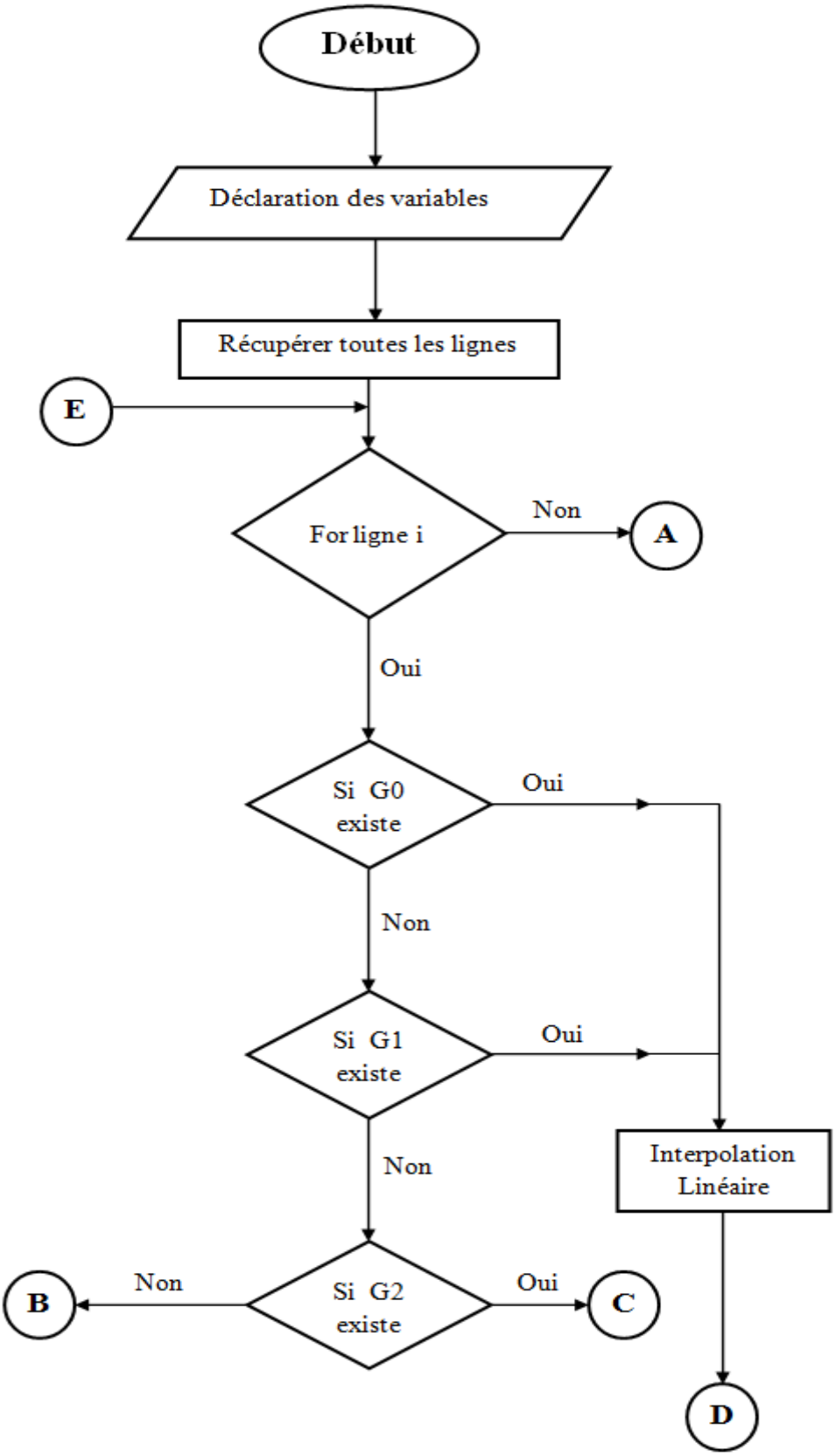
III.1. Introduction

Dans ce dernier chapitre, nous allons vous présenter l'algorithme que nous avons mis en place pour extraire les données géométriques relatives à la génération de trajectoire à partir du fichier G-code et celui qui permet de tracer ces trajectoires en superposition sur la pièce correspondante qui provient du fichier au format 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 et de génération de trajectoires

Premièrement, on veut réaliser un programme qui permet de lire et d'extraire les données (type d'interpolation et coordonnées de déplacement de l'outil « (x, y, z), (i, j, k) ») à partir d'un fichier G-code (voir figure III.1), puis enregistrer le résultat dans un fichier d'une façon compréhensible par l'utilisateur ; enfin enregistrer seulement les coordonnées dans un autre fichier sous la forme d'un tableau pour l'utiliser comme une base de données.


