

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUÉES
DÉPARTEMENT DE GENIE MECANIQUE

MÉMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Mécanique

Parcours : Master Professionnel

Spécialité : Productique

THÈME

**DÉVELOPPEMENT D'UN PROGRAMME D'EXTRACTION DES DONNÉES
GÉOMETRIQUES ET TOPOLOGIQUES D'UN MODÈLE 3D D'UNE PIÈCE A
PARTIR D'UN FICHER IGES**

Préparé par : Mrs HADJAZ Karim & BOUMAIZA Mohamed Tayeb

Devant le Jury :

Nom et prénoms	Grade	Lieu d'exercice	Qualité
Mr SAHLI Ahmed	Pr	UIK Tiaret	Président
Mr GUEMMOUR Mohamed Boutkhill	MCB	UIK Tiaret	Examinateur
Mr MECHEKOUR ElHadi	MCB	UIK Tiaret	Encadreur

PROMOTION 2016 /2017

DÉDICACES

Grâce à dieu le tout puissant, j'ai pu réaliser ce modeste travail en ce jour solennel, je dédie ce projet spécialement :

A mes parents qui m'ont entouré et m'entoure toujours d'amour et d'affection, qui m'ont souhaités et me souhaitent tout le succès dans mes études.

Personnes qui ont sacrifiés leur vie pour que je sois. Je leurs exprime toute ma gratitude et ma profonde affection, que dieu les gardes, les protèges et les bénits.

Au nom de l'amour, du respect et de l'amitié, je dédie cette mémoire à toutes les personnes

que j'aime, que j'admire, que j'estime et que je respecte :

A mes frères et ma sœur

A ma famille

A mes camarades de la promotion

A tous mes ami(e)s.

*Mes professeurs, et surtout mon encadreur **Mr. MECHEKOUR Elhadi***

A tous ceux qui m'ont aidé de près ou de loin à réaliser ce projet

Hadjaz Karim

Dédicaces

Grâce à dieu le tout puissant, j'ai pu réaliser ce modeste travail en ce jour solennel, je dédie ce mémoire spécialement :

A mes parents qui m'ont entouré et m'entoure toujours d'amour et d'affection, qui m'ont souhaités et me souhaitent tout le succès dans mes études. Personnes qui ont sacrifiés leur vie pour que je sois. Je leurs exprime toute ma gratitude et ma profonde affection, que dieu les gardes, les protèges et les bénits.

Au nom de l'amour, du respect et de l'amitié, je dédie cette mémoire à toutes les personnes que j'aime, que j'estime et que je respecte :

Ma sœur et mes frères

A mon Binôme **Hadjaz Karim**

A tous les étudiants de la promotion 2^{ème} année master productique

Mes professeurs, et surtout mon encadreur **Mr : Mechekour Elhadi**

A tous ceux qui m'ont aidé de près ou de loin à réaliser ce mémoire.

Boumaiza Mohamed Tayeb

REMERCIEMENTS

Nous remercions dieu tout puissant de nous avoir donné le courage, la volonté, la santé et la patience pour accomplir ce présent mémoire.

Un très grand remerciement a notre encadreur, pour encouragements inlassables, son amabilité et gentillesse méritent toute admiration.

Nos remerciements concertent aussi bien nos professeurs, pour l'aide qu'ils ont manifestés à notre égard, que tous les gens de département de Génie Mécanique pour leurs aides précieux et leurs disponibilités.

On ne peut pas débiter ce remerciement sans remercier nos parents, pour nous avoir donné toutes les moyens et laissé la disponibilité nécessaire à la rédaction de ce Projet Fin d'Etude.

Liste des figures

Figure I.1. Modèle fil de fer	6
Figure I.2. Surface plane	8
Figure I.3. Surface cylindrique	8
Figure I.4. Surface conique	9
Figure I.5. Surface torique (avec sphère)	9
Figure I.6. Points de contrôles S0, S1, S2, S3 et S4, avec courbe de Bézier associée (m=4)	10
Figure I.7. Surface de Bézier	11
Figure I.8. Courbe B-Spline	12
Figure I.9. Surface de type B-Spline	13
Figure I.10. Représentation d'une courbe NURBS	14
Figure I.11. Surface NURBS	14
Figure I.12. Modélisation par occupation spatiale	15
Figure I.13. Représentation d'un objet à l'aide B-Rep	16
Figure I.14. Représentation d'un objet à l'aide d'un arbre CSG	18
Figure I.15. Relation entre modèle topologique et géométrique	20
Figure I.16. Exemple de modèle CAO en B-Rep. À gauche présente la géométrie du solide. À droite présente une partie du treillis décrivant la topologie du solide.	21
Figure I.17. Topologie de solide fixé et la géométrie défférente	21
Figure I.18. Modification des coordonnées d'un point	22
Figure I.19. Modification de l'équation d'un cercle	22
Figure I.20. Pièces A et C : même géométrie ; Pièces B et C : Même topologie	22
Figure I.21. Exemple d'une pièce au format IGES	23
Figure I.22. Exemple d'une pièce au format STEP	25
Figure II.1. Interface du logiciel « FreeCAD »	29

Figure II.2. Les étapes de création d'un cube	31
Figure II.3. Console Python sur FreeCAD	31
Figure II.4. Création d'un cube par console Python	32
Figure II.5. Création d'une sphère par console Python	33
Figure II.6. Interface du « Shell Python »	34
Figure II.7. Contenu d'un fichier IGES sous format texte	36
Figure III.1. Algorithme d'extraction des données géométriques et topologiques du fichier au format IGES	45
Figure III.2. Notre programme en Python	46
Figure III.3. Fenêtre personnalisation	52
Figure III.4. Onglet macro de la fenêtre personnaliser	53
Figure III.5. Ajouter Dossier d'icône	54
Figure III.6. Icône de la macro	55
Figure III.7. Le bouton de macro est créé	56
Figure III.8. Création barre d'outils du bouton.	57
Figure III.9. La macro et son icône apparaissent.....	58
Figure III.10. Création nouvelle barre d'outils	59
Figure III.11. Ajouter le bouton à barre d'outils.....	60
Figure III.12. L'affichage de bouton sur le menu	61
Figure III.13. Ouverture de la pièce (Pièce_1)	62
Figure III.14. Lancement de l'extraction des données (Pièce_1)	63
Figure III.15. L'affichage du résultat (Pièce_1)	64
Figure III.16. Ouverture de la pièce (Pièce_2)	66
Figure III.17. Lancement de l'extraction des données (Pièce_2)	67

Liste des abréviations

DAO	Dessin Assisté par Ordinateur
2/3 D	2/3 Dimension
CAO	Conception Assistée par Ordinateur
CAFO	Conception et Fabrication Assistée par Ordinateur
NURBS	Non Uniform Rational B-Spline
B-Rep	Boundary Représentation
CSG	Constructive Solid Géométrie
DFN	Définition de Formes Numérisée.
IGES	Initial Graphic Exchange Standard
STEP	STandard for the Exchange of Product model data
CASCADE	Computer Aided Software for Computer Aided Design and Engineering
GPL	General Pupliq Licence
LGPL	Limited General Pupliq Licence
IAO	Ingénierie Assistée par Ordinateur
PLM	Product Lifecycle Management

Sommaire

Introduction générale	2
------------------------------------	---

Chapitre I : Généralités sur le modèle 3D

I.1. Introduction.....	5
I.2. Définitions.....	5
I.2.1. Définition de la DAO	5
I.2.2. Définition de la CAO	5
I.2.3. Définition de la CFAO.....	6
I.3. Modèles de représentation géométrique en CAO	6
I.3.1. La modélisation filaire (fil de fer).....	6
I.3.2. La modélisation surfacique	7
I.3.2.1. Les surfaces simples	8
I.3.2.2. Les surfaces complexes.....	9
I.3.3. Modélisation volumique (solide).....	15
I.3.3.1. Modélisation par occupation spatiale (énumération spatiale).....	15
I.3.3.2. Modélisation par Boundary Représentation (modèle B-Rep)	16
I.3.3.3. Modèles volumiques CSG	18
I.3.3.4. Modèle hybride	19
I.4. Modèle de représentation topologique en CAO.....	19
I.4.1. Les informations géométriques et topologiques au sein d'un modèle (B-Rep)	20
I.4.2. Indépendance géométrie – topologie	21
I.4.3. Dépendance géométrie – topologie	21
I.4.4. Complémentarité géométrie – topologie	22
I.5. Format d'échange des données en CAO	23
I.5.1. Format IGES (extension .igs ou .iges)	23
I.5.2. Format STEP (extension .stp ou .step).....	24

Chapitre II : Méthodologie de travail

II.1. Introduction	27
II.2. Objectif de notre travail	27
II.3. Méthodologie de travail.....	27

II.4. Présentation du logiciel de modélisation « FreeCAD »	28
II.4.1. Définition.....	28
II.4.2. Manipulation d’objets sur FreeCAD	29
II.5. Présentation du langage de programmation « Python »	34
II.5.1. Définition.....	34
II.5.2. Caractéristique	34
II.5.3. Exemples d’utilisation de Python	35
II.6. Contenus détaillés du fichier au format IGES	35

Chapitre III : Développement de l’Application

III.1. Introduction	42
III.2. Algorithme d’extraction des données	42
III.3. Programmation en Python	45
III.4. Intégration de notre programme dans le logiciel FreeCAD	46
III.4.1. Programme de la macro.....	47
III.4.2. Personnaliser la barre d’outil (ajouté un bouton)	51
Conclusion générale	69

Références bibliographiques

Annexes

Résumés

Introduction générale

Introduction générale

La conception d'un produit mécanique est une activité dynamique qui a pour objectif de répondre à des besoins spécifiés par un cahier des charges fonctionnel en termes de solutions technologiques et techniques. C'est une activité aux multiples facettes qui ne se borne pas uniquement à la définition géométrique d'un produit, mais qui interfère également avec les processus associés. Ainsi, la définition d'un produit mécanique nécessite l'intervention de différents métiers (Modélisation géométrique, Analyse, Fabrication, ...) qui permettront d'aboutir à une description globale et cohérente de celui-ci. Les interactions entre métiers qui étaient jusqu'à présent encore très procédurales et séquentielles, c'est-à-dire, qui se bornaient à suivre une organisation et une planification de tâches relativement figées, tendent à devenir plus coopératives et dynamiques.

Aujourd'hui, la Conception et Fabrication Assistées par Ordinateur (CFAO) occupe une place fondamentale au sein des entreprises industrielles, pour un meilleur compromis coût - délai - qualité, afin d'assurer une meilleure interopérabilité entre les différents métiers. Il y a de nombreux systèmes CFAO existant actuellement sur le marché. Généralement, ces logiciels étaient à la base des logiciels de CAO et qui ont évolué pour intégrer la FAO. Par contre, les logiciels dédiés uniquement à la FAO sont moins nombreux.

La fonction (CAO) c'est l'ensemble des outils logiciels et des techniques informatiques qui permettent d'assister les concepteurs dans la conception et la mise au point d'un produit. La fonction FAO est de générer le programme de Commande Numérique (CN) à partir de la géométrie de la pièce fournie par la fonction CAO du Bureau d'Etude (BE). Celui-ci fait partie intégrante du travail de la gamme au sein du Bureau de Méthodes (BDM.).

Dans notre mémoire, l'objectif était de développer un programme capable d'extraire des données (géométriques et topologiques) d'un modèle 3D issu d'un fichier au format standard IGES afin de l'intégrer dans un logiciel de modélisation CAO qui est FreeCAD. Notre travail permettra à l'ingénieur du bureau de méthode d'avoir un outil d'aide à la conception capable de proposer une liste d'informations de types géométriques ou topologiques (nombre de surfaces planes, cylindrique, ..., arêtes rectilignes...) en parallèle à la représentation 3D de la pièce, afin qu'il puisse déterminer les opérations d'usinage ou les machines nécessaires à la réalisation de la pièce. Ceci représente aussi la première étape vers le développement d'une application de génération de gamme d'usinage automatique.

Notre mémoire est organisé comme suit :

- Le premier chapitre (recherche bibliographique), nous allons parler dans la première partie sur l'aspect CAO en détaillant des parties concernant la modélisation géométrique ainsi que les méthodes de modélisation qui influencent directement les informations contenues dans le modèle 3D. Dans la deuxième partie du chapitre, nous allons voir l'échange des données CAO dans différents fichiers, en parlant sur un standard d'échange comme IGES et STEP.
- Dans le deuxième chapitre nous allons voir en détail sur le format IGES. Et nous allons voir des notions générales sur logiciel de modélisation « FreeCAD », avec l'utilisation de la « console Python » et des notions sur la programmation par le langage « Python »,
- Dans le dernier chapitre nous présentons les résultats de notre travail avec deux exemples pratiques d'un modèle 3D avec l'interface de programmation du logiciel FreeCad, l'enregistreur de macros et le langage de programmation Python. Ce dernier peut permettre aux concepteurs de choisir des solutions technologiques et techniques alternatives par comparaison. Pour faciliter l'utilisation de notre programme nous l'avons intégré dans l'application FreeCAD par un bouton dédié.
- Enfin on termine notre mémoire par une conclusion générale où un bilan des travaux effectués est dressé. Des perspectives possibles relatives aux approches développées sont dégagées pour faire suite à cette étude.

Chapitre I :

Généralités sur le modèle 3D

I.1. Introduction

Nous allons vous présenter dans ce chapitre les notions de modélisation géométrique et topologique ainsi que les formats de fichiers standards utilisés pour les échanges de données du modèle 3D des pièces en mécanique.

I.2. Définitions

I.2.1. Définition de la DAO :

Le dessin assisté par ordinateur DAO est un outil d'aide à la production de dessins de communication : " la transmission d'informations sous forme de plans".

Le dessin assisté par ordinateur (DAO) : est une discipline permettant de produire des dessins techniques avec un logiciel informatique. On le distingue de la synthèse d'image dans la mesure où il ne s'agit pas du calcul de rendu d'un modèle numérique mais de l'exécution de commandes graphiques (traits, formes diverses...). De ce fait, en DAO, la souris et le clavier remplacent le crayon et les autres instruments du dessinateur. Les dessins produits sont le plus souvent réalisés en mode vectoriel (traits cohérents). Les logiciels de DAO attribuent des coordonnées (X,Y pour les plans 2D et X,Y,Z pour les modèles 3D). Chaque élément d'un dessin est appelé entité et chaque entité contient donc des propriétés de couleur, d'épaisseur, de calque, de type de ligne, etc... [1].

I.2.2. Définition de la CAO :

Nous pouvons définir la Conception Assistée par Ordinateur (CAO) par l'ensemble des outils logiciels et des techniques informatiques qui permettent d'assister les concepteurs dans la conception et la mise au point d'un produit. La CAO offre une visibilité globale du comportement d'un objet avant qu'il n'existe, tant au niveau de son aspect que de sa structure et de son fonctionnement (sans l'avoir fabriqué réellement) [2].

I.2.3. Définition de la CFAO :

La Conception et Fabrication Assistée par Ordinateur (CFAO) est un procédé qui permet à l'aide d'un système automatisé de concevoir, programmer les formes de la pièce à usiner et de réaliser les différentes opérations d'usinage sans interrompre la chaîne de transmission des informations. Les machines utilisées en C.F.A.O sont des machines à commande numérique [3].

I.3. Modèles de représentation géométrique en CAO

Emanent originellement du domaine de la CAO /DAO. Un modèle géométrique est la représentation numérique d'un objet spatial. Il existe trois types d'approches de modélisation tridimensionnelle des objets spatiaux : la modélisation filaire (fil de fer), la modélisation surfacique, et la modélisation solide ou volumique [4].

En plus de trois modèles cités précédemment, il y a aussi la modélisation par les entités d'usinage (rainure, congé, trou assistée...). Ces différents modes de représentation ou modélisation d'un objet permettant de le définir plus ou moins complètement et présentent un certain nombre d'avantages et d'inconvénients. Dans ce qui suit les trois principaux modèles de représentation sont détaillés.

I.3.1. La modélisation filaire (fil de fer)

C'est la première modélisation d'un objet en 3D. La géométrie est défini uniquement à l'aide d'entités géométriques de bas niveau (sommet et arête) : droites, cercles, arcs de cercle, courbes, etc. (voir figure I.1). Un modèle fil de fer est bon pour la visualisation notamment dans le domaine de l'automobile [2].

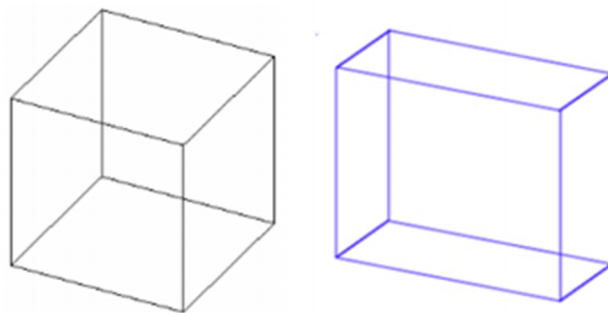


Figure I.1. Modèle fil de fer.

➤ **Avantage :**

- Structure de donnée ;
- Visualisation très rapide ;
- Nécessite un minimum de puissance de calcul ;
- simple et compacte.

➤ **Inconvénients :**

- Ambigu (on ne distingue pas le vide du plein) ;
- Solide inconsistant ;
- Possibilité de créer des solides sans aucun sens physique ;
- Impossibilité d'effectuer certains calculs (calcul de masse par exemple) ;
- Difficulté de résolution du problème de l'élimination des parties cachées [5].

I.3.2. La modélisation surfacique

C'est la deuxième technique apparue sur le marché mais c'est en fait la première qui a permis une approche 3D plus profonde que la représentation fil de fer. Dans ce modèle on distingue 2 types de surfaces : les surfaces simples et les surfaces complexes.

- ❖ Les surfaces simples : on trouve les plans, les surfaces d'extrusion et les surfaces de révolution (le cylindre, le cône, la sphère, le tore).
- ❖ Les surfaces complexes : on trouve trois grandes familles : Béziérs, B-Splines et NURBS (Non Uniform Rational B-Spline).

➤ **Avantage :**

- une description exacte de l'enveloppe de la pièce ;
- un usinage possible de la pièce en commande numérique ;
- une meilleure visualisation de la pièce par l'utilisation de techniques de rendu réaliste.

➤ **Inconvénients :**

- une technique difficile demandant un apprentissage long et une formation approfondie. En effet, il existe un grand nombre de surfaces différentes à utiliser et les modifications ne sont pas très aisées à réaliser ;
- une visualisation délicate si l'on n'utilise pas les techniques de rendu réaliste, qui demandent des calculs importants ;

- une nécessité de connaître parfaitement la pièce que l'on veut modéliser, les modifications étant assez difficiles à faire [2].

I.3.2.1. Les surfaces simples

Il existe principalement 5 types de surfaces simples ou classiques, qui sont : le plan, le cylindre, le cône, le tore et la sphère. Leurs équations sont données dans leurs repères locaux classiques à l'aide des paramètres u et v .

- **Le plan** : voir figure I.2

$S(u, v) = u\vec{x} + v\vec{y}$. Avec \vec{x} et \vec{y} sont 2 vecteurs d'unités du plan.

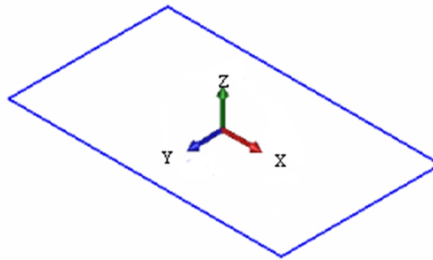


Figure I.2. Surface plane [2].

- **Le cylindre** : voir figure I.3

$S(u, v) = R * [\cos(u)\vec{e}_1 + \sin(u)\vec{e}_2] + v\vec{e}_3 + o$.Avec R est le rayon de cylindre

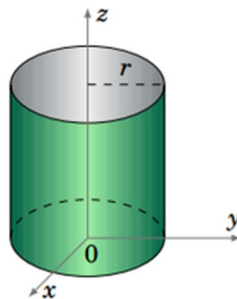


Figure I.3. Surface cylindrique [2].

- **Le cône (d'angle α)** : voir figure I.4

$S(u, v) = u * \tan(\alpha) * [\cos(u)\vec{x} + \sin(u)\vec{y}] + v\vec{z} + o$

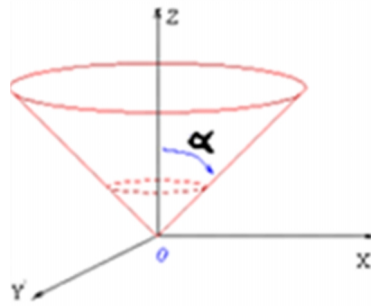


Figure I.4. Surface conique.

- **Le tore et la sphère :** voir figure I.5

$$S(u, v) = [R + r * \cos(v)] * [\cos(u) \vec{x} + \sin(u) \vec{y}] + r * \sin(v) \vec{z} + o$$

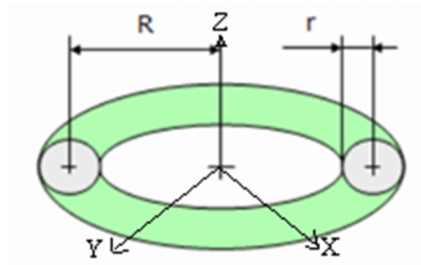


Figure I.5. Surface torique (avec sphère).

- Si $R = 0$, on retrouve l'équation d'une sphère.

I.3.2.2. Les surfaces complexes :

a) Modèle de Bézier :

Ils ont été définis dans les années 60. Une courbe de Bézier est totalement modifiée dès qu'on déplace un point de contrôle : on dit que la méthode de Bézier est une **méthode globale**.

a.1) La courbe de Bézier :

L'idée lumineuse qu'il a eu consiste à exprimer une courbe comme combinaison linéaire de points (appelés points de contrôles). Ceci signifie en particulier que la courbe est tangente en S_0 et S_4 aux droites S_0S_1 et S_2S_3 . Ceci est bien illustré sur la figure I.6 on-a des points de contrôle (ordonnés) et une courbe dont les extrémités sont deux points (ce courbe et qui est «attirée» par des autres points).

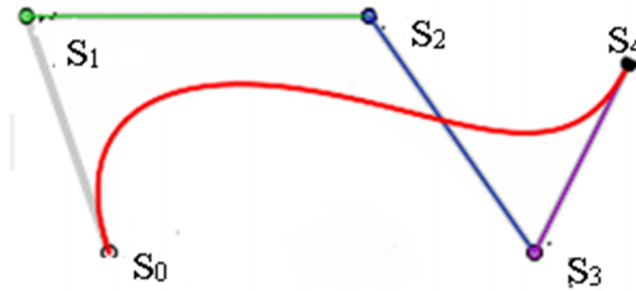


Figure I.6. Points de contrôles S0, S1, S2, S3 et S4, avec courbe de Bézier associée (m=4).

L'équation d'une courbe de Bézier :

$$P(u) = \sum_{i=0}^m B_i^m(u) S_i \quad , B_i^m = \frac{m!}{i!(m-i)!} u^i(1-u)^{m-i} , u \in [0,1] \text{ e } i \in \{0,1, \dots, m\}$$

- **B** : polynôme de Bernstein.
- **m** : degré de la courbe.

$$P(u) = \frac{m!}{i!(m-i)!} u^i(1-u)^{m-i} S_i$$

Exemple (figure précédent) :

L'équation « P (u) » est l'équation de courbe de Bézier :

$$P(u) = (1-u)^4 S_0 + 4u^2(1-u)^3 S_1 + 6u^2(1-u)^2 S_2 + 4u^3(1-u) S_3 + u^4 S_4$$

Si les coordonnées des points de contrôle sont :

$$S_0(x_0, y_0) ; S_1(x_1, y_1) ; S_2(x_2, y_2) ; S_3(x_3, y_3) \text{ e } S_4(x_4, y_4) \text{ donc :}$$

$$\begin{cases} x_c(u) = (1-u)^4 x_0 + 4u^2(1-u)^3 x_1 + 6u^2(1-u)^2 x_2 + 4u^3(1-u) x_3 + u^4 x_4 & p \quad 0 \leq u \leq 1 \\ y_c(u) = (1-u)^4 y_0 + 4u^2(1-u)^3 y_1 + 6u^2(1-u)^2 y_2 + 4u^3(1-u) y_3 + u^4 y_4 & p \quad 0 \leq u \leq 1 \end{cases}$$

a.2) La surface de Bézier : voir figure I.7

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) S_i \quad , \quad u, v \in [0,1]$$

$$i \in \{0,1, \dots, m\} , j \in \{0,1, \dots, n\}$$

- m : degré de la courbe dans le 1^e sens.
- n : degré de la courbe dans le 2^e sens.

$$\text{tel que } B_i^n = \frac{m!}{i!(m-i)!} u^i(1-u)^{m-i} \quad e \quad B_j^m = \frac{n!}{i!(n-i)!} v^i(1-v)^{n-i}$$

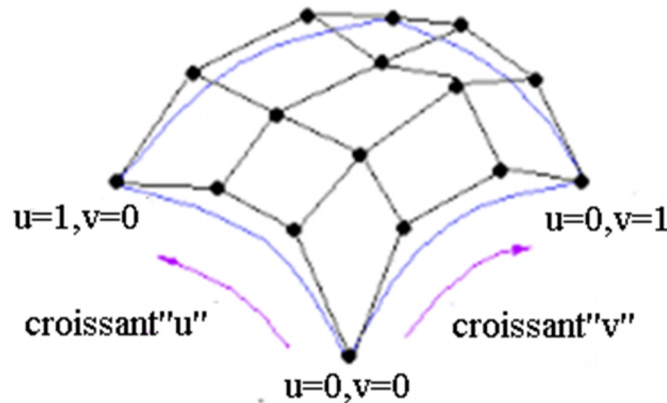


Figure I.7. Surface de Bézier.

