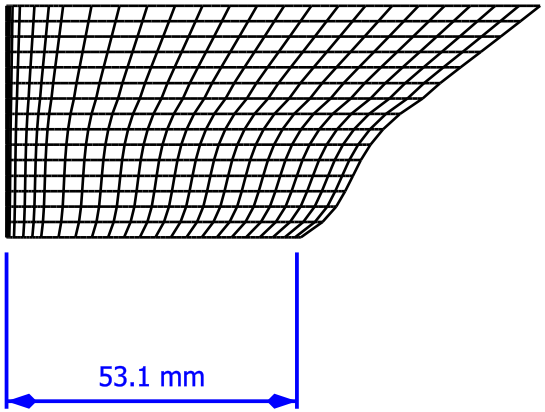


DESIGNED BY: Bitam Abderaouf DATE: 01/06/2018		Aube Stator		
SIZE A4				
SCALE 0.7	WEIGHT (kg) Weight	DRAWING NUMBER 1	SHEET EU	









DESIGNED BY: Bitam Abderaouf DATE: 01/06/2018		Aube Rotor	
SIZE A4			
SCALE 0.7	WEIGHT (kg) Weigth	DRAWING NUMBER 1	SHEET EU