Début
Déclaration des variables
Récupérer toutes les lignes
E
For ligne i
Non
A
Oui
Si G0 existe
Oui
Non
Si G1 existe
Oui
Non
Interpolation Linéaire
Si G2 existe
Non
B
Oui
C
D

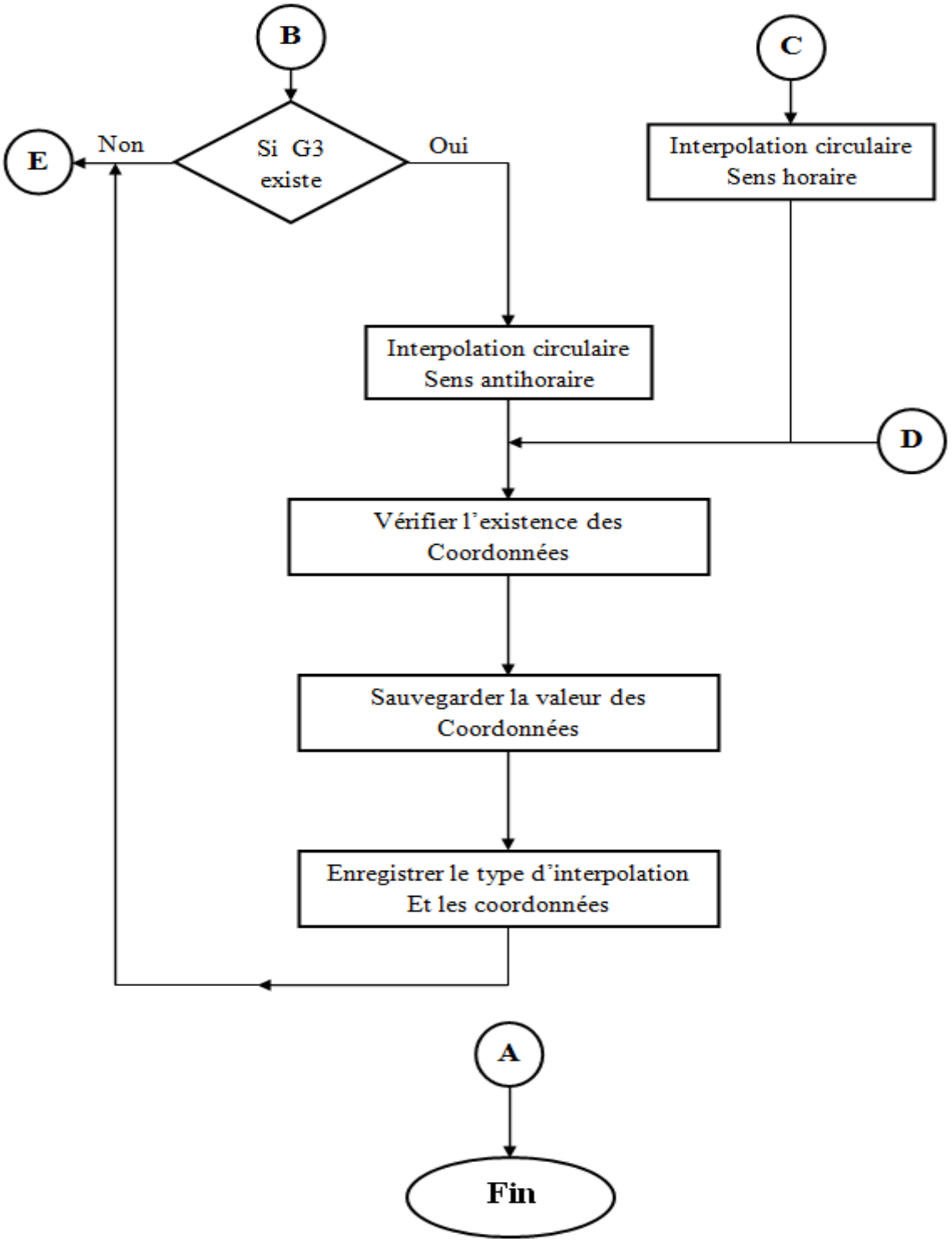


Figure III.1. Algorithme d'extraction des données géométriques des trajectoires à partir d'un fichier G-code.

III.3. Programme sur python

Une fois que l'algorithme d'extraction de données mis en place et vérifier, 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 G-code indépendamment du logiciel FreeCAD.

L'intégralité du code de programmation en langage Python est donnée dans l'annexe B.