➤ **Avantages des courbes et surfaces de Béziérs :**

- Chaque point de la courbe est une combinaison convexe de tous les points de contrôle.
- Pas de problème de non continuité aux raccords,
- Grande maniabilité des courbes.

➤ **Inconvénients de l'utilisation des courbes et surfaces de Béziérs :**

- le déplacement d'un point de contrôle implique une modification complète de la courbe.
- il est coûteux de rajouter un point.
- le degré du polynôme augmente avec le nombre de points de contrôle. Ainsi, lorsque le nombre de points de contrôle est important, le calcul des fonctions de Béziérs devient délicat [6].

b) Modèle B-Spline :

Ils ont été définis dans les années 70 pour remédier à l'inconvénient de la globalité des courbes et surfaces de Bézier.

Le déplacement d'un point de contrôle de la courbe n'affecte ainsi plus qu'une partie limitée de la courbe, ce qui amène un plus grand confort dans la conception assistée par ordinateur. La méthode B-Spline est donc une **méthode locale**.

b.1) L'équation de courbe B-Spline : voir figure I.8.

$$P(u) = \sum_{i=0}^n N_i^m(u) S_i, \quad n \geq m, u \in [u_0, u_{n+m+1}]$$

Où les polynômes N_i^m sont définis de manière récursive :

$$N_i^m(u) = \frac{u - u_i}{u_{i+m} - u_i} N_i^{m-1}(u), \quad u \leq u_{i+1}$$

$$N_i^m(u) = \frac{u - u_i}{u_{i+m} - u_i} N_i^{m-1}(u) + \frac{u_{i+m+1} - u}{u_{i+m+1} - u_{i+1}} N_{i+1}^{m-1}(u),$$

$$u_{i+1} < u < u_{i+m}$$

$$N_i^m(u) = \frac{u_{i+m+1} - u}{u_{i+m+1} - u_{i+1}} N_{i+1}^{m-1}(u), \quad u \geq u_{i+m}$$

Avec :

$$N_i^0(u) = \begin{cases} 1 & \text{si } u_i \leq u \leq u_{i+1} \\ 0 & \text{p } a \end{cases}$$

Où le vecteur de paramètres $\{u_0, u_1, \dots, u_{n-1}, u_n\}$ est le vecteur de nœud de la courbe.

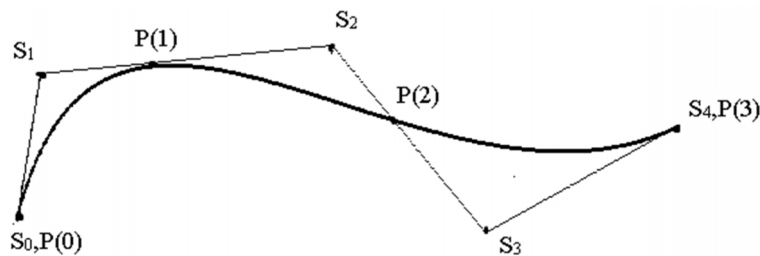


Figure I.8. Courbe B-Spline.

b.2) La surface B-Spline : voir figure I.9.

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^t N_i^m(u) N_j^r(v) S_i, \quad u \in [u_0, u_{n+m+1}], \quad v \in [v_0, v_{t+r+1}].$$

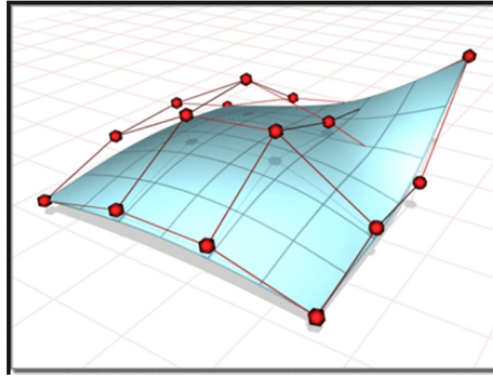


Figure I.9. Surface de type B-Spline.

➤ **Avantages des courbes et surfaces B-Spline :**

- leur capacité descriptive.
- les propriétés d'enveloppe convexe (Les points de contrôle de la courbe se situent toujours du même côté de la courbe).
- le contrôle local.
- le degré indépendant du nombre de points de contrôle.

➤ **Inconvénient des courbes et surfaces B-Spline :**

- le nombre de paramètres (un nombre énorme de paramètres dont la signification physique est parfois difficile à déterminer).

c) Modèle B-Spline rationnel (NURBS) :

Les courbes et surfaces B-Spline peuvent également être pondérées pour augmenter leur pouvoir descriptif ; on obtient alors les courbes et surfaces B-Spline rationnelles.

Les NURBS sont utilisées pour représenter mathématiquement des objets géométriques. Elles généralisent la représentation par les B-Splines des courbes et des surfaces en ajoutant un dénominateur. Elles sont en fait définies avec des points en coordonnées homogènes. Une B-Spline ressemble à une représentation polynomiale par morceaux, alors qu'une NURBS est une représentation par fractions rationnelles par morceaux.

c.1) L'équation de courbe B-Spline rationnelle (NURBS): voir figure I.10.

$$P(u) = \sum_{i=0}^n \frac{N_{i1}(u)S_iH_i}{\sum_{j=0}^n H_jN_j(u)} \quad , \quad u \in [u_0, u_{n+m+1}].$$

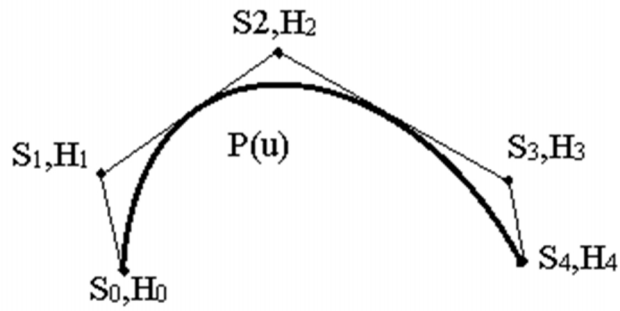


Figure I.10. Représentation d'une courbe NURBS.

c.2) L'équation de surface B-Spline rationnelle (NURBS): voir figure I.11.

$$P(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^t N_{i1}(u) N_j(v) S_i H_i}{\sum_{k=0}^n \sum_{l=0}^t N_k(u) N_l(v) H_k}$$

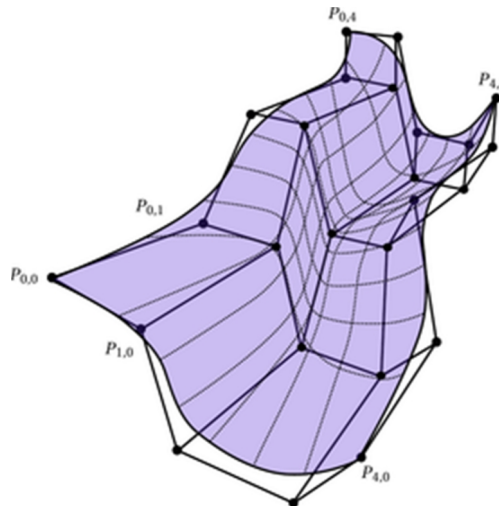


Figure I.11. Surface NURBS.

➤ **Avantage :**

- facilité et précision pour évaluer une forme.
- capacité pour approximer des formes complexes.
- simplicité de construction et d'implémentation.
- faible complexité des algorithmes utilisés.

Pour plus détaille sur les équations voir [7].

I.3.3. Modélisation volumique (solide) :

C'est la technique de représentation d'un objet la plus utilisée. Le solide se caractérise par son aspect homogène 3D, ses limites (le solide occupe un espace fini) et ses frontières qui définissent un intérieur et un extérieur au volume.

Dans les paragraphes suivants nous allons introduire les principaux modèles de solide à savoir : la représentation par énumération spatiale (occupation spatiale), la représentation par les frontières (B-Rep), la représentation CSG et la représentation hybride.

I.3.3.1. Modélisation par occupation spatiale (énumération spatiale) :

L'objet est représenté par un ensemble de cellules occupées par le volume de l'objet dans l'espace. Celles-ci peuvent être des volumes quelconques, mais l'usage de cubes de taille fixe est très fréquent (figure I.12). Ces cubes appelés Voxels sont des petits éléments de volume discrétisés. Ils sont localisés le plus souvent par les coordonnées d'un point 3D. Un objet est représenté par un tableau de triplets (x, y, z) appelé tableau spatial, habituellement ordonné et dont l'arbre correspond à l'ordre de construction de l'objet. Cette approche est simple à mettre en œuvre. Elle facilite le calcul des propriétés physiques d'un objet et la mise en œuvre des opérations booléennes (union, intersection et différence). Elle reste invariante en translation, mais ne l'est pas en rotation. Le principal inconvénient de la modélisation par occupation spatiale est qu'elle ne donne qu'une approximation plus ou moins grossière de l'objet, selon le niveau de résolution choisi [5].

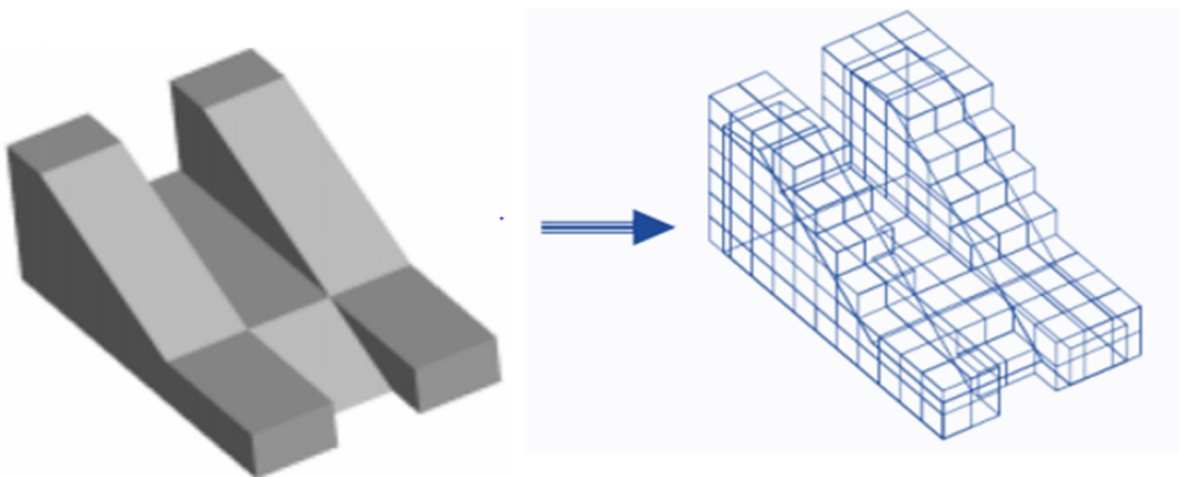


Figure I.12. Modélisation par occupation spatiale [5].

I.3.3.2. Modélisation par Boundary Représentation (modèle B-Rep) :

Le modèle B-Rep propose une décomposition d'un solide en faces qui sont elles-mêmes délimitées par des arêtes (droites ou courbes) bornées par des sommets (voir figure I.13). Un B-Rep bien formé ne peut s'entrecroiser, doit être fermé, orientable, borné et topologiquement connecté. Ces aspects, qui feront l'objet de la prochaine partie, sont pris en compte par un modèle topologique qui explicite les relations entre ces primitives géométriques.

Pour valider un modèle B-REP, chaque face doit vérifier un ensemble de propriétés, qui assurent la reconnaissance de l'objet en tant que volume occupant une portion de l'espace ainsi que son homogénéité. En fait le modèle B-REP décrit les frontières comme une juxtaposition de plusieurs faces orientées. Chaque face est composée d'une surface (entité qui décrit la géométrie de la face) et est bornée par au moins un contour fermé et orienté. Chaque contour est formé d'un ensemble d'arêtes. Chaque arête comporte une courbe porteuse et limité par deux sommets.

- a) Solide valide : pas d'arrêtes ni de faces pendantes et pas de faces qui se coupent (équation d'Euler), surface orientable (règle Moebius).
- b) Représentation par frontière : modélisation d'un volume par un ensemble de surfaces (polygonaux).
- c) Frontière : ensemble de faces, reliées par des arêtes qui se rencontrent à des sommets.
- d) Attributs géométrique : surface, courbe (arrêtes) et point [2].

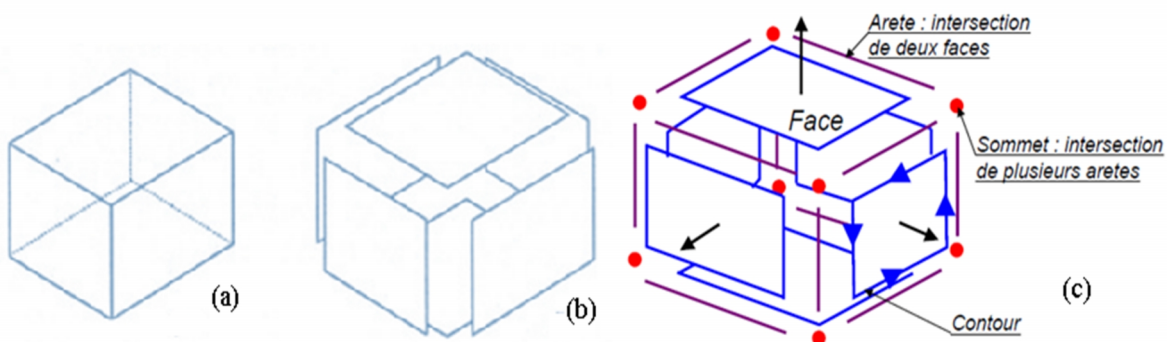


Figure I.13. Représentation d'un objet à l'aide B-Rep [2].

Validité d'un modèle B-Rep :

La complexité de prototypage d'un modèle B-Rep peut croître très rapidement avec le nombre d'entités géométriques mises en œuvre. Aussi, des méthodes permettant de vérifier la validité d'un modèle sont indispensables pour assurer sa cohérence. Appliquée à un modèle B-Rep, la formule d'Euler (3.1) permet de vérifier la cohérence du nombre d'entités géométriques de dimensions différentes qui composent un solide [8] :

$$F - A + S = 2(C - H) + R$$

Avec:

- F : le nombre de faces.
- A : le nombre d'arêtes (courbes de bords).
- S : le nombre de sommets (de nœuds).
- C : le nombre de composantes connexes.
- H : le nombre d'anses.
- R : le nombre de trous dans les faces.

Exemple :

Pour un cube on a : $F = 6$, $A = 12$, $S = 8$, $C = 1$, $H = 0$, $R = 0$.

Donc :

$$\begin{cases} F - A + S = 2 \\ 2(C - H) + R = 2 \end{cases}$$

➤ **Avantage:**

- ce modèle est facile en description, visualisation et transformation géométrique ;
- il est aussi unique (Pour une pièce mécanique, il existe un seul modèle B-REP).

➤ **Inconvénients:**

- difficultés de réalisation des opérations logiques (dues au grand nombre de faces constituant l'objet initial) ;
- absence d'algorithmes généraux (La plupart des algorithmes associés à ce modèle traitent des volumes dont les frontières sont composées de facettes planes, elles-mêmes délimitées par des segments de droite pour faciliter les opérations logiques).

I.3.3.3. Modèles volumiques CSG :

La Constructive Solid Géométrie (CSG) est un des modèles constructifs les plus utilisés. Il assemble, à l'aide d'opérateurs booléens régularisés des primitives volumiques (Figure I.14) comme des sphères, des cubes, des tores, etc. Ce type de modélisation est associé à un arbre binaire dont :

- les nœuds internes représentent des opérations booléennes ou des transformations géométriques (rotations, translations, mise à l'échelle);
- les nœuds terminaux correspondent aux primitives solides.

L'arbre d'un modèle CSG laisse la possibilité d'instancier les primitives solides (en fonction de paramètres différents dont la position, la taille, l'orientation) et de les grouper hiérarchiquement [4].

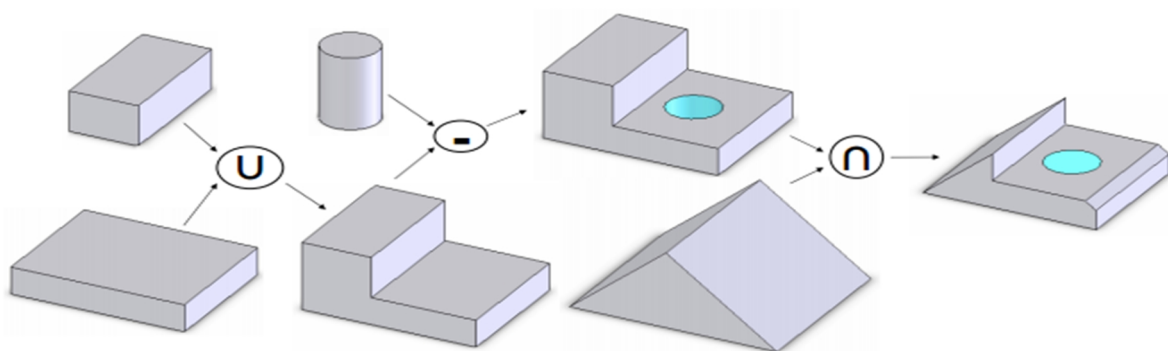


Figure I.14. Représentation d'un objet à l'aide d'un arbre CSG.

(L'ensemble de ces opérations est représenté par un arbre illustré, ici, par des flèches)

➤ **Avantage :**

- facilité de création
- facilité de réalisation des transformations géométriques.
- facilité de réalisation des opérations booléennes.
- Validité.
- Concision.

➤ **Inconvénients :**

- difficultés de calcul des propriétés géométriques et physiques d'un objet complexe.
- difficultés pour effectuer certaines conversions (arbre CSG vers le modèle de représentation par les frontières par exemple).
- non unicité : une pièce mécanique possède plusieurs représentations CSG.

I.3.3.4. Modèle hybride :

Le modèle hybride permet de cumuler les avantages de la représentation BREP et CSG en limitant au mieux leurs inconvénients respectifs. Dans ce modeleur, les primitives de base sont, soit celles du modèle CSG (cône, pyramide, cube,...), soit des solides définis par leurs frontières. Lors de chaque opération sur l'arbre de construction, les éléments frontières (représentant la peau de l'objet) sont construits. C'est le modèle le plus souvent utilisé actuellement.

➤ **Avantage :**

- le modèle Hybride permet de construire aussi simplement un solide qu'avec un arbre CSG et même mieux puisque la création de primitives BREP est possible ;
- le modèle Hybride permet facilement d'effectuer une visualisation rapide du solide ;
- le modèle Hybride permet d'effectuer aisément des opérations complexes comme le calcul du volume d'un solide, l'appartenance d'un point au solide etc. ;
- le Hybride permet de disposer de la définition exact de la surface du solide ;
- le modèle Hybride offre la possibilité de placement de contraintes géométriques et de contraintes d'assemblages.

➤ **Inconvénients :**

- il est nécessaire de stocker une très grande quantité d'informations ;
- il est nécessaire de maintenir en permanence la cohérence entre les deux représentations de l'information.

I.4. Modèle de représentation topologique en CAO

La topologie est une branche des mathématiques traitant des relations de voisinage qui s'établissent entre des figures géométriques, et qui ne sont pas altérées par la déformation des figures. L'application des principes de la topologie en CAO assure la cohérence des données

géométriques dans la représentation par les frontières ainsi que l'élimination de la redondance des entités dans la représentation par les frontières.

I.4.1. Les informations géométriques et topologiques au sein d'un modèle (B-Rep) :

Les modèles de CAO se basent dans la plupart de temps sur le modèle B-REP. Ce modèle est utilisé pour la reconstruction de modèle CAO en vue de garantir la comptabilité avec la plupart des logiciels de CAO [2].

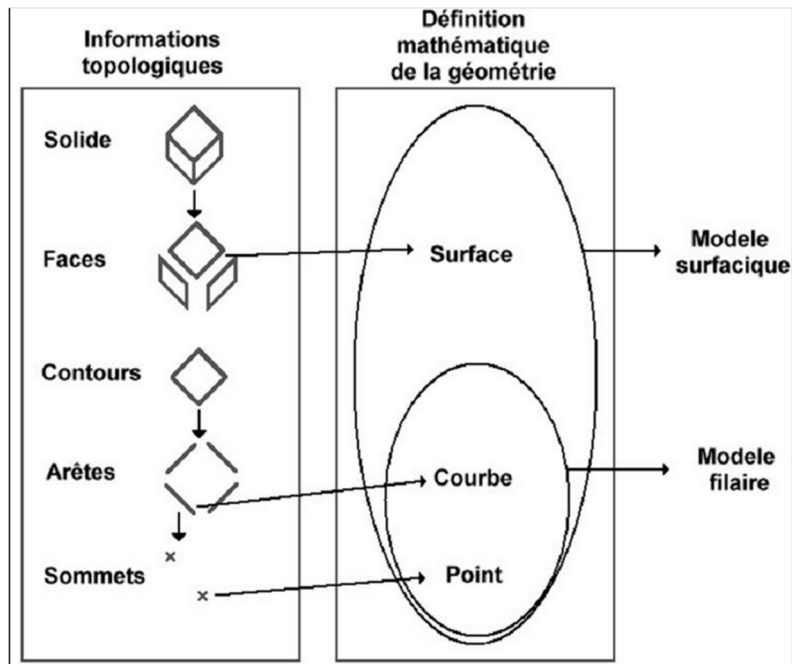


Figure I.15. Relation entre modèle topologique et géométrique [S1].

Dans ce qui suit nous allons présenter une caractéristique détaillée de modèle B-Rep. Un modèle B-REP est constitué d'informations topologiques et géométriques (voir figure I.15 et figure I.16) :

- les informations topologiques existantes au sein d'un modèle B-REP sont les connectivités, les liens et les orientations des différentes entités (Faces, contours, arêtes et sommets).
- les informations géométriques du modèle B-REP se résument dans la nature géométrique des surfaces et courbes porteuses des différentes faces et arêtes (face plane, courbe circulaire...).

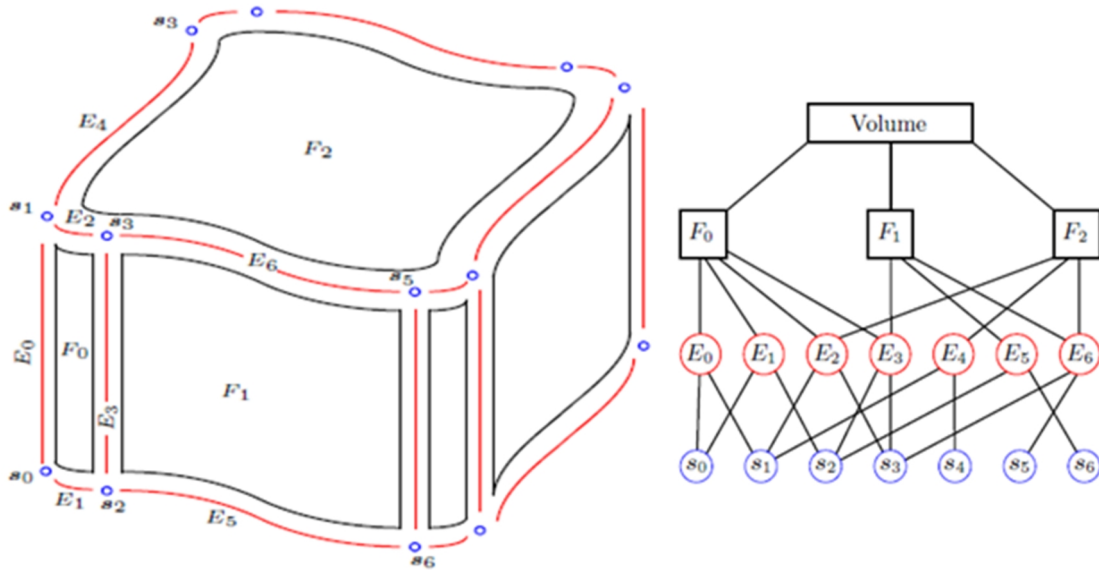


Figure I.16. Exemple de modèle CAO en B-Rep. À gauche présente la géométrie du solide. À droite présente une partie du treillis décrivant la topologie du solide.

I.4.2. Indépendance géométrie – topologie :

Une matrice de transformation n'impacte que la géométrie, la topologie demeure inchangée. Exemples: Rotation, translation d'un objet (voir figure I.17) :

- Les coordonnées des points, les équations des courbes et des surfaces changent.
- Les relations entre les faces, arêtes, sommets restent inchangés.

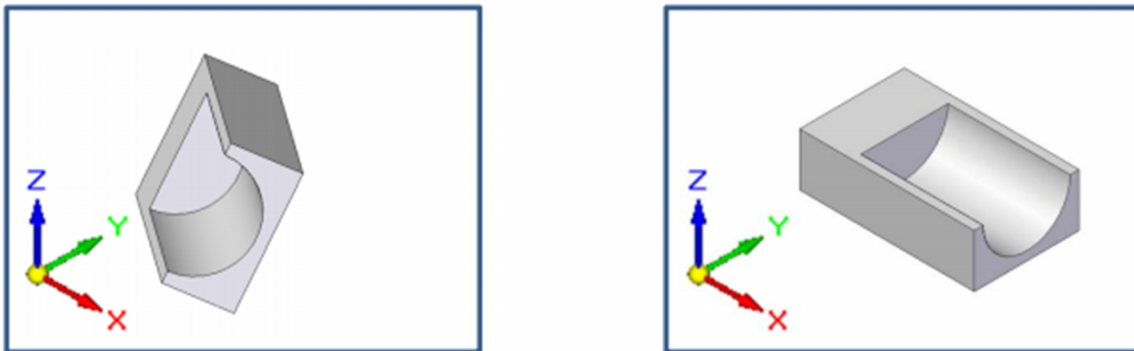


Figure I.17. Topologie de solide fixé et la géométrie défférente.

I.4.3. Dépendance géométrie – topologie :

Dans certains cas, la modification de la géométrie influence directement la topologie. Exemples suivants de la figure I.18 et I.19.



Figure I.18. Modification des coordonnées d'un point.

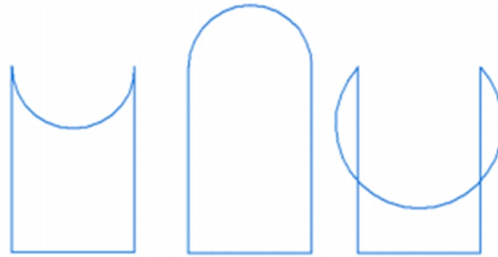


Figure I.19. Modification de l'équation d'un cercle.

I.4.4. Complémentarité géométrie – topologie :

Suffisance: Géométrie + Topologie (Propriété d'une représentation géométrique qui contient assez d'information pour distinguer un objet de tous les autres du domaine). Voir la figure I.20 [8].

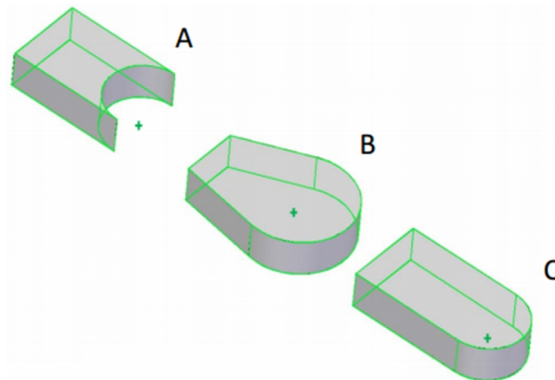


Figure I.20. Pièces A et C : même géométrie ; Pièces B et C : Même topologie.

I.5. Format d'échange des données en CAO

La conception de la pièce à fabriquer est réalisée à l'aide d'un logiciel de Conception assistée par ordinateur (CAO) : on nomme le fichier ainsi obtenu "modélisation 3D" ou encore "DFN" pour Définition de Formes Numérisée.

Cette modélisation en trois dimensions de la pièce à réaliser est ensuite « exportée », c'est-à-dire sauvée depuis la CAO dans un fichier intermédiaire en utilisant un standard d'échange comme IGES, STEP ...

I.5.1. Format IGES (extension .igs ou .iges)

IGES (Initial Graphic Exchange Standard) ; Ce format, malgré ses limitations, est l'un des plus répandus pour la visualisation ou l'export de modèles CAO 3D (voir figure I.21). Il représente des modèles filaires ou surfaciques. Si votre modèle est solide, l'export vers un format IGES ne permettra de sauvegarder que la "peau" extérieure de votre solide. Pour récupérer un solide, il vous faudra alors utiliser une fonction de votre modèleur permettant la génération d'un solide à partir de sa frontière surfacique. La plupart des modèleurs CAO permet l'import de géométrie IGES [S2].



Figure I.21. Exemple d'une pièce au format IGES

Les fichiers IGES comprennent cinq sections :

- Section de départ : en-tête de début comprend de l'information utile à la personne recevant ou exploitant le fichier.
- Section globale : comprend 24 champs nécessaires à la traduction et à l'utilisation du fichier.
- Description de la section Répertoire

- Générée par le préprocesseur IGES
- Décrit les entités identifiées par des codes et des sous codes.
- Contient des pointeurs vers les données de la section suivante.
- Contient des informations géométriques et non géométriques.
- Comprend 20 champs de 8 caractères en 2 lignes de 80 caractères.
- Description de la section des paramètres
- Comprend les données numériques (coordonnées, dimensions, nombre de points, etc.) associées aux éléments géométriques identifiés à la section précédente.
- Comprend des pointeurs vers les entités de la section précédente.
- Description de la section de fin : marque la fin du fichier et comprend le sous-total des enregistrements a des fins de vérification de la transmission de données.
- Types d'entités géométriques
- Types d'entités non géométriques [9]

I.5.2. Format STEP (extension .stp ou .step)

STEP (STandard for the Exchange of Product model data) ; STEP représente la volonté de la communauté des utilisateurs des logiciels de CAO à avoir un format complet sous tous les angles pour l'échange de données autour du modèle 3D solide (voir figure I.22). Il tend à s'imposer comme le standard d'échange entre les principales plates-formes de CAO. Néanmoins son interprétation peut varier d'un éditeur à l'autre, ce qui entraîne parfois erreurs ou pertes d'informations sur les modèles. De plus, la géométrie solide est récupérée en tant que bloc non éditable (l'arbre des spécifications n'est pas transmis) mais peut servir de base à de nouvelles fonctions (on peut réaliser un congé, se servir du contour frontière d'une face etc.). Pour les assemblages, STEP encapsule l'ensemble des modèles pièces et produits dans un unique fichier, ce qui facilite grandement les échanges.

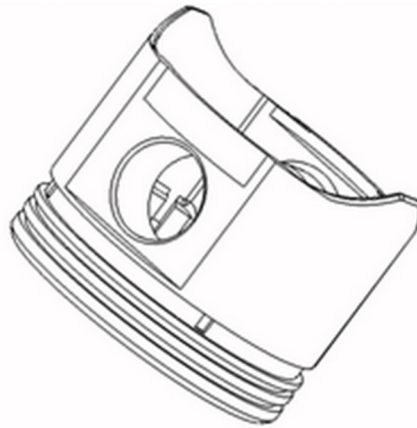


Figure I.22. Exemple d'une pièce au format STEP

Subdivision du format en 8 classes :

- Introduction (Part 1-9): Sommaire et principes généraux.
- Méthodes de description (Part 11-19): Présentation du langage EXPRESS.
- Méthodes d'implantation (Part 21-29): Comment EXPRESS est traduit sous forme de fichiers et autres mécanismes d'archivage.
- Cadre et méthodologie de test et conformité (Part 31-39): Méthode de test permettant la validation de l'implantation.
- Ressources intégrées (Part 41-99 et 101-199)
 - Représentation de la géométrie et de la structure du produit (Part 41-99);
 - Applications d'usage très général telles que le dessin et les éléments finis (Part 101-199)
- Protocoles d'application (Part 201-299): Implémentation de STEP dans le cadre d'applications industrielles spécifiques.
- Résumé des tests (Part 301-399): Associe des procédures de tests aux applications spécifiques.
- Interprétation de la construction des applications (Part 501-): Description de diverses méthodes de construction d'entités et de méthodes de modélisation spécifique [10].

Chapitre II :

Méthodologie de travail

II.1. Introduction

Nous avons vu dans le chapitre précédent des généralités sur le domaine de modélisation 3D en CAO. Maintenant, nous allons vous présenter dans ce chapitre notre démarche pour extraire les données géométriques et topologiques d'une pièce 3D ainsi que les outils qui nous ont permis de réaliser notre travail.

II.2. Objectif de notre travail

L'objectif de notre travail est de développer une application à partir de la technologie OpenCASCADE [S3] [S4], qui est une technologie libre (Open Source) proposant toutes les fonctionnalités d'un modéleur volumique de type CAO, qui nous permet d'extraire les données géométriques et volumiques d'un modèle 3D d'une pièce à partir de son fichier au format IGES.

Notre travail permettra à l'ingénieur du bureau de méthode d'avoir un outil d'aide à la conception capable de proposer une liste d'informations de types géométriques ou topologiques (nombre de surfaces planes, cylindrique,..., arêtes rectilignes...) intégrée au logiciel de modélisation et en parallèle à la représentation 3D de la pièce, afin qu'il puisse déterminer les opérations d'usinage ou les machines nécessaires à la réalisation de la pièce. Ceci représente aussi la première étape vers le développement d'une application de génération de gamme d'usinage automatique.

II.3. Méthodologie de travail

Afin d'atteindre notre objectif cité dans le paragraphe précédent, nous avons fait le choix de nous baser sur le logiciel de modélisation « FreeCAD » qui est développé à partir de la technologie OpenCASCADE, et qui permet plus facilement l'exploitation des fonctionnalités avancées de la modélisation grâce aux nombreuses fonctionnalités qui possède.

Etant donné que le logiciel « FreeCAD » est développé avec le langage de programmation « Python », nous avons été obligé d'apprendre ce langage afin d'intégrer notre programme dans le logiciel « FreeCAD » et pouvoir bénéficier de toutes ses ressources.

Dans notre démarche on considère que l'utilisateur a déjà ouvert le logiciel de modélisation et qu'il a une pièce active sauvegardée au format standard IGES. A partir de là, notre application une fois exécutée va :

- récupérer le nom du fichier ouvert ;
- localiser le fichier au format IGES de la pièce ;
- ouvrir le fichier et le charger dans la mémoire pour le traitement ;
- lire chaque ligne du contenu textuel du fichier IGES et extraire les données géométriques et topologiques ;
- incrémenter des compteurs pour chaque entités ciblés ;
- afficher le résultat sous forme d'une liste dans une fenêtre dans le logiciel ;
- sauvegarder le résultat dans un fichier texte pour une exploitation ultérieure.

Pour aider le lecteur à comprendre la suite de notre travail, nous allons vous présenter le logiciel de modélisation « FreeCAD » ainsi que le langage de programmation « Python ». Et pour finir, nous allons détailler le contenu textuel du fichier au format IGES que nous avons présenté succinctement dans le chapitre précédent, afin de localiser les informations géométriques et topologiques qui nous intéressent.

II.4. Présentation du logiciel de modélisation « FreeCAD »

II.4.1. Définition :

FreeCAD est un modèleur 3D (voir figure II.1) d'application générale à base de technologies OpenCASCADE. Le développement est entièrement Open Source (licences GPL et LGPL). FreeCAD est orienté vers le génie mécanique et la conception de produits, mais vise également d'autres disciplines, telles que l'architecture ou d'autres champs d'activité d'ingénierie.

FreeCAD propose des fonctionnalités similaires à « Catia », « SolidWorks » ou « SolidEdge », et s'inscrit donc aussi dans les catégories CFAO/IAO/PLM. Il vise la conception paramétrique, avec une architecture modulaire qui permet l'ajout facile de fonctionnalités supplémentaires sans modifier le cœur du programme [S5].

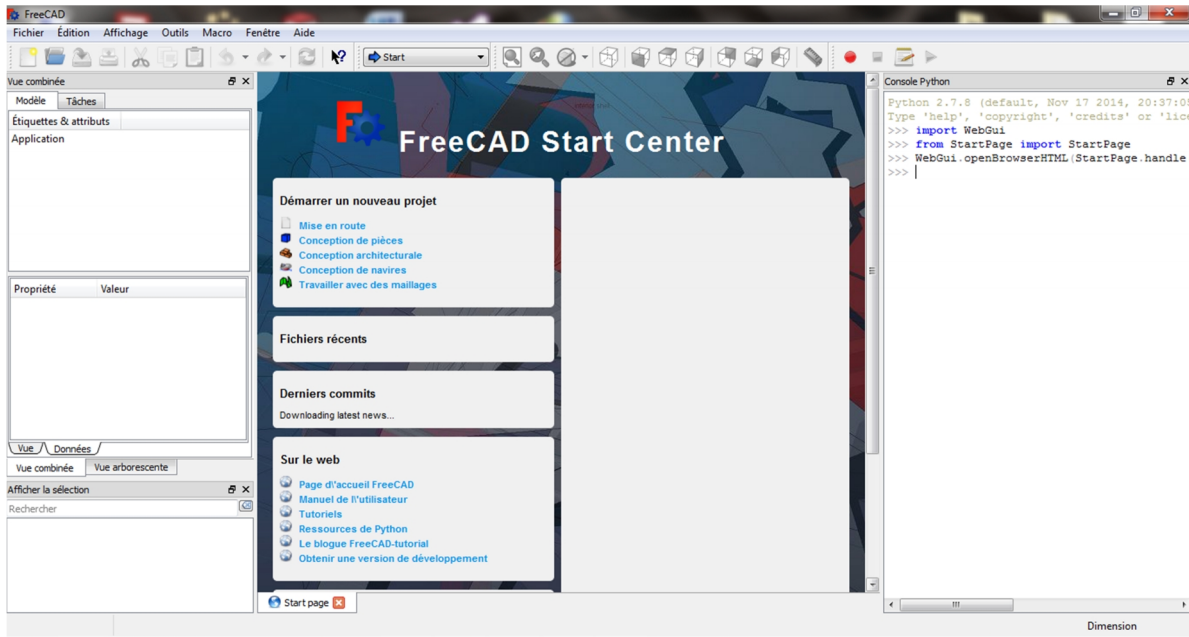


Figure II.1. Interface du logiciel « FreeCAD 0.16 ».

Caractéristique :

- Libre (Open Source) et complètement modulaire, permettant ainsi une personnalisation.
- Un modeleur 3D à conception paramétrique qui permet d'éditer facilement votre design en remontant dans l'historique du modèle afin d'en changer les paramètres.
- Il ouvre et sauvegarde plusieurs formats de fichier tels que STEP, IGES et d'autres.






II.4.2. Manipulation d'objets sur FreeCAD :

Les capacités CAO de FreeCAD sont basées sur le noyau OpenCASCADE. L'établissement de la pièce de FreeCAD permet d'accéder et d'utiliser les objets et les fonctions OpenCASCADE. OpenCASCADE 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.

a) Les outils :

Les outils de l'établi pièce sont situés dans le menu « Pièce » (ou Part Design) qui apparaît lorsque l'on charge l'établi pièce [S6], [11].

Voici quelques outils permettant de créer des primitives :

-  **Cube:** Dessine un cube en spécifiant ses dimensions.
-  **Cône:** Dessine un cône en spécifiant ses dimensions.
-  **Cylindre:** Dessine un cylindre en spécifiant ses dimensions.
-  **Sphère:** Dessine une sphère en spécifiant ses dimensions.
-  **Tore:** Dessine un tore (anneau) en spécifiant ses dimensions.

Exemple :

Pour créer un cube sur FreeCAD (voir figure II.2), on suit les étapes suivantes :

- 1) Créer un nouveau document vide.
- 2) Sous "Basculer entre les ateliers" choisir « Part Design ».
- 3) Clic sur « Cube », le cube est dessiné automatiquement.
- 4) Clic sur « Modèle » et sélectionner « Cube ».
- 5) Sous propriété régler longueur, largeur et hauteur.

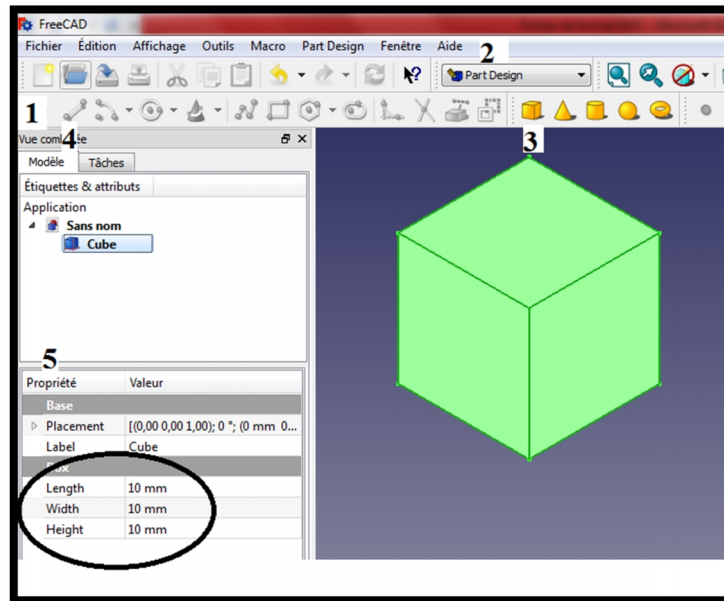


Figure II.2. Les étapes de création d'un cube.

b) Console Python : voir figure II.3.

FreeCAD offre également la possibilité incroyable d'écrire des scripts d'objets 100% Python. Ces "objets" se comporteront exactement comme n'importe quels autres objets (exemple du Cube) dans FreeCAD, et sont, sauvegardés et restaurés automatiquement dans le répertoire de chargement/sauvegarde [S6].

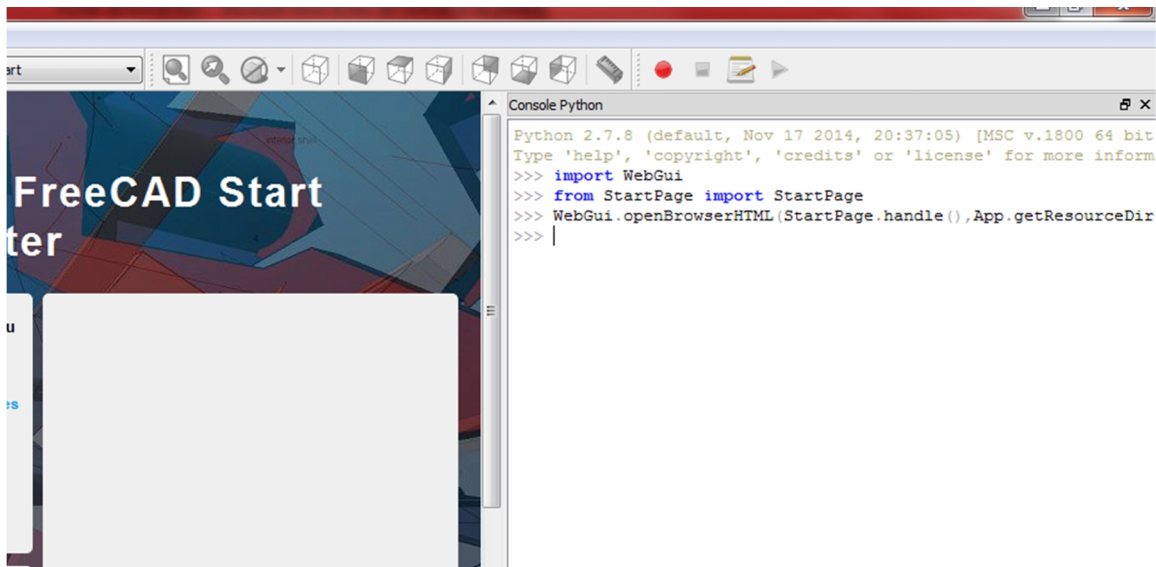


Figure II.3. Console Python sur FreeCAD.

c) Utilisation du langage « Python » dans « FreeCAD » :

Nous allons vous présenter la manipulation des objets liés au modèle solide par la « console Python » est cela à travers deux (02) méthodes.

c.1) Méthode 1 (exemple avec un cube) : voir figure II.4.

- Pour créer un nouveau document vide, et sou console Python taper la commande ci-dessous:

Doc = FreeCAD.newDocument()

- Ecrire la commande « **import Part** »
- Pour créer un objet, nous utilisons la commande « **make X ()** » (qui est une méthode du "Module Part"), tel que X représente l'objet qui sera créé avec les paramètres de cet objet entre les parenthèses ; par exemple pour crée un cube, taper la commande :

h = Part.makeBox(15,20,10) avec :15, 20 et 10 représenter longueur, largeur et hauteur.

- Afficher le cube par la commande : *Part.show(h)*

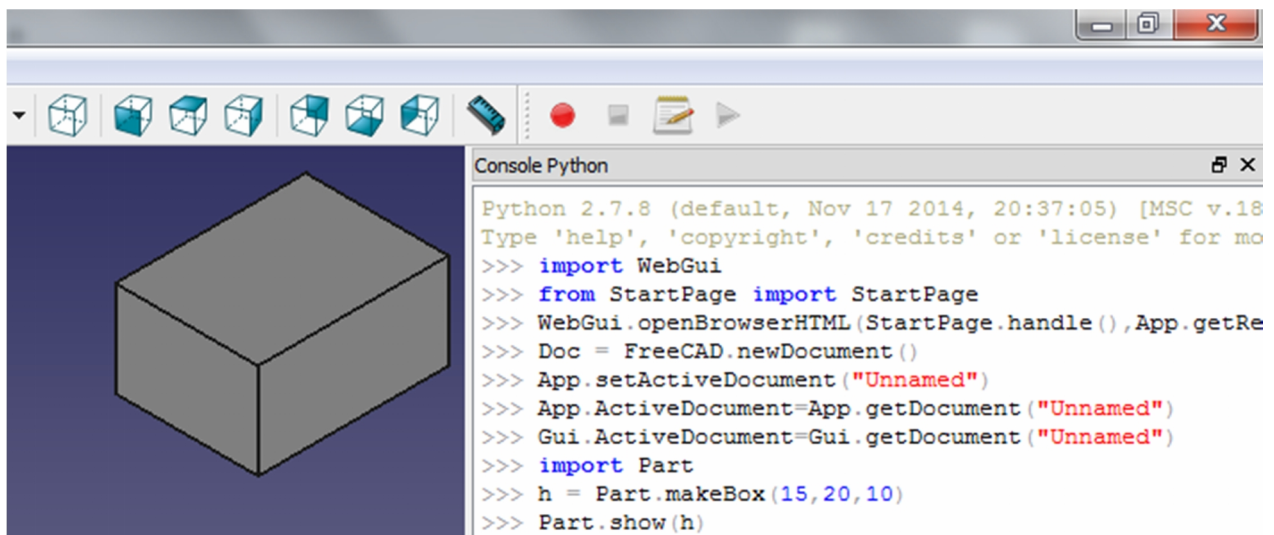


Figure II.4. Création d'un cube par console Python.

La combinaison de makeX() avec d'autres méthodes sont disponibles :

- **makeCircle(r)**: Construit un cercle de rayon (r).
- **makeCone(r1,r2,h)**: Construit un cône de (rayon1,rayon2,hauteur).

- **makeCylinder(r,h)**: Construit un cylindre de (**rayon, hauteur**).
- **makeLine((x1,y1,z1),(x2,y2,z2))**: Construit une ligne aux coordonnées (**x1,y1,z1**),(**x2,y2,z2**) dans l'espace 3D.
- **makePlane(L,l)**: Construit un rectangle de (**longueur, largeur**).
- **makePolygon(list)**: Construit un polygone (**liste de points**).
- **makeSphere(r)**: Construit une sphère de (**rayon**).
- **makeTorus(r1,r2)**: Construit un tore de (**rayon1,rayon2**).

c.2) Méthode 2 (exemple de sphère) : voir figure II.5.

- Créer un nouveau document vide : `doc = FreeCAD.newDocument()`
- Pour créer un objet, nous utilisons la commande suivant :

`Objet1 = doc.addObject("Part::Sphere","nom de l'objet")` .