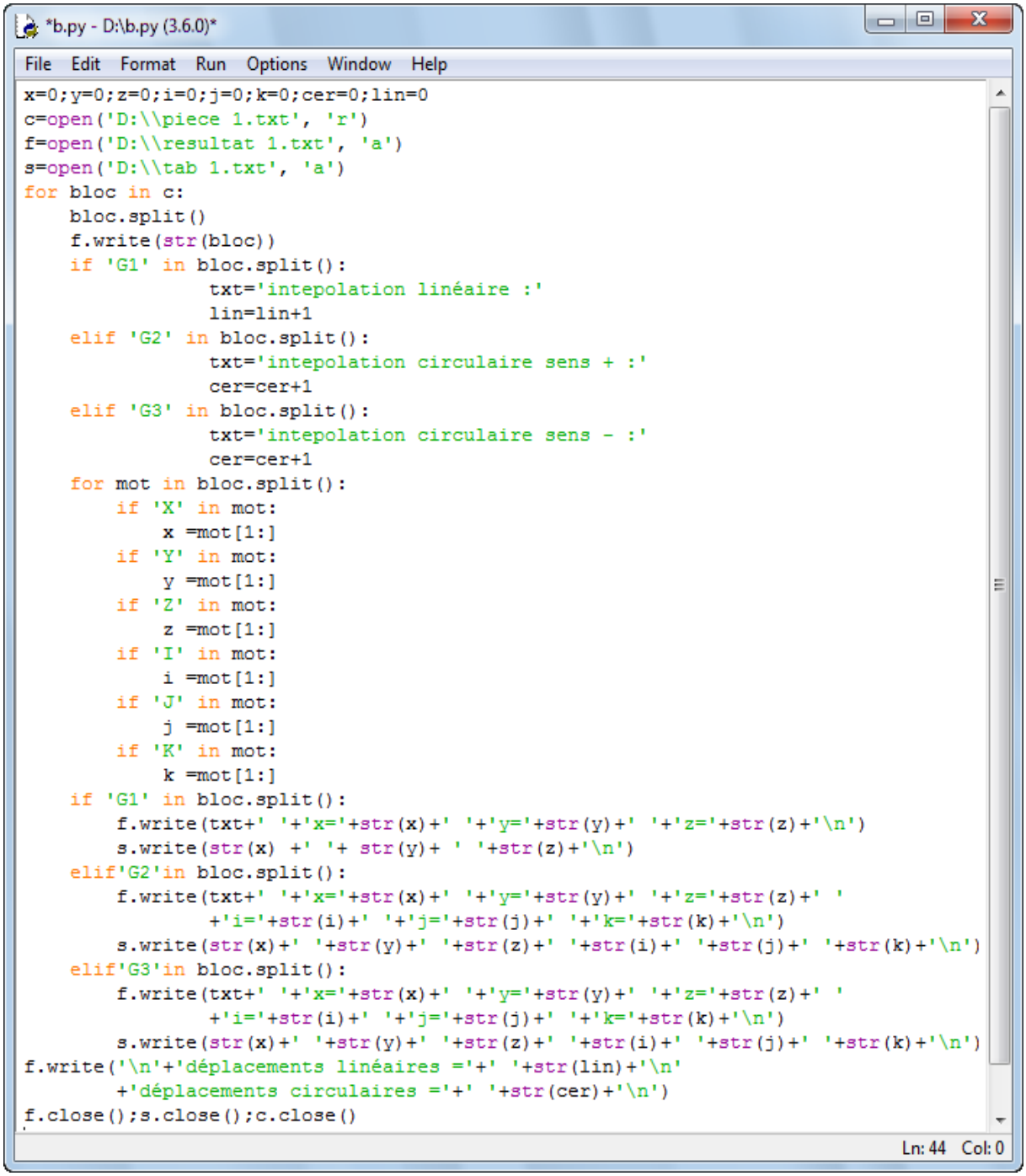


Figure III.2. Programme en langage Python

Pour exécuter le programme, on appuis simplement sur le bouton F5 pour lancer l'exécution. Lorsqu'on exécute le programme sur le fichier de la figure III.3, le résultat sera comme dans les figures (III.4 et III.5).