- Afficher le sphère par le command `doc.recompute()`

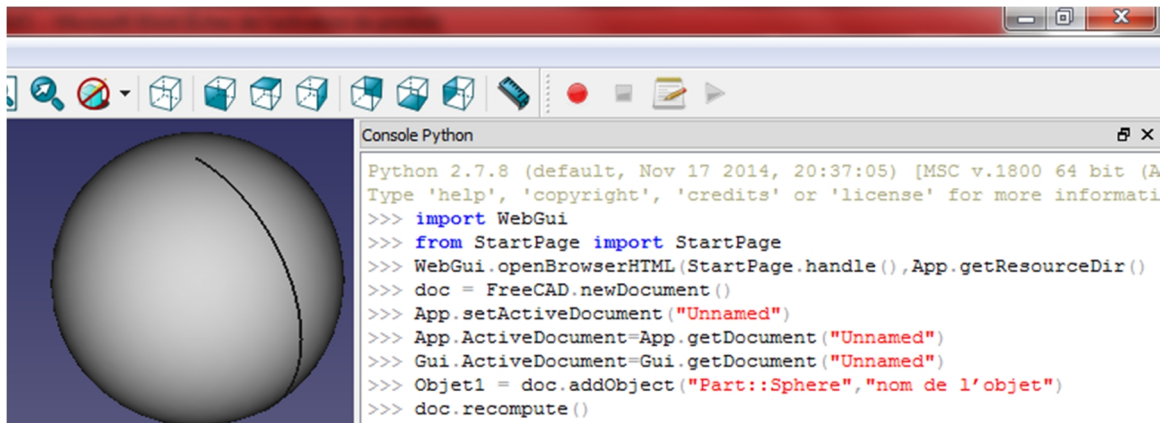


Figure II.5. Création d'une sphère par console Python.

Remarques :

- Dans cette méthode les paramètres tels que : longueur, largeur ...etc. ; sont égales par défaut à 10, et le rayon égale à 2.
- Pour créer autre objet remplacer le mot « Sphère » par un autre mot par exemple :
 - Box : pour créer un cube.
 - Cylinder : pour créer une forme cylindrique.
 - etc.

II.5. Présentation du langage de programmation « Python »

II.5.1. Définition :

Python, développé depuis 1989 par Guido Van Rossum et de nombreux contributeurs bénévoles, est un langage de programmation généraliste, facile à apprendre et rapide à mettre en œuvre. Python est généraliste car, selon les multiples réalisations qu'il a à son actif, il peut être utilisé dans tous les domaines : écriture d'applications pour le Web, programmes de calculs mathématiques, interfaces graphiques, programmation de scripts systèmes, etc. (voir figure II.6) [12], [13].

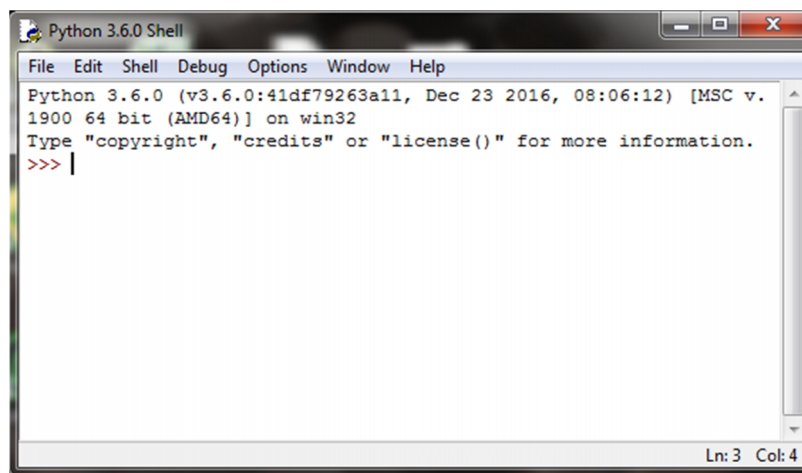


Figure II.6. Interface du « Shell Python ».

II.5.2. Caractéristiques [14] :

- conçu pour produire du code de qualité, portable (fonctionne sous différentes variantes d'Unix, Windows ...) et facile à intégrer.
- de haut niveau, orienté objet et totalement libre.
- hautement productif : La conception d'applications en Python est très rapide car certains aspects de programmation sont gérés automatiquement, comme la gestion des ressources mémoire et le typage des données
- dynamique.

Le langage Python peut être interprété (interprétation du byte code compilé) ou traduit en byte code, qui est ensuite interprété par une machine virtuelle Python. Il est intégrable avec des langages comme le C, le C++ ou Java [12].

II.5.3. Exemples d'utilisation de Python :

- Editer un fichier

Pour éditer un fichier en python on utilise la fonction « open ».

- Crée un fichier :

```
>>> nom=open("nom_de_fichier.txt","w")
```

« w » : pour une ouverture en écriture (WRITE), à chaque ouverture le contenu du fichier est écrasé. Si le fichier n'existe pas python le crée.

Pour crée un fichier dans une partition ou dossier déférente, faire comme suite :

```
>>> nom=open("E:\\dossier1\\nom_de_fichier.txt","w")
```

- Lire un fichier :

```
>>> nom=open("E:\\dossier1\\nom_de_fichier.txt","r")
```

« r » : pour une ouverture en lecture (READ).

- Ecrire dans un fichier :

```
>>> nom=open("E:\\dossier1\\nom_de_fichier.txt","a")
```

```
>>> nom.write("Bonjour")
```

« a » : une ouverture en mode ajout à la fin du fichier (APPEND). Si le fichier n'existe pas python le crée.

A noter que pour le mode d'ouverture 'a', si vous voulez écrire à la ligne, vous pouvez utiliser le saut de ligne \n:

```
>>> nom.write("\nBonsoir")
```

II.6. Contenus détaillés du fichier au format IGES

Les fichiers IGES peuvent être sauvegardés en format binaire ou texte. Dans notre travail, seulement le format texte sera brièvement discuté afin de faciliter la compréhension de la suite de notre travail.

Chapitre II : Méthodologie de travail

Cette partie permet de définir les paramètres généraux permettant de traiter le fichier IGES.

Paramètre	Identification	Paramètre	Identification
1	Caractère spécial de délimitation des champs	15	Nom des unités
2	Caractère de délimitation d'un enregistrement	16	Nombre possible d'épaisseur de lignes différentes (1 à 32768)
3	Identification du logiciel CAO	17	Épaisseur maximale des lignes
4	Nom du fichier IGES	18	Date et heure de la création du fichier
5	Identification du logiciel traducteur ayant généré le fichier	19	Résolution minimale utilisable par l'utilisateur (la plus petite distance)
6	Version du préprocesseur	20	Coordonnée maximale des limites du modèle
7 à 11	Précision des nombres (entiers, réels simple et double précision)	21	Nom de l'auteur
12	Identification du logiciel CAO récepteur	22	Organisation
13	Échelle du modèle	23	Numéro de la version IGES utilisée
14	Indicateur d'unités (1=pouce)	24	Norme de dessin utilisée (1=ISO, 2=AFNOR, 3=ANSI, 4=BSI, 5=CSA)

Tableau II.2. Paramètres généraux permettant de traiter le fichier IGES.

3. Section répertoire : définitions d'entités.

Col: 1-8	Col: 9-16	Col: 17-24	Col: 25-32	Col : 33 40	Col : 41-48	Col : 49-56	Col : 7-64	Col : 65-72	Col : 73-80
Type d'entité	Pointeur vers la section Données	Structure	Motif de ligne	Niveau	Vue	Matrice de transfo	Étiquette affichage	Numéro de statut	Numéro de Séquence
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Type d'entité	Épaisseur de ligne	Couleur	Nombre de lignes dans la section Données	Numéro de formulaire			Étiquette d'entité	Numéro de l'entité	Numéro de Séquence
C11	C12	C13	C14	C15	C16	C17	C18	C18	C20

Tableau II.3. Structure de la section « Répertoire » d'un fichier IGES.

1. Les champs C1 et C11 : donnent le type d'entité. Les plus récentes versions des fichiers.

IGES supportent plus de 50 types d'entités dont les principaux sont :

Numéro d'élément	Type d'élément	Numéro d'élément	Type d'élément
100	Le Cercle et arc de cercle.	114	Les surfaces spline.
104	Arc conique.	116	Le point.
108	Le plan.	120	La surface de révolution.
110	La ligne.	126	Courbe B-spline.
112	Les courbes spline.	128	Surface B-spline.

Tableau II.4. Type d'entités de fichier IGES.

Pour autre des entités de fichier IGES voir l'annexe I.

2. Le champ C2 : donne l'emplacement pour les données de ces entités dans la section Données de paramètres. Cet emplacement est simplement le numéro de ligne dans la section PD qui a la première ligne de ces données d'entité

3. Le champ C3 : Zéro ou pointeur vers une entité de définition. Ne s'applique pas à la plupart des entités

4. Le champ C4 : Numéro de modèle de police de ligne ou pointeur vers l'entité de motif de police de ligne. Le nombre signifie:

0 : Aucun motif spécifié (par défaut)

1 : Solide

2 : En rupture de stock

3 : Fantômes.

4 : Ligne centrale

5 : pointillés

5. Le champ C5 : Niveau Spécifie les niveaux à associer à cette entité. Permet à l'entité d'apparaître sur plus d'un niveau.

6. Le champ C6 : Vue Spécifie les options de visualisation. Ceux-ci sont:

0 : Indique une visibilité et des caractéristiques égales dans toutes les vues. Pointeur par défaut à l'entité voir, qu'il peut être visualisé à partir de l'entité Reference a View Visible Associativité.

7. Le champ C7 : Références une entité matricielle de transformation (Type 124) ou est nulle par défaut (pas de transformation)

8. Le champ C8 : Références d'une associativité d'affichage d'étiquette (type 402, formulaire 5) qui définit comment l'étiquette d'entité apparaît.

9. Le champ C9 : Contient quatre sections de deux nombres.

- état vide. Soit 00 pour la normale, soit 01 pour blanked.
- Commutateur d'entité subordonné: 00 pour indépendant, 01 pour dépendant physiquement, 02 pour logiquement dépendant et 03 pour les deux.
- Entité Use flag: est 00 pour Géométrie, 01 pour annotation, 02 pour la définition, 03 pour autre, 04 pour Logiciel, 05 pour 2D paramétriques et 06 pour la géométrie de construction.
- Enfin, est la hiérarchie, où 00 indique le haut vers le haut (utilisez les caractéristiques de cette entité), 01 est différé à l'échelle mondiale (n'utilisez pas les caractéristiques de cette entité) et 02 utilise la propriété hiérarchie (utilisez l'entité hiérarchique (Type 406) pour déterminer les caractéristiques du regroupement hiérarchique).

10. Les champs C10 et C20 : est le numéro de ligne de cette section (pas du haut du fichier). Ceci est également utilisé pour indiquer cette entité de saisie de données.

11. Le champ C12 : Spécifie l'épaisseur lors de l'affichage de l'entité. Le plus petit est 1, 0 est par défaut.

12. Le champ C13 : spécifie la couleur de l'entité. Les principaux codes de couleur sont les suivants :

0 : Aucune. 3 : Vert. 6 : Magenta.

1 : Noir. 4 : Bleu. 7 : Cyan.

2 : Rouge. 5 : Jaune. 8 : Blanc.

13. le champ C14 : Spécifie le nombre de lignes que cette entité prend en compte dans la section Paramètres.

14. Le champ 15 : Indique le formulaire, ou la représentation de cette entité. Modifie la manière dont les données des paramètres sont interprétées. La valeur par défaut est 0.

15. Les champs 16 et 17 : non utilisé (réservé).

16. Le champ 18 : Identifiant spécifié par l'application justifié à droite.

17. Le champ 19 : Qualificatif numérique pour l'étiquette de l'entité. Les deux forment ensemble un identifiant unique pour l'entité.

4. **Section P :** Section des paramètres (Définition géométrique des entités 2D et 3D).

Données de paramètres vient juste après les entrées de données et répertorie les données pour chaque entrée respective.

Section T : Section Totale. C'est une ligne qui récapitule le nombre de lignes pour chaque section.

Chapitre III :

Développement de l'application

III.1. Introduction

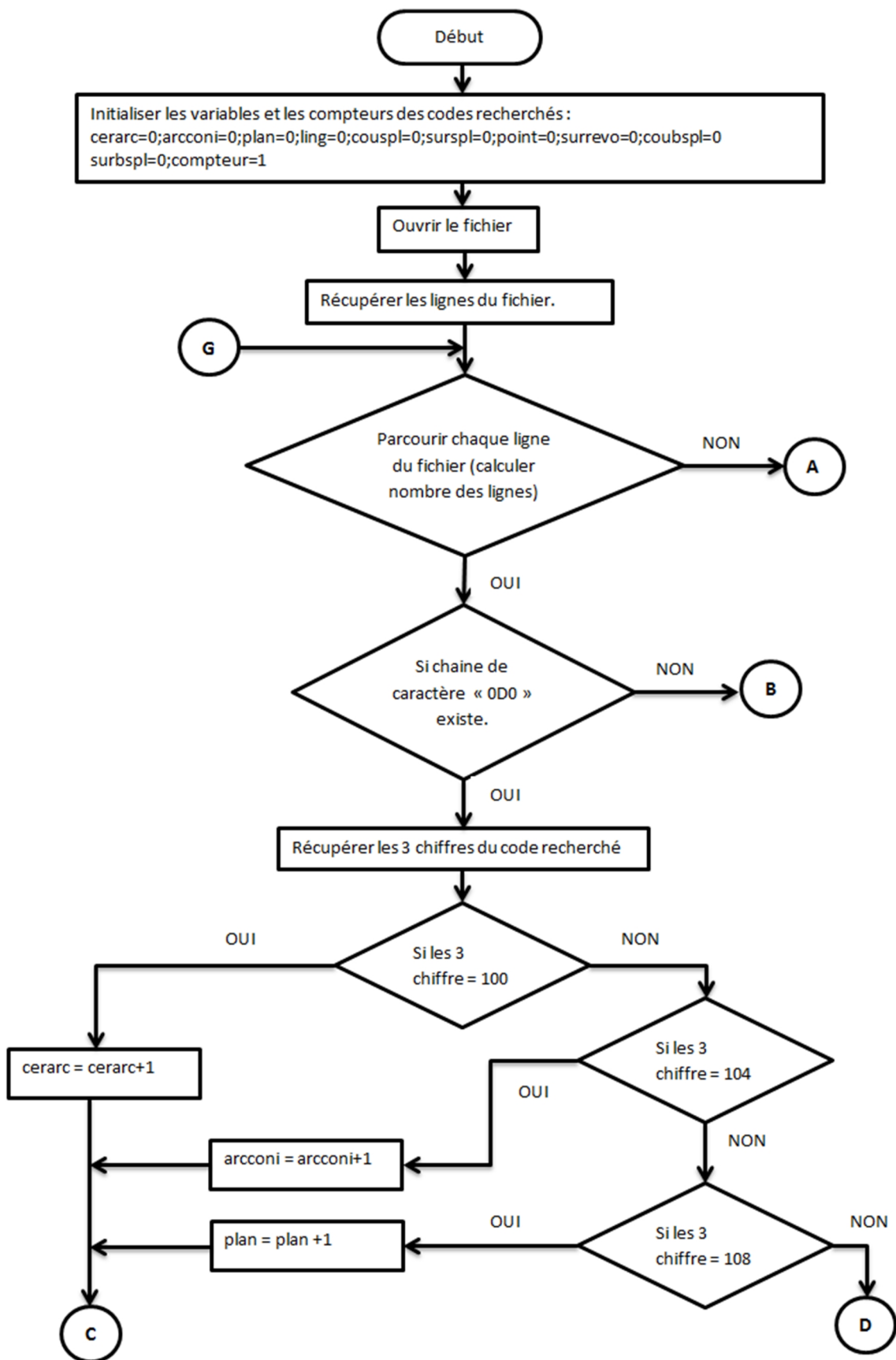
Dans ce dernier chapitre, nous allons vous présenter l'algorithme que nous avons mis en place pour extraire les données géométriques et topologiques du modèle CAO à partir du fichier IGES. Par la suite, nous allons traduire cet algorithme en langage Python. Et pour finir, nous allons décrire en détails comment nous avons intégré notre code dans l'application FreeCAD en utilisant les Macros.

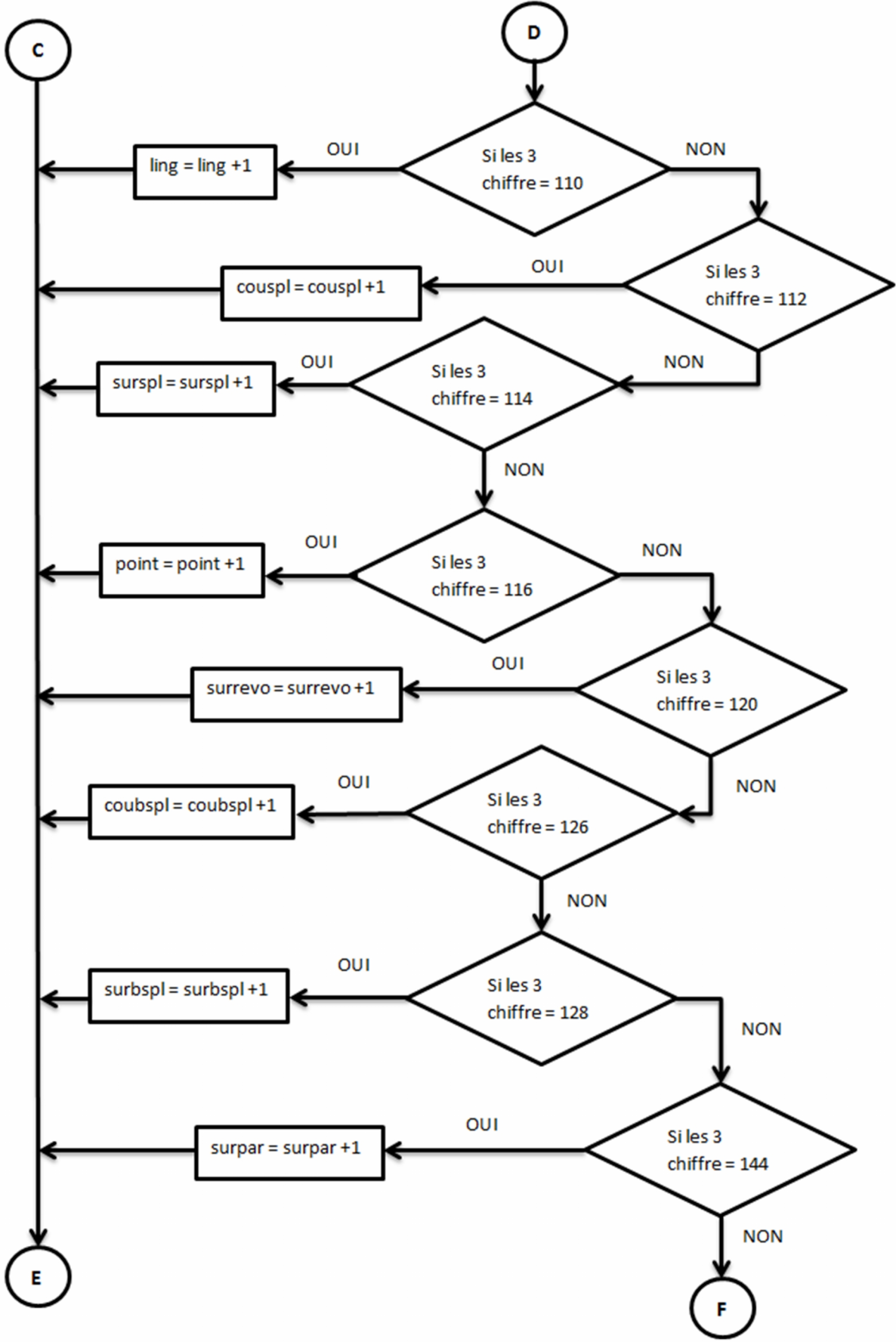
III.2. Algorithme d'extraction des données

Comme vous le savez, un algorithme est une suite d'instructions ayant pour but de résoudre un problème donné. Ces instructions doivent être exécutées de façon automatique par un ordinateur.

Pour atteindre les objectifs fixés dans notre travail, nous avons mis en place un algorithme qui suit les étapes suivantes (voir figure III.1) :

- Initialiser les paramètres et compteurs ;
- Ouvrir le fichier après avoir extrait le nom et le chemin du fichier ouvert dans FreeCAD ;
- Récupère les lignes du fichier IGES au format texte ;
- Parcourir chaque ligne du fichier dans l'ordre ;
 - Vérifier le code correspondant aux données géométriques et topologiques que nous voulons extraire pour ne garder que les lignes qui nous intéressent ;
 - Incrémenter les compteurs pour chaque donnée extraite ;
- Afficher le résultat sous forme de liste dans une fenêtre dans le logiciel FreeCAD ;
- Sauvegarder le résultat dans un fichier texte pour une exploitation ultérieure.





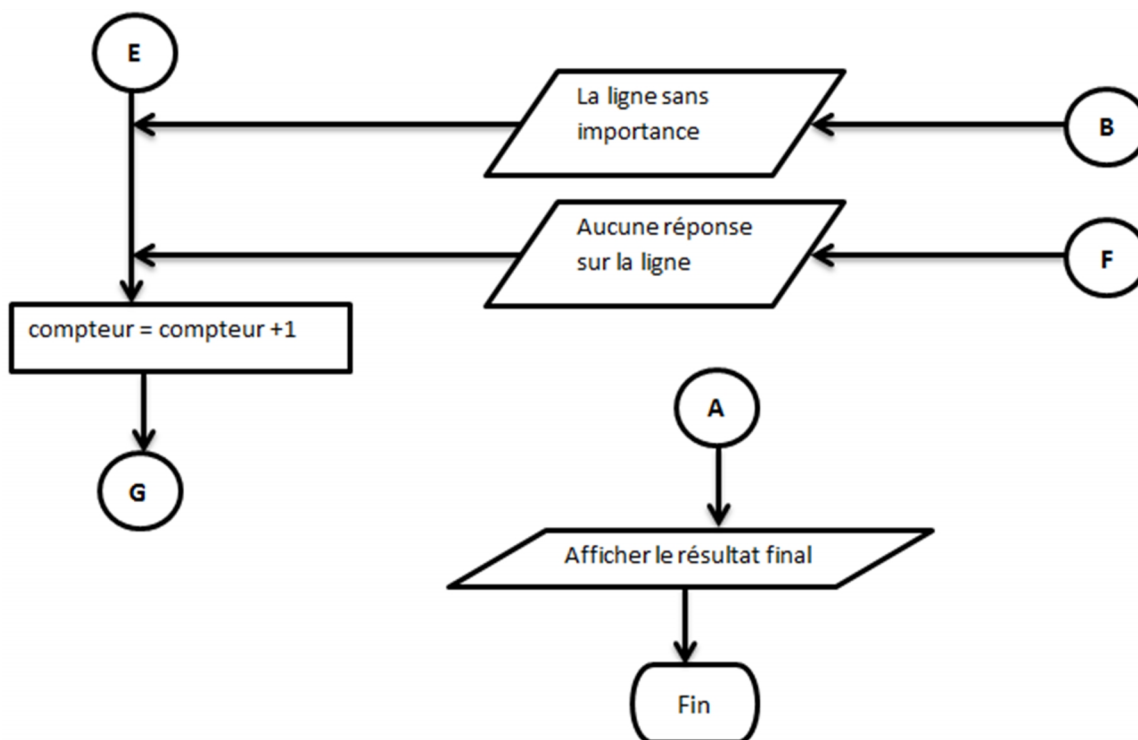


Figure III.1. Algorithme d'extraction des données géométriques et topologiques du fichier au format IGES.

III.3. Programmation en Python

Une fois que l'algorithme d'extraction de données mis en place et vérifié, nous avons transcrit cet algorithme en langage python (voir figure III.2) pour être sûr de la qualité des résultats qui nous donne en faisant le teste sur des fichiers IGES indépendamment du logiciel FreeCAD. L'intégralité du code de programmation en langage Python est donnée dans l'annexe II.



```
finale.py - C:\Users\HK\Desktop\finale.py (3.6.0)
File Edit Format Run Options Window Help
Cercles_et_Arcs_de_cercle = 0;Arcs_coniques = 0;Plans = 0;Lignes = 0;
Courbes_Spline = 0;Surfaces_Spline = 0;Points = 0;Surfaces_Revolutions = 0
Courbes_Bspline = 0;Surfaces_Bspline = 0;compteur=1
Fichier_IGES=open('E://Piece//Piece_1.iges','r')
lignes=[line.strip('\n')for line in Fichier_IGES.readlines()]
for line in lignes:
    if line.find('0D0') != -1:
        if line[5:8] == '100':
            Cercles_et_Arcs_de_cercle=Cercles_et_Arcs_de_cercle+1
        elif line[5:8] == '104':
            Arcs_coniques=Arcs_coniques+1
        elif line[5:8] == '108':
            Plans=Plans+1
        elif line[5:8] == '110':
            Lignes=Lignes+1
        elif line[5:8] == '112':
            Courbes_Spline=Courbes_Spline+1
        elif line[5:8] == '114':
            Surfaces_Spline=Surfaces_Spline+1
        elif line[5:8] == '116':
            Points=Points+1
        elif line[5:8] == '120':
            Surfaces_Revolutions=Surfaces_Revolutions+1
        elif line[5:8] == '126':
            Courbes_Bspline=Courbes_Bspline+1
        elif line[5:8] == '128':
            Surfaces_Bspline=Surfaces_Bspline+1
        else:
            print('aucune réponse sur la ligne : ',compteur)
    else:
        print('la ligne :',compteur,'sans amportance')
    compteur=compteur+1
Fichier_IGES.close()
print('Cercles_et_Arcs_de_cercle =', int(Cercles_et_Arcs_de_cercle/2))
print('Arcs_coniques =', int(Arcs_coniques/2))
print('Plans =', int(Plans/2))
print('Lignes =', int(Lignes/2))
print('Courbes_Spline =', int(Courbes_Spline/2))
print('Surfaces_Spline =', int(Surfaces_Spline/2))
print('Points =', int(Points/2))
print('Surfaces_Revolutions =', int(Surfaces_Revolutions/2))
print('Courbes_Bspline =', int(Courbes_Bspline/2))
print('Surfaces_Bspline =', int(Surfaces_Bspline/2))
Ln: 1 Col:
```

Figure III.2. Code source sur Python.

III.4. Intégration de notre programme dans le logiciel FreeCAD

Nous allons vous présenter l'intégration de notre programme dans le logiciel FreeCAD en utilisant les macros et cela sous forme de tutoriel, étape par étape, afin de faciliter la compréhension et pour que notre travail puisse être facilement exploité et réutilisé par d'autres personnes.

III.4.1. Programme de la macro :

Voici les étapes nécessaires pour l'intégration de notre programme sous forme de macro dans le logiciel FreeCAD :

- Importation des modules des bibliothèques nécessaires :
 - Importer le module « QtGui » à l'intérieur du module « PySide » pour accéder à l'interface Qt de FreeCAD.
 - Importer les modules « subprocess » et « os » pour l'affichage du résultat.

```
from PySide import QtGui
import subprocess
import os
```

- Déclaration des variables nécessaires :

```
Cercles_et_Arcs_de_cercle = 0 ; Arcs_coniques = 0 ; Plans = 0 ; Lignes = 0
Courbes_Spline = 0 ; Surfaces_Spline = 0 ; Points = 0 ; Surfaces_Revolutions = 0 ;
Courbes_Bspline = 0 ; Surfaces_Bspline = 0 ; compteur=1
```

- Obtention du chemin du document actuel : Pour obtenir le chemin du document (pièce) actif. Il doit exister deux documents de l'extension « FCSTD » et « IGES », avec le même nom.

```
Ma_piece=str(Gui.ActiveDocument.Document.FileName)
Ma_piece_2=Ma_piece.replace('FCStd','iges')
Fichier_IGES=open(Ma_piece_2,'r')
```

- Ouvrir et récupérer les lignes du fichier (document IGES) : lire tous les lignes du fichier, avec enlèvement de l'espace à la fin de chaque ligne.

```
lignes=[line.strip('\n')for line in Fichier_IGES.readlines()]
```

- Parcourir chaque ligne du fichier à partir de l'ensemble des lignes : c'est-à-dire lire chaque fois une seule ligne de l'ensemble des lignes.

```
for line in lignes:
```

- Rechercher les lignes avec la chaîne de caractère qui nous intéresse : pour chaque ligne nous cherchons sur la chaîne de caractère '0D0' si existe ou non.

```
if line.find('0D0') != -1:
```

- Choisir la condition : on récupère les 3 chiffres du code recherché et faire des boucles (les conditions suivantes sont appliquées sur une seule ligne) pour chaque code et quand on le trouve on incrémente le compteur.

```
if line[5:8] == '100':  
    Cercles_et_Arcs_de_cercle=Cercles_et_Arcs_de_cercle+1  
elif line[5:8] == '104':  
    Arcs_coniques=Arcs_coniques+1  
elif line[5:8] == '108':  
    Plans=Plans+1  
elif line[5:8] == '110':  
    Lignes=Lignes+1  
elif line[5:8] == '112':  
    Courbes_Spline=Courbes_Spline+1  
elif line[5:8] == '114':  
    Surfaces_Spline=Surfaces_Spline+1  
elif line[5:8] == '116':  
    Points=Points+1  
elif line[5:8] == '120':  
    Surfaces_Revolutions=Surfaces_Revolutions+1  
elif line[5:8] == '126':  
    Courbes_Bspline=Courbes_Bspline+1  
elif line[5:8] == '128':  
    Surfaces_Bspline=Surfaces_Bspline+1  
else:  
    print(' ')  
else:  
    print(' ')  
compteur=compteur+1
```

- Fermeture du fichier: pour appliquer le programme sur le fichier (pièce) que nous avons ouvert au début, il faut le fermer.

```
Fichier_IGES.close()
```

- Créer un fichier pour le résultat : nous avons créé un fichier texte pour sauvegarder le résultat final des compteurs (mode créer) :

```
Fichier_resultat=open('E:/resultat.txt','w')
```

- Ajouter les résultats à un fichier : nous ajoutons le résultat final des compteurs dans le fichier que nous avons créé (avec un commentaire au début du fichier)

```
Fichier_resultat.write('*****'+'\n')
Fichier_resultat.write(' *** Le Resultat Final : ***'+'\n')
Fichier_resultat.write('*****'+'\n\n')
Fichier_resultat.write(' - Cercles_et_Arcs_de_cercle...=
'+str(Cercles_et_Arcs_de_cercle/2)+'\n')
Fichier_resultat.write(' - Arcs_coniques.....= '+str(Arcs_coniques/2)+'\n')
Fichier_resultat.write(' - Plans.....= '+str(Plans/2)+'\n')
Fichier_resultat.write(' - Lignes.....= '+str(Lignes/2)+'\n')
Fichier_resultat.write(' - Courbes_Spline.....= '+str(Courbes_Spline/2)+'\n')
Fichier_resultat.write(' - Surfaces_Spline.....= '+str(Surfaces_Spline/2)+'\n')
Fichier_resultat.write(' - Points.....= '+str(Points/2)+'\n')
Fichier_resultat.write(' - Surfaces_Revolutions.....= '+str(Surfaces_Revolutions/2)+'\n')
Fichier_resultat.write(' - Courbes_Bspline.....= '+str(Courbes_Bspline/2)+'\n')
Fichier_resultat.write(' - Surfaces_Bspline.....= '+str(Surfaces_Bspline/2)+'\n')
```

- Fermeture du fichier : pour sauvegarder notre résultat automatiquement dans ce fichier (Fichier_resultat), il faut le fermer .

```
Fichier_resultat.close()
```

- Ouvrir le fichier de résultat : ce fichier est qu'on sauvegarder notre dernier résultat (mode lecture) :

```
Affichier_resultat=open('E:/resultat.txt','r')
```


- Récupérer les lignes du fichier : pour l'affichage des résultats, il faut lire chaque ligne du fichier de notre résultat.

```
ligne_1=Affichier_resultat.readline()
ligne_2=Affichier_resultat.readline()
ligne_3=Affichier_resultat.readline()
ligne_4=Affichier_resultat.readline()
ligne_5=Affichier_resultat.readline()
ligne_6=Affichier_resultat.readline()
ligne_7=Affichier_resultat.readline()
ligne_8=Affichier_resultat.readline()
ligne_9=Affichier_resultat.readline()
ligne_10=Affichier_resultat.readline()
ligne_11=Affichier_resultat.readline()
ligne_12=Affichier_resultat.readline()
ligne_13=Affichier_resultat.readline()
ligne_14=Affichier_resultat.readline()
ligne_15=Affichier_resultat.readline()
```

- Fenêtre des résultats : pour afficher le résultat (les lignes qui sont récupérées) dans une nouvelle fenêtre (boîte de message):

```
reply = QtGui.QMessageBox.information(None,"Information",ligne_1+ligne_2+ligne_3
+ligne_4+ligne_5+ligne_6+ligne_7+ligne_8+ligne_9+ligne_10+ligne_11+ligne_12
+ligne_13+ligne_14+ligne_15)
```

- Fermer du fichier résultat : pour terminer la récupération et l'affichage du résultat.

```
Affichier_resultat.close()
```

- Obtenir du chemin de bloc-notes: le résultat doit être ouvert sous la forme d'un fichier texte ; il faut déclarer le chemin du programme qu'on utilise pour ouvrir ce genre de fichier, dans ce cas s'est notepad.

```
chemin_du_Blocnots = 'C:\\Windows\\System32\\notepad.exe'
```

- Obtenir le chemin du fichier « resultat ».

```
chemin_du_fichier_resultat = 'E:\\resultat.txt'
```

- Ouverture de fichier « resultat » avec « notepad » :

```
subprocess.call([chemin_du_Blocnots, chemin_du_fichier_resultat])
```

L'intégralité du code de programmation de la macro en langage Python est donnée dans l'annexe III

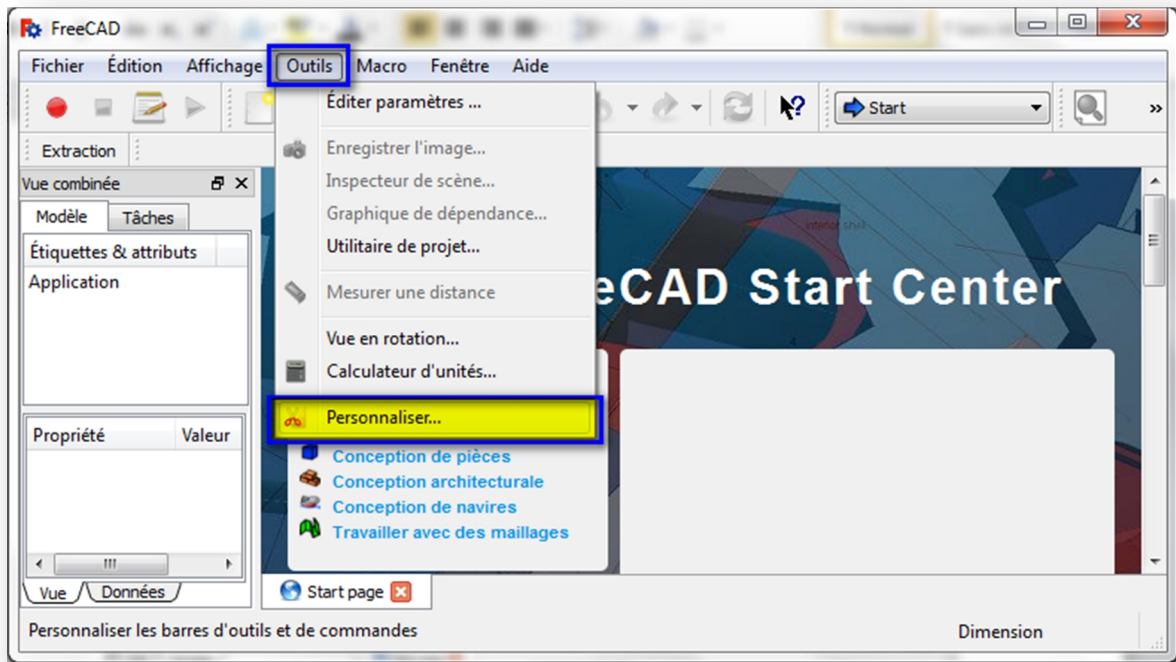
III.4.2. Personnaliser la barre d'outil (ajouté un bouton) :

Dans cette partie, nous montrons la personnalisation des barres d'outils. Des outils y compris les Macro-outils utilisés dans les différents ateliers. Dans un exemple d'une Macro devient une Macro-outil par la création d'un Menu de texte, un Outil de pointage et une Icône. Par la suite, cette Macro-outil devient une partie d'une barre d'outils supplémentaire dans l'établi.

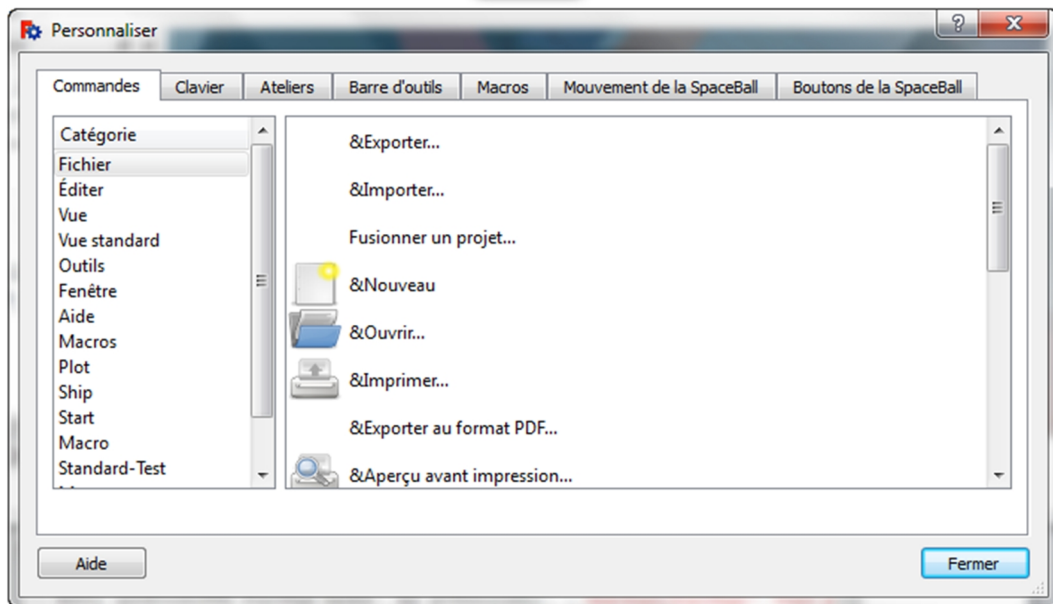
Création du bouton :

Pour crée notre bouton, nous allons suivre les étapes suivantes :

- Cliquez sur **Menu > Outils > Personnaliser** (voire figure III.3.a)
- La fenêtre "Personnaliser" est affichée (voire figure III.3.b)



a



b

Figure III.3. Fenêtre personnalisation.

- Sélectionner l'onglet « Macro », cliquer sur le bouton [...] pour sélectionner un icône pour la macro. (Voir figure III.4)

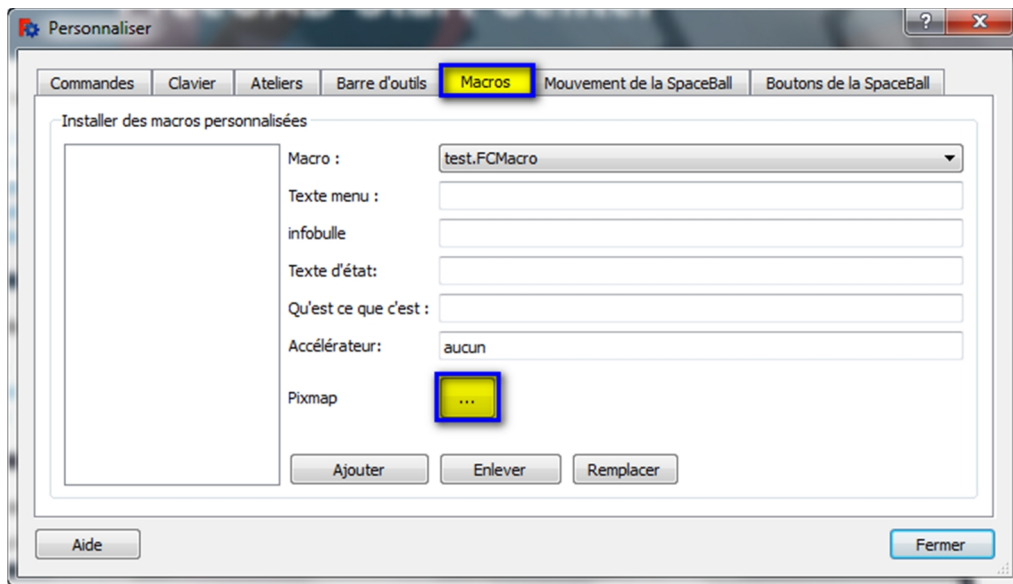


Figure III.4. Onglet macro de la fenêtre personnaliser.

- Dans la fenêtre qui s'affiche, sélectionner dans la liste une icône pour la macro. Si l'icône n'est pas listé, cliquer sur le bouton Dossier d'icônes... (voire figure III.5. a).
- Dans la fenêtre qui affiche cliquer sur + (voire figure III.5. b)
- Sélectionner le dossier des icônes pour ajouter des icônes à la bibliothèque (voire Figure III.5. c).

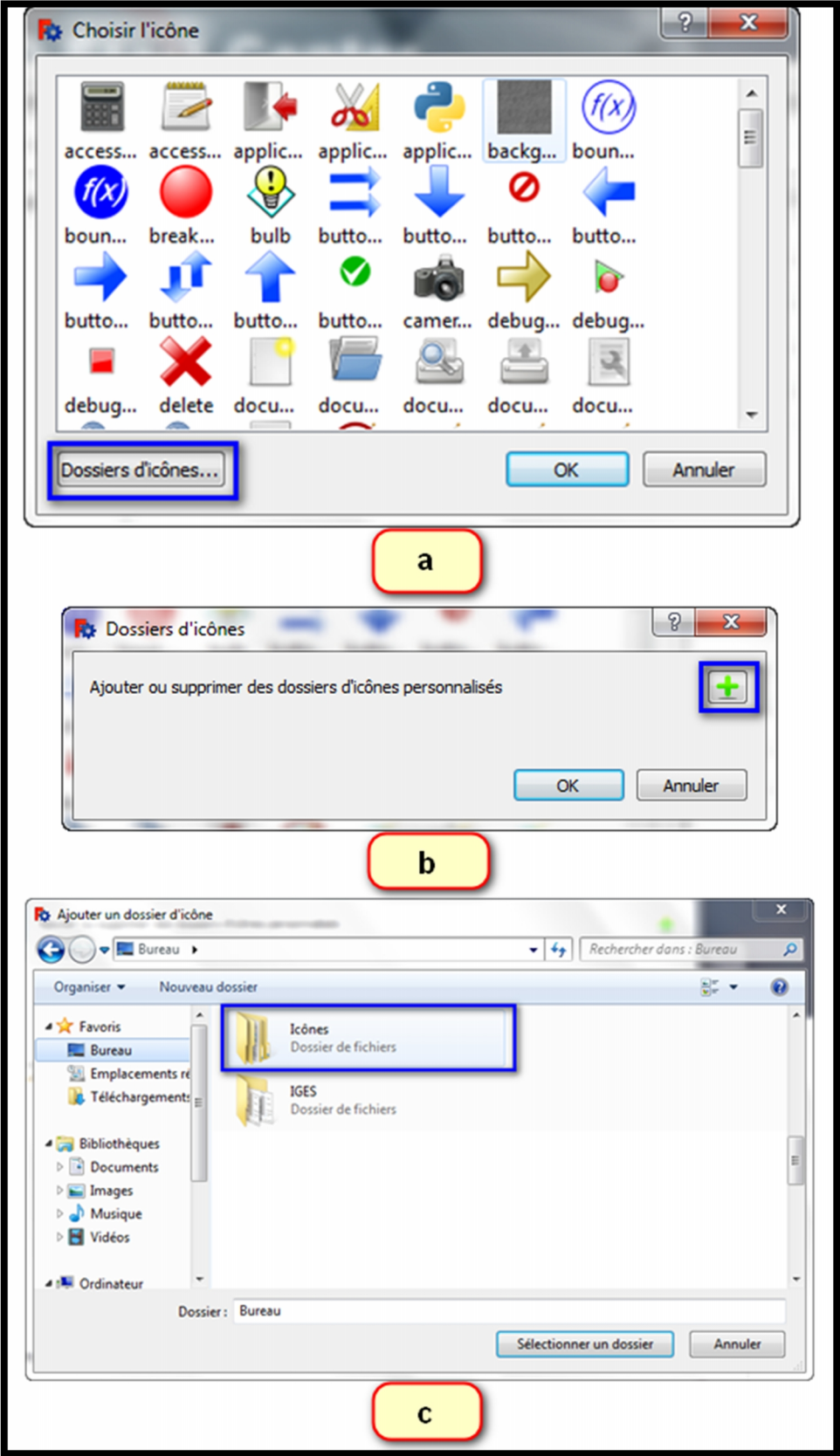


Figure III.5. Ajouter Dossier d'icône.

- Sélectionner l'icône et cliquer sur le bouton **Ok** pour valider (voire figure III.6. a)
- l'icône sélectionnée est maintenant affichée (voire figure III.6. b).

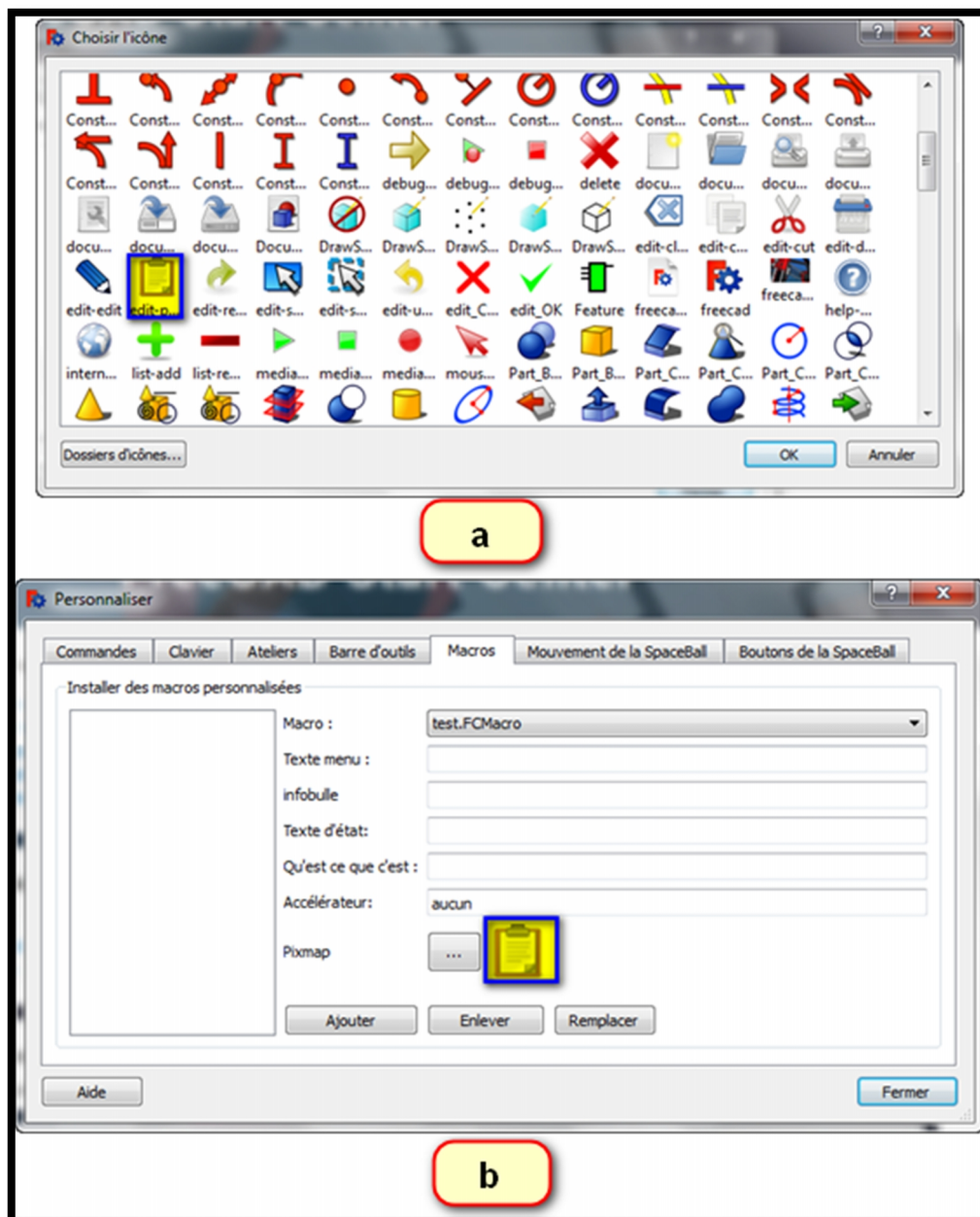


Figure III.6. Icône de la macro.

- Choisir la macro et spécifier un texte de menu (ce texte apparaîtra comme l'étiquette dans le menu); on remplit également l'info-bulle (qui est le texte qui apparaît lorsque la souris survole le bouton de la barre d'outils); il y a d'autres champs textes pour notre bouton qui sont facultatifs mais que nous pouvons remplir, si termine cliquer sur le bouton **Ajoute** (voire figure III.7. a)

- le bouton est maintenant créé (voire figure III.7. b).

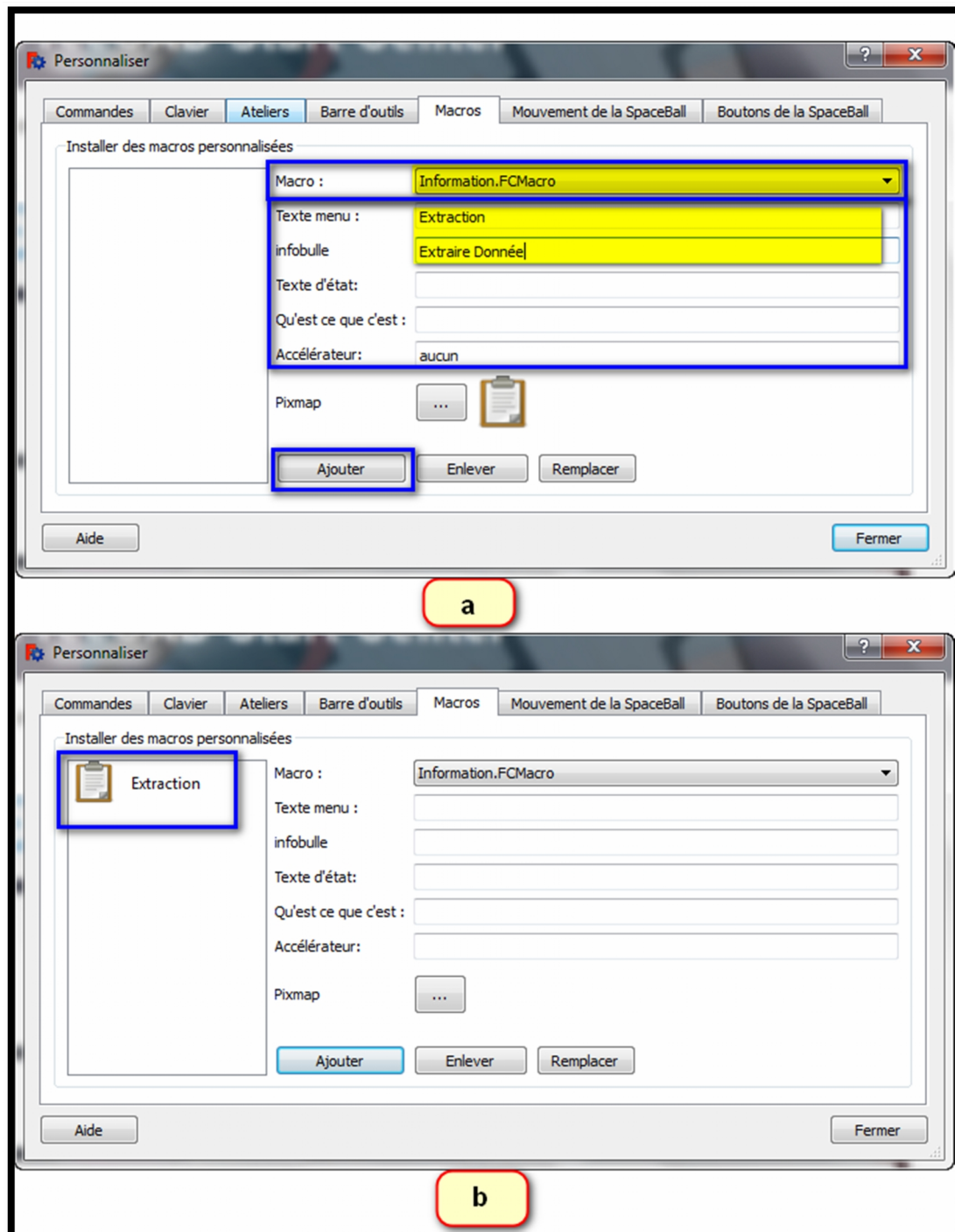
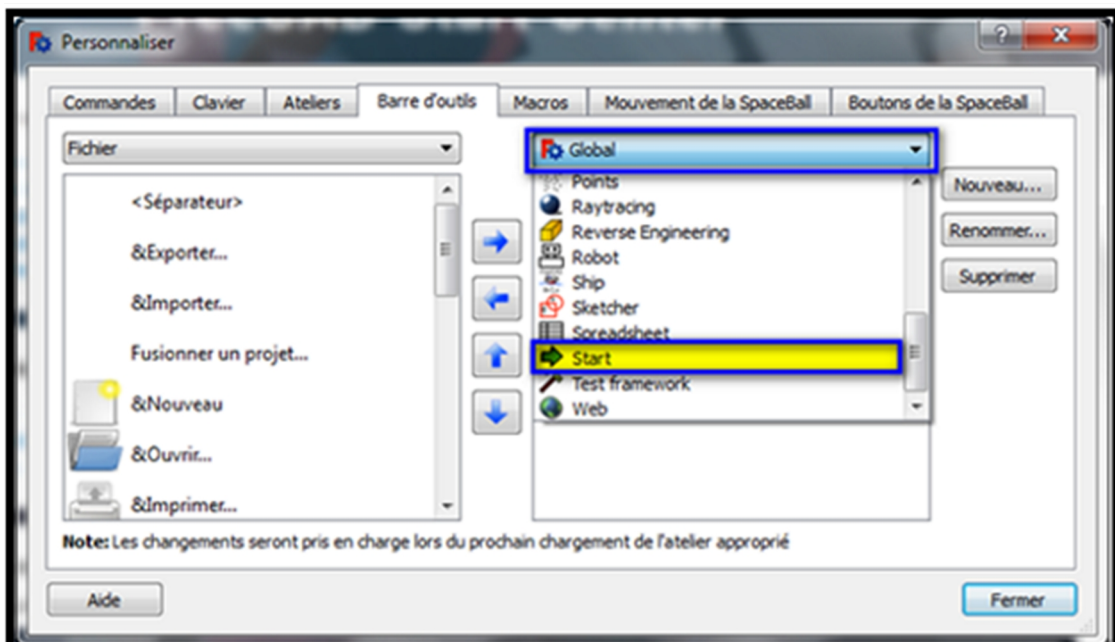


Figure III.7. Le bouton de macro est créé.

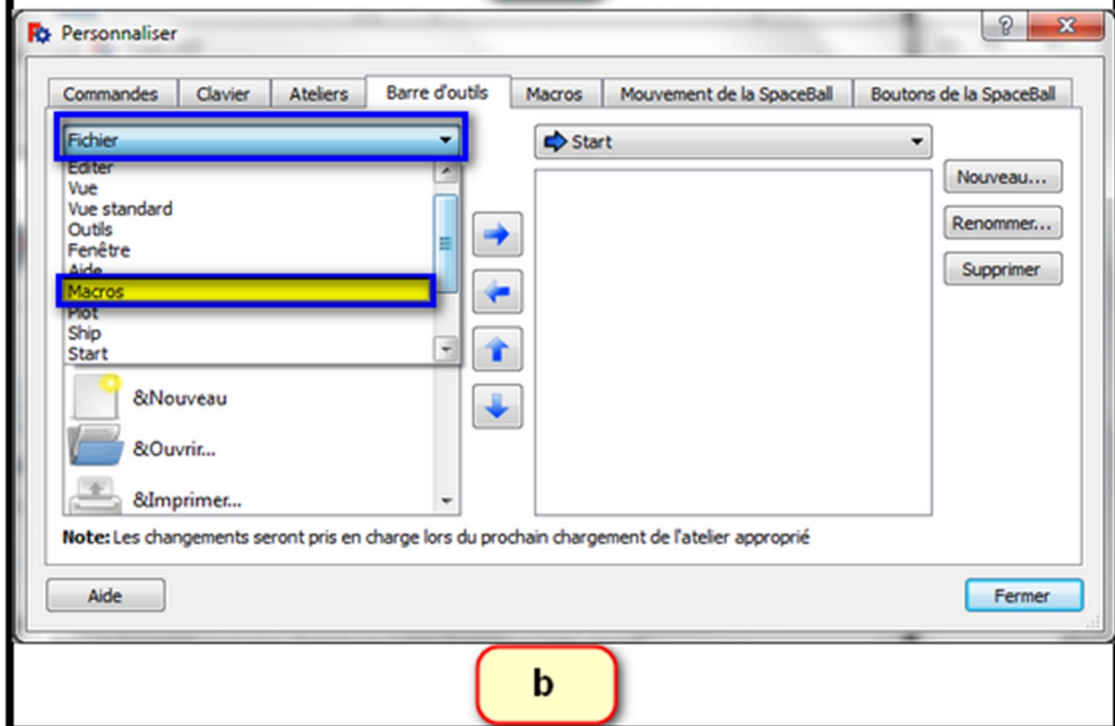
- créer une barre d'outils pour le bouton, cliquez sur l'onglet "Barre d'outils" et l'atelier sur lequel sera attachée la Barre d'outils, dans notre cas st Start (voire figure III.8. a).

Chapitre III : Développement de l'Application

- Dans la fenêtre de gauche, sélectionnez l'onglet Fichier et choisir Macros (voire figure III.8. b).



a



b

Figure III.8. Création barre d'outils du bouton.

- Maintenant la macro et son icône apparaissent.

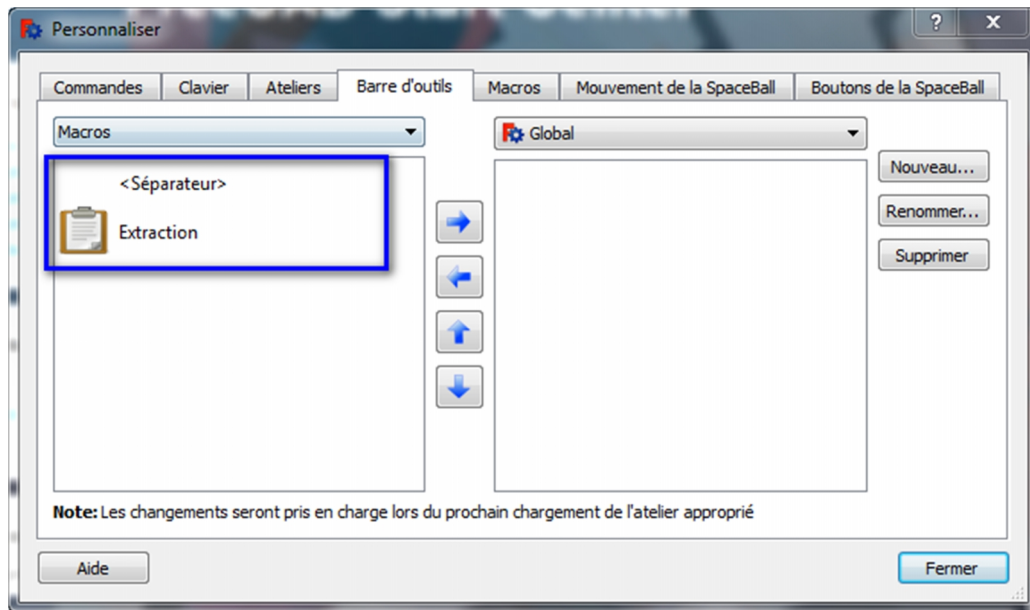


Figure III.9. La macro et son icône apparaissent.

- Cliquez le bouton **Nouveau...** (choisissez votre barre d'outils ici votre barre d'outils est attachée à l'atelier Start).
- dans la fenêtre Nouvelle barre d'outils entrez le nom de la nouvelle barre d'outils et validez avec le bouton **OK** .

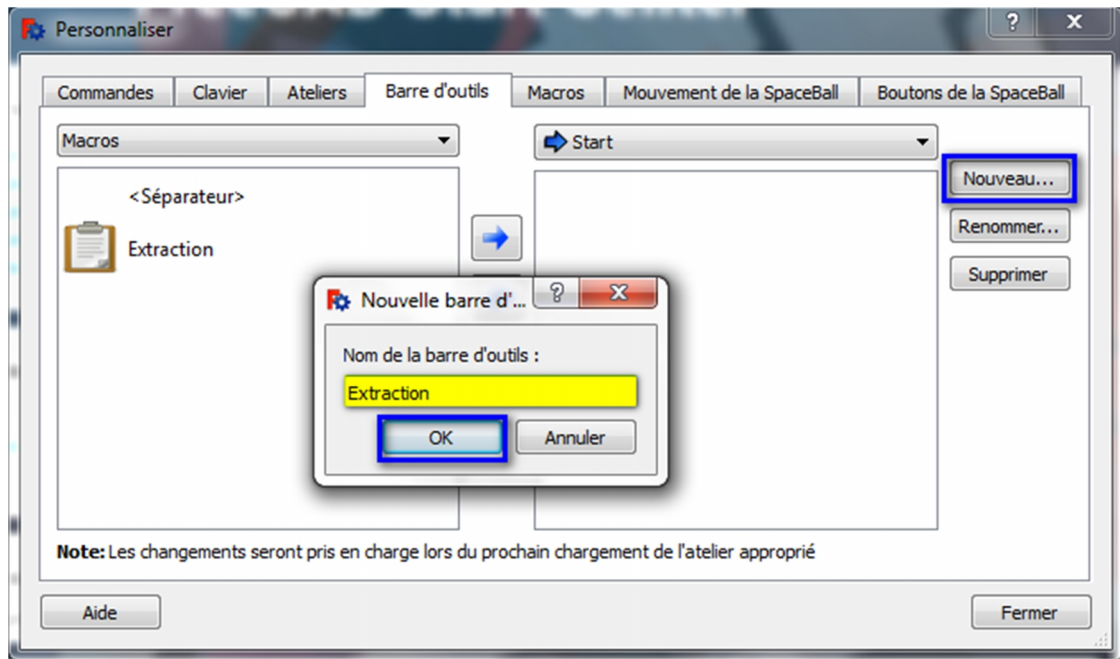


Figure III.10. Création nouvelle barre d'outils

- la barre d'outils est maintenant créée, assurez-vous de cocher la case à cocher pour valider la barre à outils et votre barre à outils apparaît. Puis sélectionnez la macro par son icône dans la fenêtre de gauche puis cliquez sur le **Bouton** flèche bleue pour faire passer l'icône de la macro dans la fenêtre de droite (voire figure III.11. a).
- maintenant une barre d'outils appelée "Extraction" avec un bouton qui s'appelle aussi "Extraction" a été créé (voire figure III.11. b).

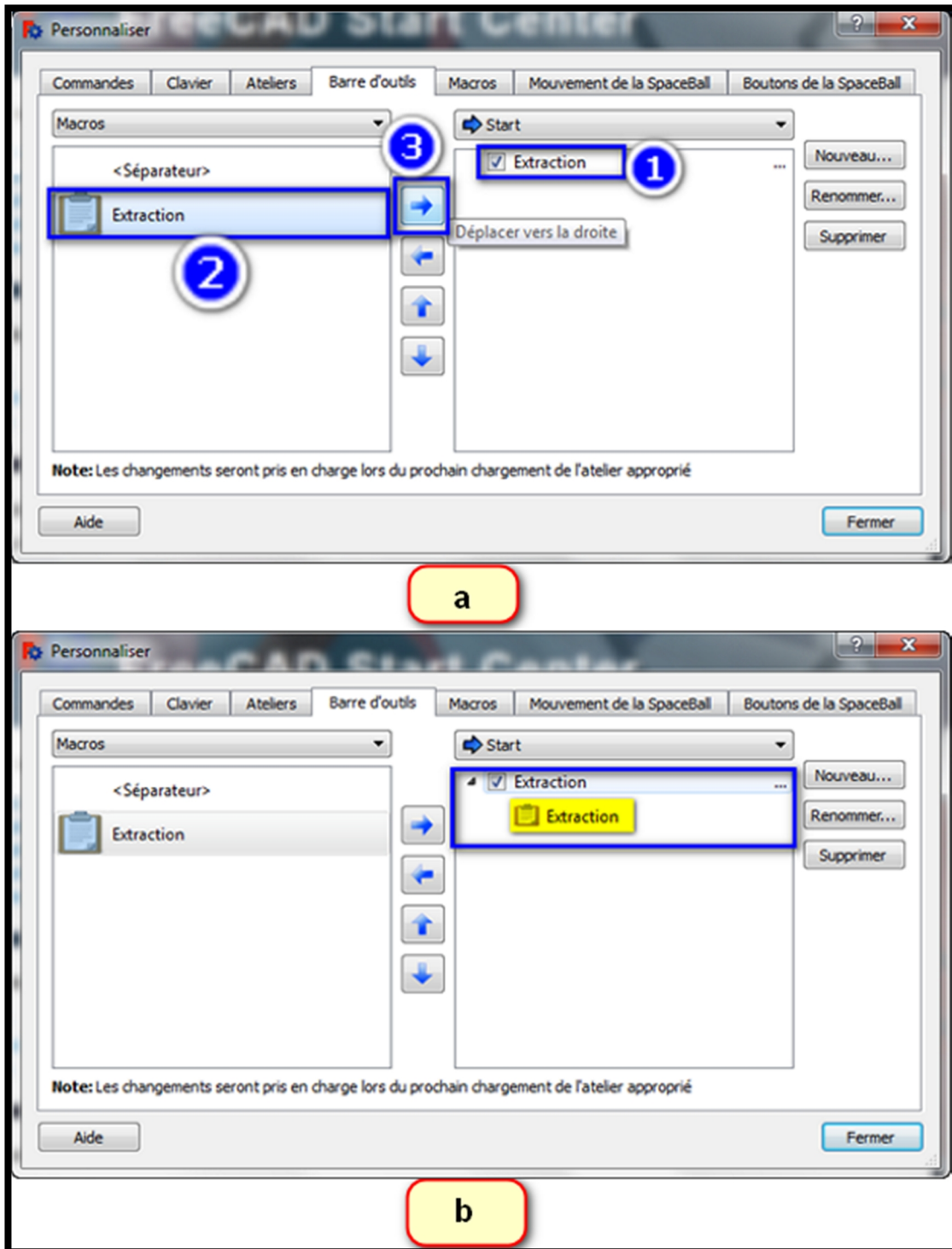


Figure III.11. Ajouter le bouton à barre d'outils.

- la nouvelle barre d'outils et son bouton sont maintenant affichés dans la fenêtre active attachée à l'atelier "Start" et aussi visible dans le menu en faisant clic droit sur un endroit libre de la barre d'outils.

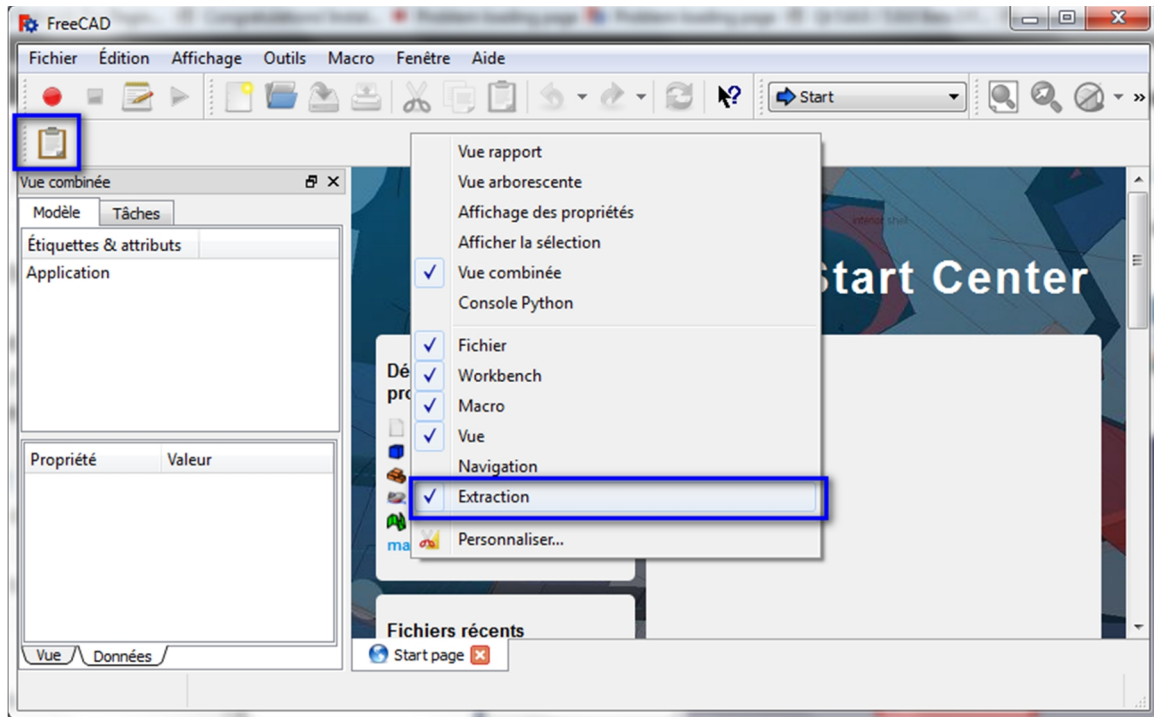


Figure III.12. L'affichage de bouton sur le menu.

III.5. Exemples d'exécution du programme

Pour comprendre la façon dont il fonctionne par la macro, nous donnerons deux exemples de deux différentes pièces, mais avant lancement de la macro, d'abord vérifier si le format des pièces est « FCSTD », avec autre format « IGES » de la même pièce (chaque pièce sauvegarder à deux formats et le nom c'est le même).

III.5.1. Exemple d'une pièce simple :

Dans cet exemple, nous applique la macro sur une pièce simple.

- A partir de l'onglet **Fichier** clique sur **Ouvrir...** (voire figure III.13. a)
- Sélectionner la pièce à format « FCSTD » puis clique sur **Ouvrir** (voire figure III.13. b).

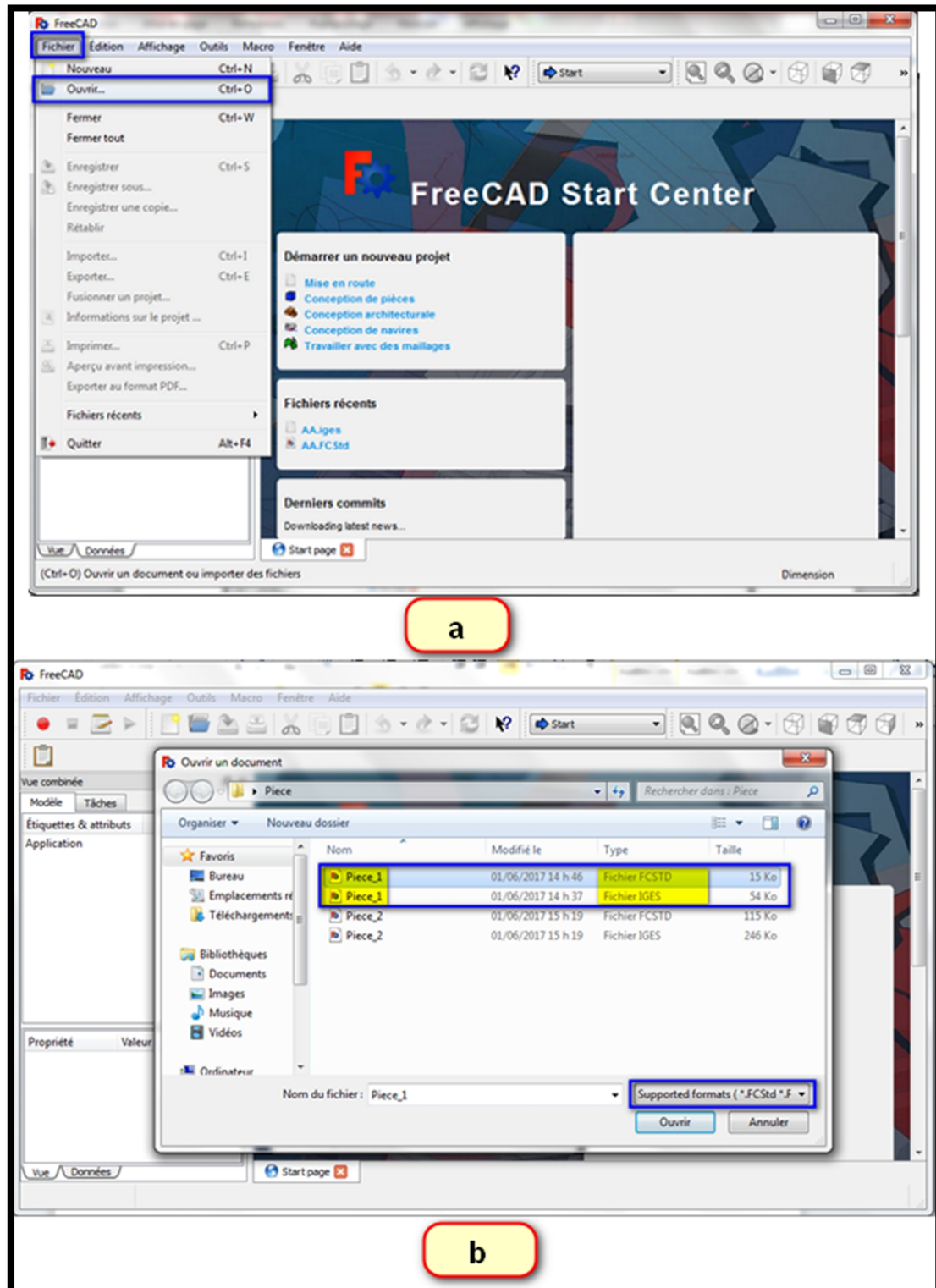


Figure III.13. Ouverture de la pièce (Pièce_1).

- Maintenant la pièce un est active (voire figure III.14. a).
- Cliquer sur le bouton que nous avons créé (voire figure III.14. b).

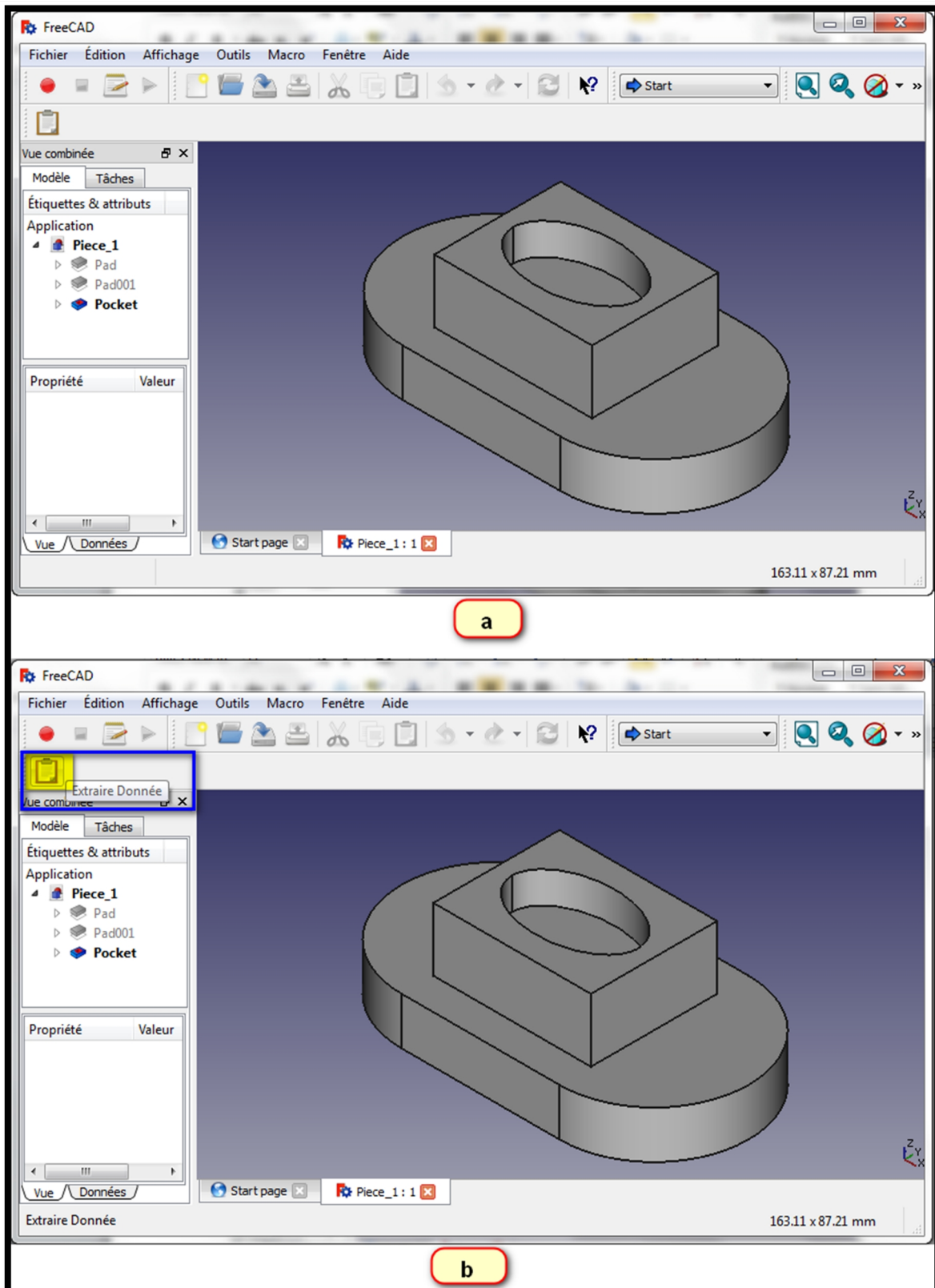


Figure III.14. Lancement de l'extraction des données (Pièce_1).

- Une fenêtre sera affichée qui contient les données de la pièce (voire figure III.15. a).

- Cliquer sur bouton **OK** pour fermer la fenêtre, une autre fenêtre de Bloc-notes (fichier texte) contienne les mêmes données (voire figure III.15. b).

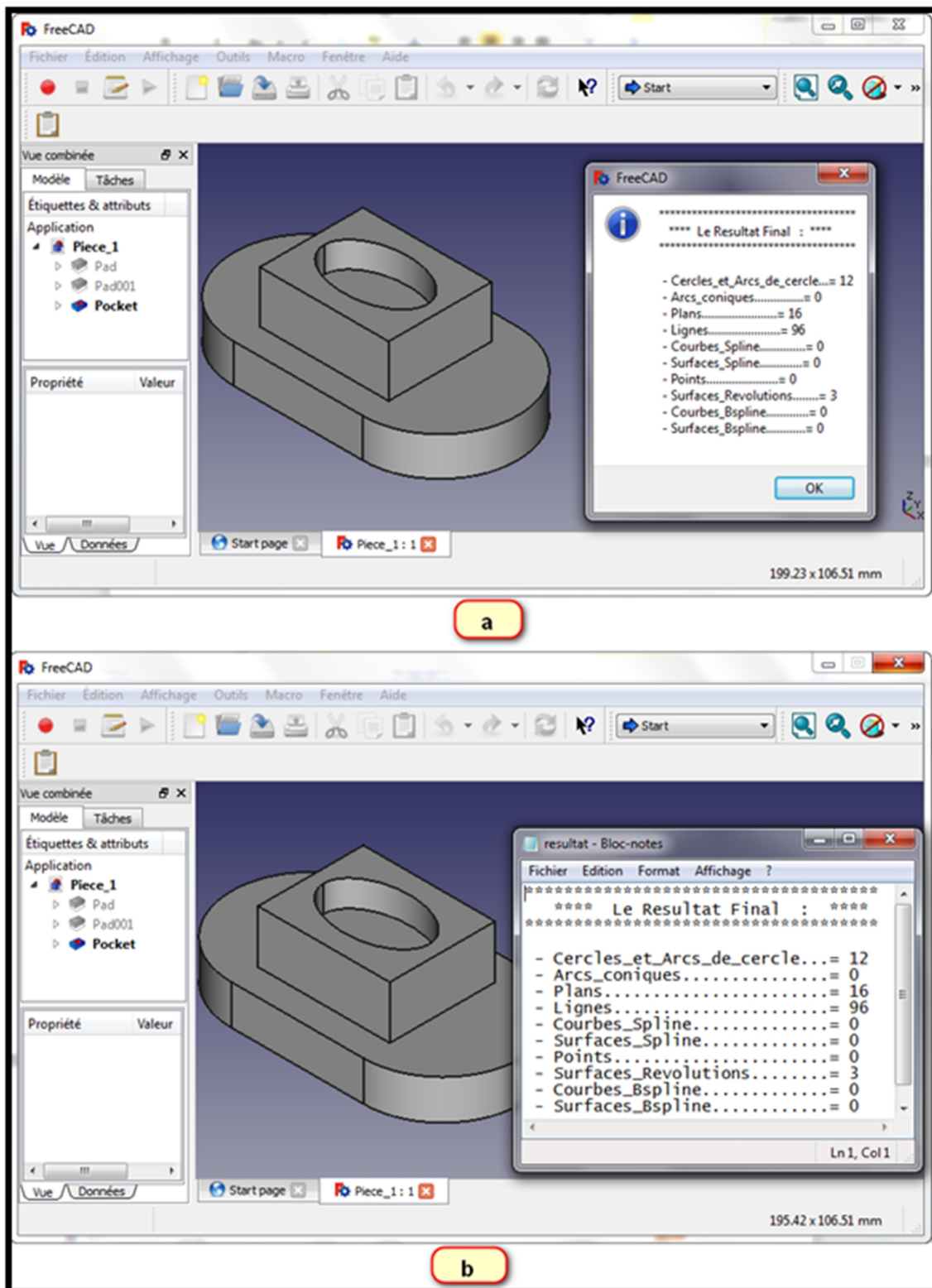


Figure III.15. L'affichage du résultat (Pièce_1)

III.5.1. Exemple d'une pièce complexe (avec des détails cachés) :

Dans cet exemple, nous appliquons la macro sur une pièce compliquée à la pièce dans l'exemple passé.

- Selon l'onglet **Fichier** clique sur **Ouvrir...**, puis sélectionner deuxième pièce (voir figure III.16. a).
- Maintenant la pièce deux est active (voir figure III.16. b).

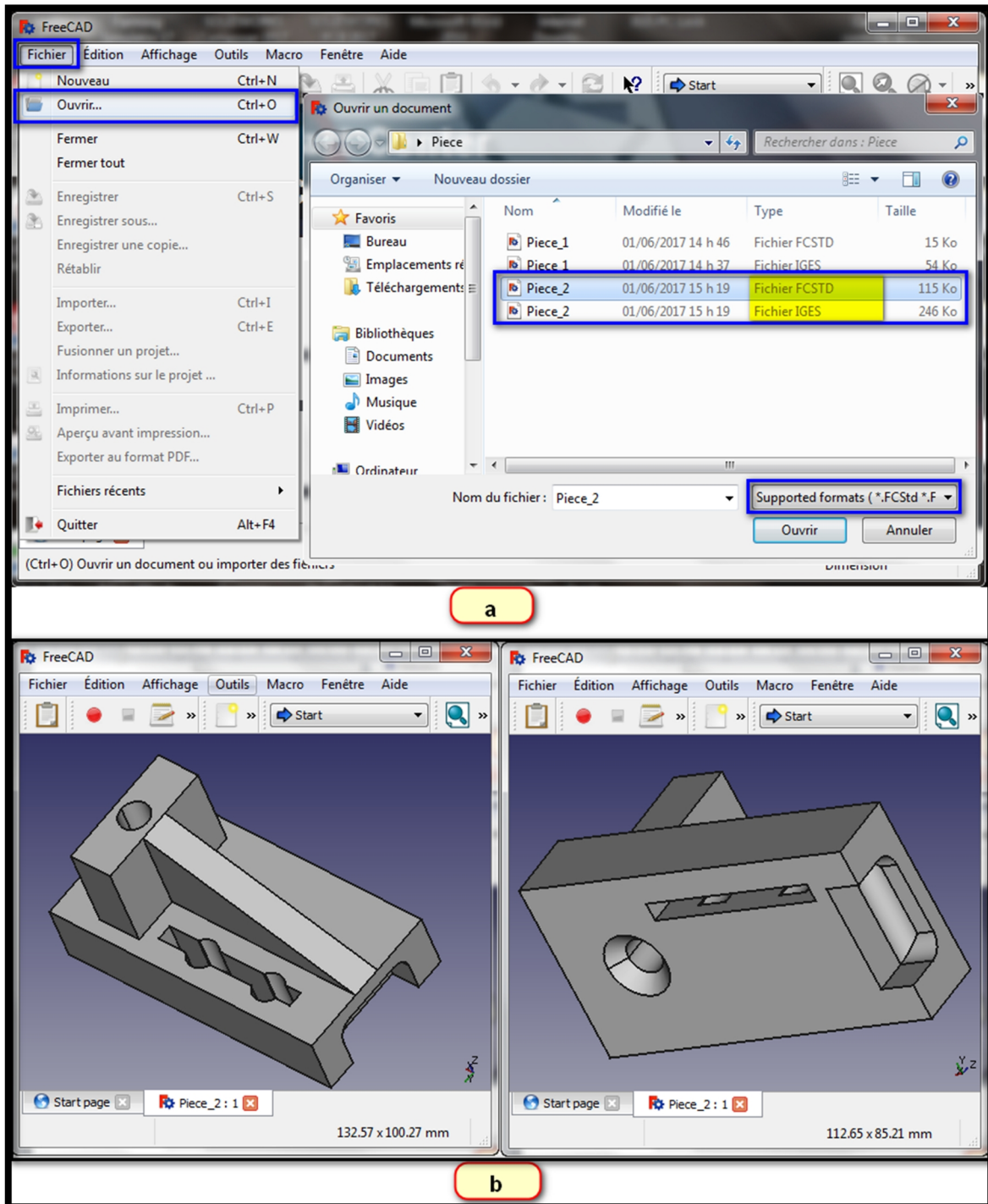


Figure III.16. Ouverture de la pièce (Pièce_2).

- Cliquer sur le bouton, une nouvelle fenêtre sera affichée qui contient les données de la pièce (voire figure III.17. a), puis cliquer sur bouton **OK** pour fermer la fenêtre, une autre fenêtre de Bloc-notes (fichier texte) contienne les mêmes données (voire figure III.17. b).

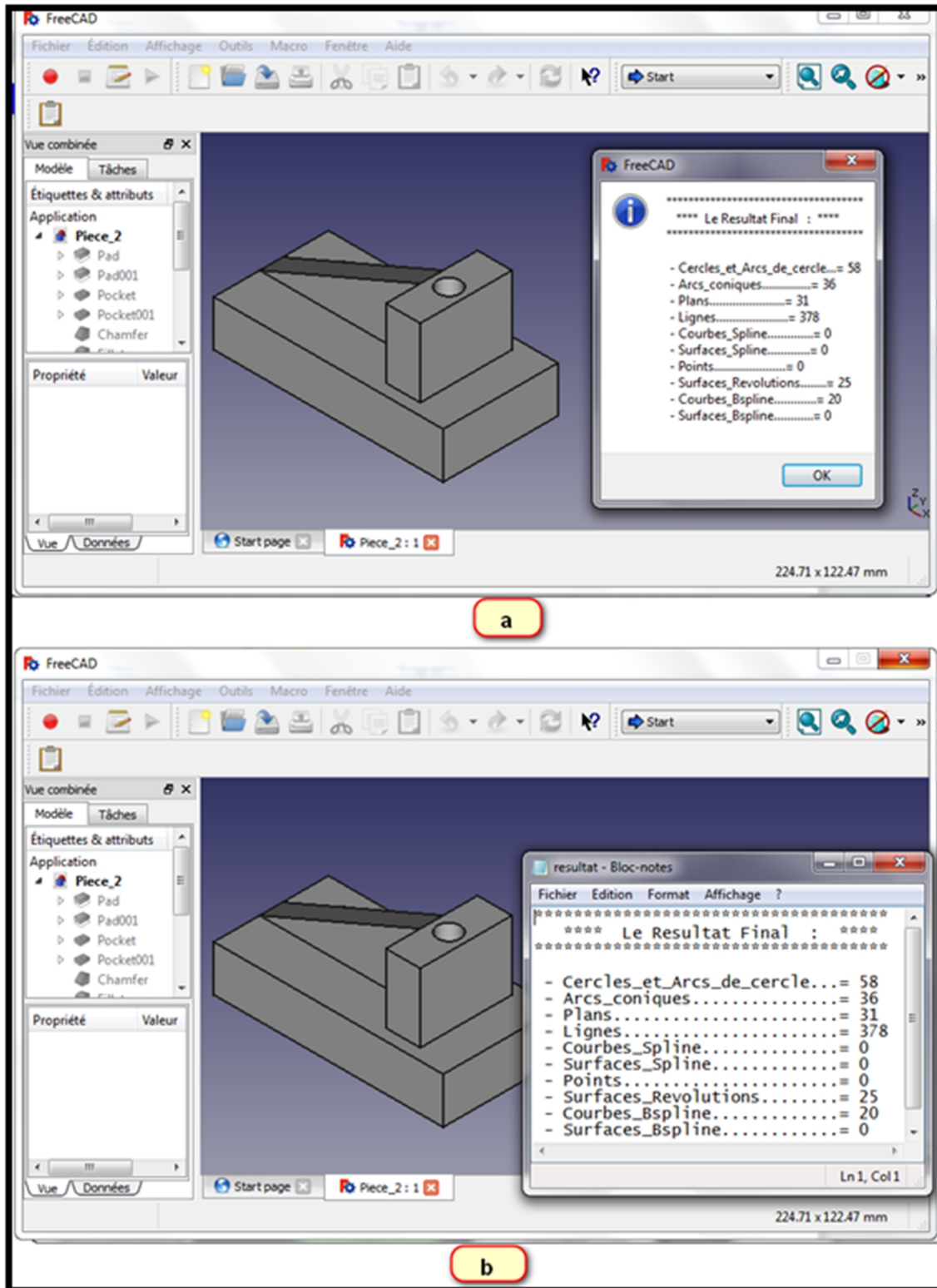


Figure III.17. Lancement de l'extraction des données (Pièce_2).

Conclusion générale

Conclusion générale

Dans ce mémoire, nous avons réalisé une fonctionnalité d'outil d'aide à la conception sous forme d'une macro intégrée au logiciel de modélisation FreeCAD. Notre programme permet d'extraire les données géométriques et topologiques du modèle géométrique d'une pièce mécanique à partir de son fichier IGES.

Pour réaliser notre travail, nous avons dû voir les différents types de modélisations géométriques dans le domaine de la CAO, la représentation topologique d'une pièce 3D ainsi que les formats de fichiers standards utilisés pour les échanges de données du modèle 3D des pièces en CFAO, plus particulièrement le format IGES que nous avons détaillé pour l'utiliser dans notre approche de programmation.

Au cours de notre mémoire, nous avons pu apprendre et maîtriser le langage de programmation Python ainsi que l'utilisation du logiciel de modélisation FreeCAD. Par la suite, nous avons utilisé le langage Python pour créer une Macro intégrant notre programme d'extraction de données géométriques et topologiques directement dans le logiciel FreeCAD.

Nous espérons que notre travail constitue une aide pour l'ingénieur du bureau de méthode afin d'avoir un outil capable de proposer une liste d'informations de types géométriques ou topologiques (nombre de surfaces planes, cylindrique,...) intégrée au logiciel de modélisation et en parallèle à la représentation 3D de la pièce, pour qu'il puisse déterminer les opérations d'usinage ou les machines nécessaires à la réalisation de la pièce.

Finalement, notre travail peut être considéré comme une recherche ouverte ; on peut proposer comme continuation à notre travail l'utilisation des données géométriques et topologiques pour identifier les « entités d'usinage » (voir annexe IV), (références Omar Zirmi), une fois les opérations d'usinage identifiées il restera à possible d'utiliser des méthodes d'optimisation pour les agencer et obtenir une gamme d'usinage automatique. L'autre proposition est d'élargir notre programme avec l'utilisation de formats de fichier différents d'IGES comme STEP ou DXF, et qui peuvent contenir d'autres types d'informations nécessaires au travail de l'ingénieur du bureau d'étude ou méthode. Un travail de recherche dans ce sens pourrait venir compléter notre travail.

Références

Bibliographiques

Bibliographie

- [1] M^{elle} CHAREB-YSSAAD Ismahane, ‘‘ Dessin Assisté par Ordinateur ’’, Master 1 Hydro-Informatique, Université Aboubekr Belkaid Tlemcen.
- [2] BORHEN louhichi, ‘‘ Intégration CAO/Calcul par reconstruction du modèle CAO à partir des résultats éléments finis ’’, Thèse de doctorat, Ecole Nationale d’Ingénieurs de Monastir, 5019 Monastir, 2008.
- [3] Cours de 1^{re} année Master, Production Automatique, université Ibn Khaldoun Tiaret, 2015-2016.
- [4] BENOIT Poupeau, ‘‘ Analyses et Requêtes de données géographiques 3D- Contributions de la cristallographie Géométrique – ’’, Thèse de doctorat, Université Paris-Est - Marne-La-Vallee Institut Géographique National, 2008.
- [5] VERONIQUE Moron, ‘‘ Mise En Correspondance De Données 3d Avec Un Modèle Cao : Application A L’inspection Automatique ’’, Thèse de doctorat, Institut national des sciences appliquées de Lyon, Villeurbanne, FRANCE (Université de soutenance), 1996.
- [6] THOMAS Guillod, ‘‘ Interpolations, courbes de Bézier et B-Splines ’’, Lycée Blaise-Cendrars sous la direction de Jean-Bernard Mathey, 2008.
- [7] JEAN CLOUDE Léon, ‘‘ Modélisation et construction de surfaces pour la CFOA ’’, édition Hermès, 1991.
- [8] : ROLAND Maranzana, ‘‘ Personnalisation des systèmes de CAO pour la mécanique (Modélisation géométrique) ’’, Le génie pour l’industrie (<https://www.etsmtl.ca>), 2014.
- [9] ROLAND Maranzana, ‘‘ Personnalisation des Systèmes de CAO Appliquée à la mécanique ’’,
- [10] ALAIN Plantec, ‘‘ Outils et méthodes de la norme STEP (Standard ISO 10303) ’’, Lab-STICC/CNRS UMR 6285 Université de Bretagne Occidentale.
- [11] YORIK VAN Havre et communauté FreeCAD, ‘‘ FreeCAD manual ’’, 2016
- [12] MAUDE Manouvrier, ‘‘ TP de Langage Python 3 ’’, Université Paris (DAUPHINE), 2014-2015.
- [13] GERARD Swinnen, ‘‘ Apprendre à programmer avec Python 3 ’’, éditions EYROLLES, Copyright (C) 2000-2010.
- [14] TAREK Ziadé, ‘‘ Programmation Python - Conception et optimisation - ’’, éditions EYROLLES, 2006, 2009.
- [15] PATRICK Terriault et JEAN-FRANÇOIS Chatelain, ‘‘ Développement de produits assisté par ordinateur _ Notes de cours (Partie CAO) ’’, Département de génie mécanique, rédigé : juillet 2008 et révisé : Août 2011.

Webographie

- [S1] <http://www.uqtr.ca/lirics/magic/ctu.html> (Consulté le 07/06/2017 à 11:45).
- [S2] <http://www.cad-magazine.com/content/les-formats-décharge-de-données> (consulté le 08/06/2017 à 12:36).
- [S3] <https://www.opencascade.com/doc/occt-7.1.0/overview/html/index.html> (Consulté le 02/03/2017 à 13:18).
- [S4] : https://www.opencascade.com/doc/occt7.1.0/overview/html/occt_dev_guides_tests.html (Consulté le 02/03/2017 à 11:27).
- [S5] https://freecadweb.org/wiki/About_FreeCAD/fr (Consulté le 19/04/2017 à 15:03).
- [S6] https://www.freecadweb.org/wiki/Part_Module/fr (Consulté le 30/05/2017 à 15:30).
- [S7] : https://www.freecadweb.org/wiki/FreeCAD_Scripting_Basics/fr (Consulté le 30/05/2017 à 15:48).

ANNEXES

Annexe I : Entités fichier IGES

Type	Identification	
	Français	Anglais
100	Arc circulaire	Circular Arc
102	Courbe composée	Composite Curve
104	Arc conique (Conic Arc)	Conic Arc
106	Copieux information	Copious Data
108	Plan	Plane
110	Ligne	Line
112	Courbes spline paramétrique	Parametric Spline Curve
114	Surfaces spline paramétrique	Parametric Spline Surface
116	Point	Point
118	Surface réglémentée	Ruled Surface
120	Surface de révolution	Surface of Revolution
122	Cylindre tabulé	Tabulated Cylinder
123	Direction	Direction
124	Matrice de transformation	Transformation Matrix
126	Courbe B-spline rationnelle	Rational B-Spline Curve
128	Surface de B-spline	Rational B-Spline Surface
130	Courbe décalée	Offset Curve
140	Surface décalée	Offset Surface
141	Frontière	Boundary
142	Courbe sur une surface paramétrique	Curve on a Parametric Surface
143	Surface réservée	Bounded Surface
144	Surface coupée de surface	Surface Trimmed Surface
150	Block (Block)	Block
152	Morceau angulaire droite	Right Angular Wedge
154	Cylindre circulaire droite	Right Circular Cylinder
156	Cone circulaire droite	Right Circular Cone
158	Sphère	Sphere
160	Tore	Torus
162	Solide de revolution	Solid of Revolution
164	Solide d'extrusion linéaire	Solid of Linear Extrusion

Annexe I : Entités fichier IGES

168	Ellipse (Ellipsoid)	Ellipsoid
180	Arbre booléenne	Boolean Tree
186	Objet de multiples solide B-Rep	Manifold Solid B-Rep Object
190	Surface plane	Plane Surface
192	Surface circulaire cylindrique droite	Right Circular Cylindrical Surface
194	Surface circulaire conique droite	Right Circular Conical Surface
196	Surface sphérique	Spherical Surface
198	Surface de Toroïdale	Toroidal Surface
308	Définition de sous-figure	Subfigure Definition
314	Définition du couleur	Color Definition
408	Cas de sous-figure singulier	Singular Subfigure Instance
502	Liste des sommets	Vertex List
504	Liste des bordures	Edge List
508	Boucle	Loop
510	Face	Face
514	Coquille	Shell

```
Cercles_et_Arcs_de_cercle = 0;Arcs_coniques = 0;Plans = 0;Lignes = 0;
Courbes_Spline = 0;Surfaces_Spline = 0;Points = 0;Surfaces_Revolutions = 0
Courbes_Bspline = 0;Surfaces_Bspline = 0;compteur=1
Fichier_IGES=open('E://Piece//Piece_1.iges','r')
lignes=[line.strip('\n')for line in Fichier_IGES.readlines()]
for line in lignes:
    if line.find('0D0') != -1:
        if line[5:8] == '100':
            Cercles_et_Arcs_de_cercle=Cercles_et_Arcs_de_cercle+1
        elif line[5:8] == '104':
            Arcs_coniques=Arcs_coniques+1
        elif line[5:8] == '108':
            Plans=Plans+1
        elif line[5:8] == '110':
            Lignes=Lignes+1
        elif line[5:8] == '112':
            Courbes_Spline=Courbes_Spline+1
        elif line[5:8] == '114':
            Surfaces_Spline=Surfaces_Spline+1
        elif line[5:8] == '116':
            Points=Points+1
        elif line[5:8] == '120':
            Surfaces_Revolutions=Surfaces_Revolutions+1
        elif line[5:8] == '126':
            Courbes_Bspline=Courbes_Bspline+1
        elif line[5:8] == '128':
            Surfaces_Bspline=Surfaces_Bspline+1
        else:
            print('aucune réponse sur la ligne : ',compteur)
        else:
            print('la ligne : ',compteur,'sans ampportance')
        compteur=compteur+1

Fichier_IGES.close()
```

```
print('Cercles_et_Arcs_de_cercle =', int(Cercles_et_Arcs_de_cercle/2))
```

```
print('Arcs_coniques =', int(Arcs_coniques/2))
```

```
print('Plans =', int(Plans/2))
```

```
print('Lignes =', int(Lignes/2))
```

```
print('Courbes_Spline =', int(Courbes_Spline/2))
```

```
print('Surfaces_Spline =', int(Surfaces_Spline/2))
```

```
print('Points =', int(Points/2))
```

```
print('Surfaces_Revolutions =', int(Surfaces_Revolutions/2))
```

```
print('Courbes_Bspline =', int(Courbes_Bspline/2))
```

```
print('Surfaces_Bspline =', int(Surfaces_Bspline/2))
```

```
from PySide import QtGui
import subprocess
import os
Cercles_et_Arcs_de_cercle = 0 ; Arcs_coniques = 0 ; Plans = 0 ; Lignes = 0
Courbes_Spline = 0 ; Surfaces_Spline = 0 ; Points = 0 ; Surfaces_Revolutions = 0 ;
Courbes_Bspline = 0 ; Surfaces_Bspline = 0 ; compteur=1
Ma_piece=str(Gui.ActiveDocument.Document.FileName)
Ma_piece_2=Ma_piece.replace('FCStd','iges')
Fichier_IGES=open(Ma_piece_2,'r')
lignes=[line.strip('\n')for line in Fichier_IGES.readlines()]
for line in lignes:
    if line.find('OD0') != -1:
        if line[5:8] == '100':
            Cercles_et_Arcs_de_cercle=Cercles_et_Arcs_de_cercle+1
        elif line[5:8] == '104':
            Arcs_coniques=Arcs_coniques+1
        elif line[5:8] == '108':
            Plans=Plans+1
        elif line[5:8] == '110':
            Lignes=Lignes+1
        elif line[5:8] == '112':
            Courbes_Spline=Courbes_Spline+1
        elif line[5:8] == '114':
            Surfaces_Spline=Surfaces_Spline+1
        elif line[5:8] == '116':
            Points=Points+1
        elif line[5:8] == '120':
            Surfaces_Revolutions=Surfaces_Revolutions+1
        elif line[5:8] == '126':
            Courbes_Bspline=Courbes_Bspline+1
        elif line[5:8] == '128':
            Surfaces_Bspline=Surfaces_Bspline+1
    else:
        print('')
```

```
else:
    print(' ')
    compteur=compteur+1
Fichier_IGES.close()
Fichier_resultat=open('E:/resultat.txt','w')
Fichier_resultat.write('*****'+'\n')
Fichier_resultat.write('**** Le Resultat Finale : ****'+'\n')
Fichier_resultat.write('*****'+'\n\n')
Fichier_resultat.write(' - Cercles_et_Arcs_de_cercle...=
'+str(Cercles_et_Arcs_de_cercle/2)+'\n')
Fichier_resultat.write(' - Arcs_coniques.....= '+str(Arcs_coniques/2)+'\n')
Fichier_resultat.write(' - Plans.....= '+str(Plans/2)+'\n')
Fichier_resultat.write(' - Lignes.....= '+str(Lignes/2)+'\n')
Fichier_resultat.write(' - Courbes_Spline.....= '+str(Courbes_Spline/2)+'\n')
Fichier_resultat.write(' - Surfaces_Spline.....= '+str(Surfaces_Spline/2)+'\n')
Fichier_resultat.write(' - Points.....= '+str(Points/2)+'\n')
Fichier_resultat.write(' - Surfaces_Revolutions.....= '+str(Surfaces_Revolutions/2)+'\n')
Fichier_resultat.write(' - Courbes_Bspline.....= '+str(Courbes_Bspline/2)+'\n')
Fichier_resultat.write(' - Surfaces_Bspline.....= '+str(Surfaces_Bspline/2)+'\n')
Fichier_resultat.close()
Affichier_resultat=open('E:/resultat.txt','r')
ligne_1=Affichier_resultat.readline()
ligne_2=Affichier_resultat.readline()
ligne_3=Affichier_resultat.readline()
ligne_4=Affichier_resultat.readline()
ligne_5=Affichier_resultat.readline()
ligne_6=Affichier_resultat.readline()
ligne_7=Affichier_resultat.readline()
ligne_8=Affichier_resultat.readline()
ligne_9=Affichier_resultat.readline()
ligne_10=Affichier_resultat.readline()
ligne_11=Affichier_resultat.readline()
ligne_12=Affichier_resultat.readline()
ligne_13=Affichier_resultat.readline()
```

```
ligne_14=Affichier_resultat.readline()
ligne_15=Affichier_resultat.readline()
reply = QtGui.QMessageBox.information(None,"Information",ligne_1+ligne_2+ligne_3
+ligne_4+ligne_5+ligne_6+ligne_7+ligne_8+ligne_9+ligne_10+ligne_11+ligne_12
+ligne_13+ligne_14+ligne_15)
Affichier_resultat.close()
chemin_du_Blocnots = 'C:\\Windows\\System32\\notepad.exe'
```

Il existe différents types d'entité comme par exemple :

- Les entités de forme : Elles contiennent de la géométrie explicite et décrivent des formes génériques, souvent propres à un domaine d'application donné.
- Les entités de fabrication : Elles regroupent les informations sur la forme des objets, sur les procédés de fabrication (usinage, moulage, soudure), sur les outils (type, chemin d'accès), etc.
- Les entités de tolérance : Elles caractérisent l'erreur de déformation admissible de la forme (taille).
- Les entités fonctionnelles : Elles correspondent à l'ensemble des caractéristiques relatives à des fonctions spécifiques pouvant contenir des intentions de conception, des attributs non-géométriques, des paramètres de performance, etc.

Les entités d'usinage, sous-classe des entités de fabrication, nous intéressent tout particulièrement ici, bien qu'il existe d'autres travaux définissant des entités selon des critères propres à d'autres domaines tels que la fonderie, la forge le moulage par injection, l'analyse par éléments finis ou la tôlerie. L'intégration des entités d'usinage au modèle CAO se fait selon deux méthodes; concevoir directement la pièce à partir d'entité ou bien identifier les entités à partir d'un modèle CAO. La conception par entité n'est pas évidente, surtout que le concepteur raisonne du point de vue fonctionnel en manipulant de la géométrie qui ne correspond pas toujours à des entités d'usinage, car il conçoit avec des extrusions, des balayages et des opérations booléennes entre solides. Par contre, l'usinage est caractérisé par des opérations d'enlèvement de matière. D'autre part, l'interaction entre entités d'usinage altère souvent la sémantique individuelle des entités.

Techniques de reconnaissance automatique des entités d'usinage

Les techniques de reconnaissance automatique des entités d'usinage divise en six groupes de base : volumique, heuristique, symbolique, topologique, centré processus, et hybride (**Figure 1**). Chacune de ces méthodes se subdivise en sous classes. Nous allons détailler quelques-unes de ces approches.

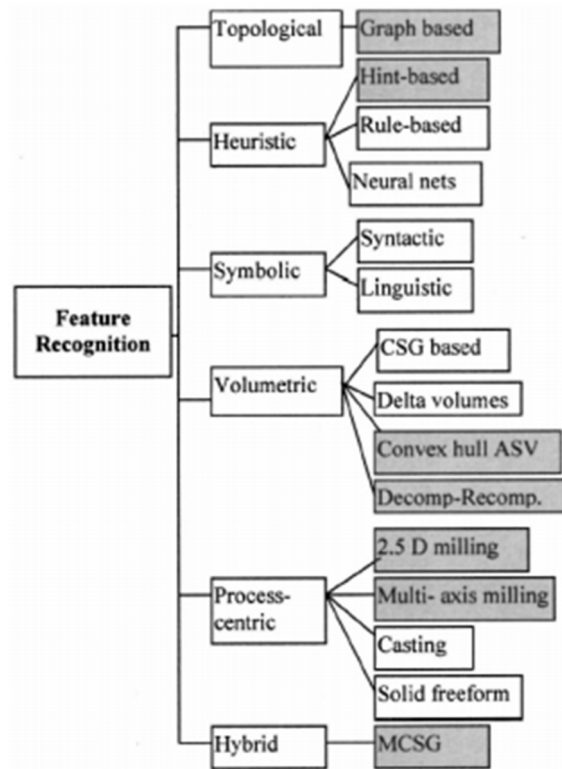


Figure 1 : Taxonomie des techniques de reconnaissance d'entités.

Les approches volumiques

Les approches volumiques calculent un brut englobant la totalité de la pièce, puis essaient de reconstituer la pièce d'origine en identifiant les volumes à enlever au brut pour obtenir la pièce finie. On classe ces approches en trois catégories : les méthodes par décomposition convexe, les méthodes par décomposition cellulaire et les méthodes basées sur un arbre CSG.

• Les méthodes par décomposition convexe

C'est une méthode algorithmique qui décompose un volume en lui soustrayant ses composants convexes. Au début, on crée un volume convexe (VC1) de la pièce, après la soustraction du volume VC1 de la totalité du volume de la pièce, il résulte un volume V_p^* (figure 2). Par la suite le volume convexe du volume V_p^* est créé et on réapplique l'étape précédente. Le processus est répété jusqu'à l'obtention d'un volume égale à zéro. Les volumes obtenus seront utilisés pour générer des entités d'usinage. Le problème de cette méthode c'est que les composants résultant ont des formes qui ne peuvent pas toujours être directement appliquées pour générer des entités d'usinage, et la convergence n'est pas toujours assurée (figure 2).

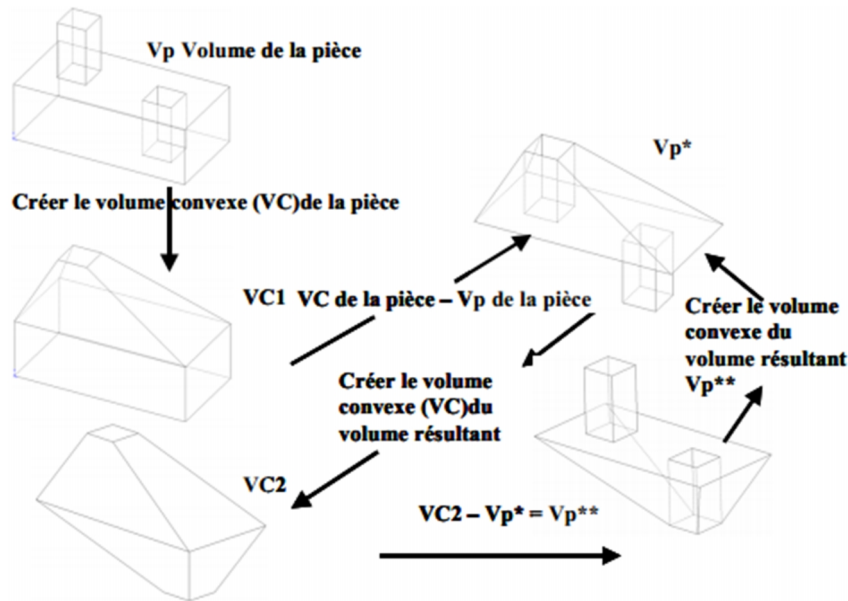


Figure 2 : Décomposition en enveloppes convexes.

- **Les méthodes de décomposition/recomposition cellulaire ou la décomposition en delta volume**

Elle consiste à déterminer le volume total de la matière à usiner puis le décomposer en un ensemble de petites cellules indépendantes qui correspondent à des opérations d'usinage distinctes. Puis ces cellules seront recomposées pour former des entités d'usinage. La décomposition d'une forme complexe en formes convexes simples (des cellules). Dans une méthode pour décomposer un polyèdre en un maximum d'enveloppes convexes est décrit. Le procédé est décrit comme capable de décomposer n'importe quel polyèdre en un maximum d'enveloppes convexes. La méthode de décomposition est appliquée aux deltas volumes pour produire un maximum de cellules convexes. Le delta volume est défini comme le volume de matière à usiner, qui est obtenu par une opération booléenne entre le brut et le modèle de la pièce (Figure). Ces cellules sont alors employées pour produire des interprétations multiples d'entités. Cette approche a été également étendue aux modèles ayant des surfaces cylindriques et planes.

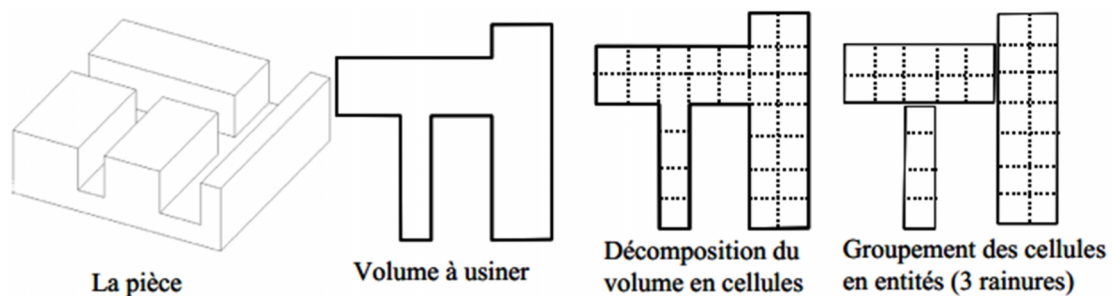


Figure 3 : Décomposition cellulaire.

Les approches topologiques

Ces approches de reconnaissance sont basées sur l'isomorphisme des sous graphes du modèle B-Rep est déduit un graphe de la pièce. Les faces sont les nœuds du graphe tandis que les relations entre faces forment les arcs du graphe d'adjacence de faces (FAG en anglais, pour face d'adjacency graph). Le FAG complété par des informations additionnelles devient alors un AAG (Attributed face Adjacency Graph). Indique qu'une entité est d'abord modélisée sous forme de graphe qui représente les contraintes topologiques et géométriques qui la caractérisent. Ensuite on cherche le sous graphe qui lui correspond le mieux dans la structure FAG de la pièce en utilisant les algorithmes issus de la théorie des graphes.

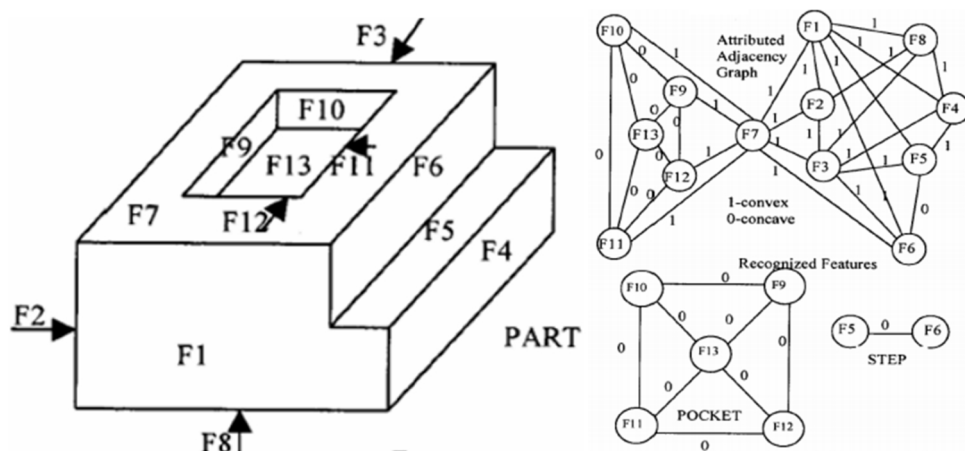


Figure 4 : Exemple d'un AAG graphe d'une poche fermée.

Les approches heuristiques

Ces approches peuvent être basées sur les règles, qui analysent directement un modèle par les biais de règles de reconnaissance et de liaison. Une liaison est un indice appartenant à la géométrie nominale, c'est l'indicateur de l'existence possible d'une entité. Ayant plus de liaisons que d'entités à reconnaître ces méthodes doivent faire face à de multiples interprétations éventuelles.

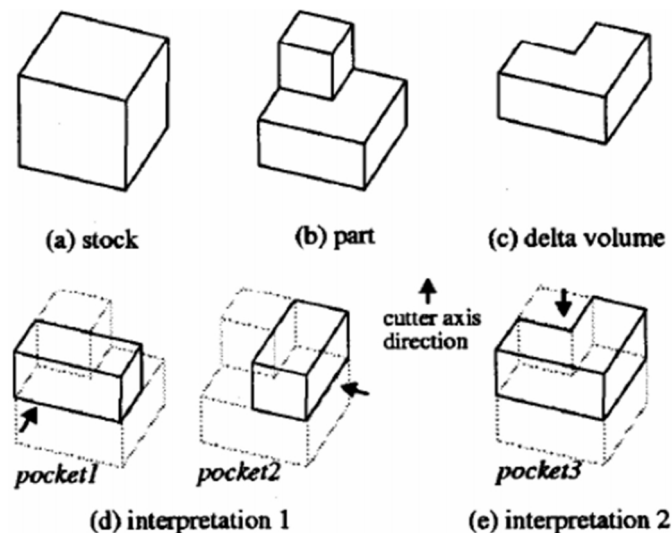
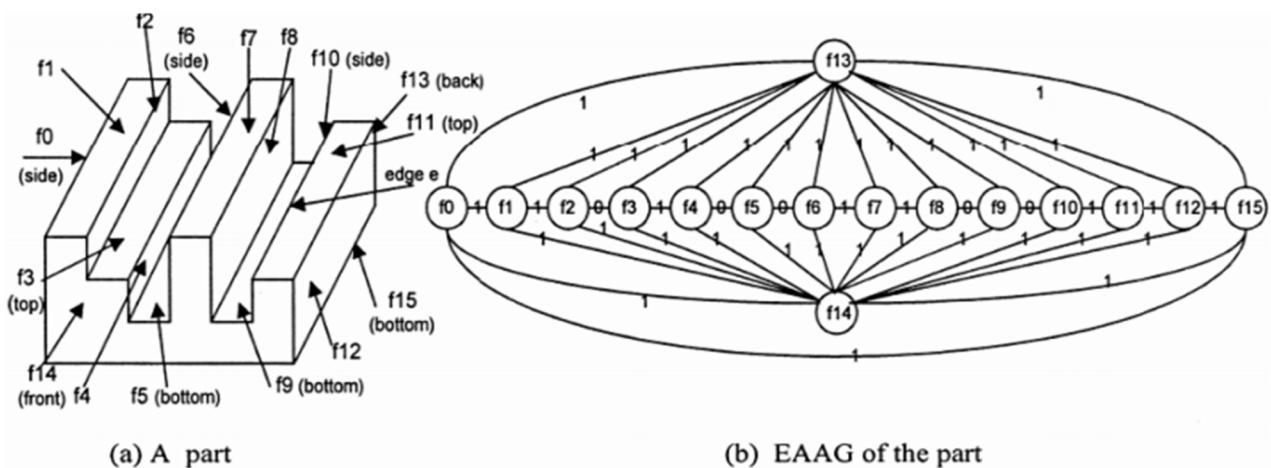


Figure 5 : Le problème d'interprétations multiples**Les approches hybrides**

Les approches hybrides mélangent les approches préalablement citées. On essaie de remédier aux inconvénients de certaines approches en comblant ce manque par les résultats d'une autre. Combine une méthode de reconnaissance basée sur les graphes avec une reconnaissance par traces. Les traces sont définies à l'aide d'un graphe d'adjacence avec attributs étendus EAAG (Extended AAG – détails dans les approches topologiques) complété par l'adjonction de liens virtuels correspondant aux entités disparues par interactions. Pour reconnaître les entités, on extrait des sous graphes MFAG (Manufacturing Face Adjacency Graph) du graphe EAAG. L'étude des MFAG permet de repérer les interactions entre les différentes entités. Pour reconstruire les entités en interaction, on construit le MCSG (Minimal Condition Sub-Graph) issu de l'EAAG de la pièce. Ce MCSG est par la suite utilisé comme une trace par une approche heuristique.

**Figure 6** : Exemple de graphe EAAG d'une pièce.

Résumé

La Conception et Fabrication Assistées par Ordinateur (CFAO) occupe une place fondamentale au sein des entreprises industrielles, pour un meilleur compromis coût/délai/qualité, et pour assurer une meilleure interopérabilité entre les différents métiers. Dans ce contexte, notre travail a pour but de développer une application qui permet d'extraire les données géométriques et volumiques d'un modèle 3D, qui permettra à l'ingénieur du bureau de méthode d'avoir un outil d'aide à la conception capable de proposer une liste d'informations (nombre de surfaces planes, cylindrique,..., arêtes rectilignes...) en parallèle au modèle 3D de la pièce, afin qu'il puisse déterminer les opérations d'usinage ou les machines nécessaires à la réalisation de la pièce. Ceci représente la première étape vers le développement d'une application de génération de gamme d'usinage automatique.

ملخص:

التصميم و التصنيع بمساعدة الكمبيوتر (CFAO)، يحتل مكانة أساسية ضمن الشركات الصناعية من أجل توافق أفضل بين التكلفة، الوقت و نوعية أفضل، وضمان تحسين إمكانية التشغيل المتداخل بين مهن مختلفة، عملنا يهدف لتطوير تطبيق يسمح باستخراج بيانات هندسية وحجمية لنموذج ثلاثي الأبعاد، الذي سوف يسمح لمهندس لمكتب المناهج أن تكون له أداة تصميم قادرة على اقتراح قائمة بالمعلومات (عدد السطوح المستوية، أسطوانية، الحواف المستقيمة...)، بالتوازي مع النموذج الثلاثي الأبعاد للقطعة، حتى أنه يمكن تحديد عمليات القطع أو الآلات اللازمة لإنشاء القطعة. وهذا يمثل خطوة أولى نحو تطوير تطبيق إنشاء آلي للتشكيلة الميكانيكية.