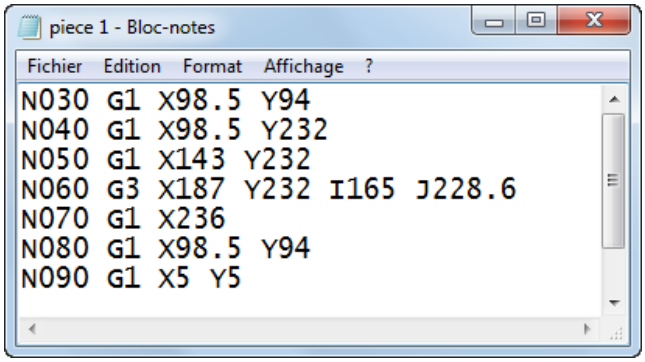


Figure III.3. Fichier texte d'un G-code

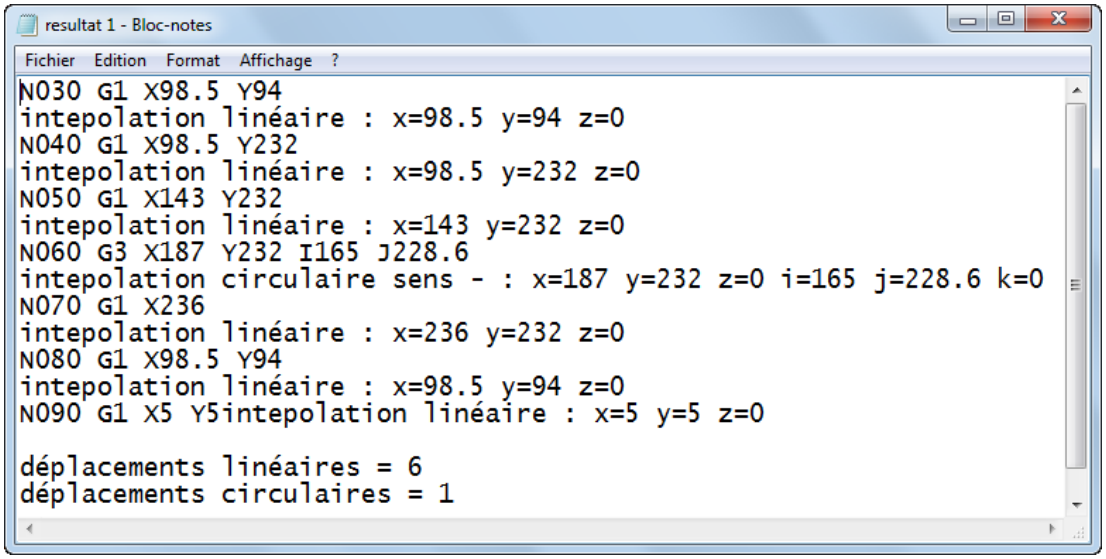


Figure III.4. Résultat du programme sous format compréhensible par l'utilisateur

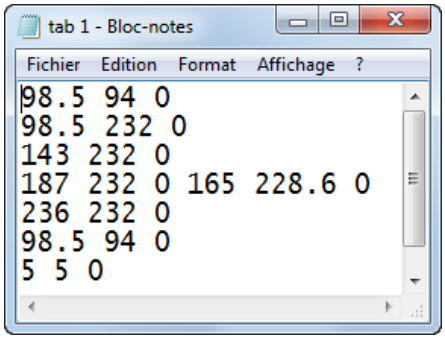


Figure III.5. Résultat du programme sous format de base de données pour le traitement

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 Génération des trajectoires dans le logiciel FreeCAD

Dans le paragraphe précédent nous avons créé un programme qui lie le fichier G-code pour extraire les données, le résultat de ce programme était un fichier texte qui contient l'ensemble des données relatives aux trajectoires de l'outil.

Dans ce paragraphe nous allons créer un programme similaire au précédent sous la forme d'une macro, la différence est que l'objectif de cette macro est de générer la trajectoire d'outil a partir des données extraites.

Pour cela nous allons suivre les étapes suivantes :

- Importer les modules nécessaires :

Nous avons d'abord besoin d'importer le module **Part** afin que nous puissions utiliser son contenu Python.
Nous allons également importer le module **Base** à l'intérieur du module de FreeCAD.
Nous avons besoin aussi d'importer le module **math** pour la mesure des angles.
En plus, nous importons le module **sqrt** à l'intérieur du module de math.

```
import Part
from FreeCAD import Base
import math
from math import sqrt
```

- Déclarer les variables :

```
x = 0; y = 0; z = 0; i = 0; j = 0; k = 0; Xi = 0; Yi = 0; Zi = 0
```

- Récupérer le chemin du document actif : (l'extension du document doit être « FCSTD »).

```
chemin1=str(Gui.ActiveDocument.Document.FileName)
```

- Ouvrir le fichier G-code : On peut ouvrir le fichier G-code en changeant seulement l'extension dans le même chemin de la pièce. (le fichier G-code doit être dans le même répertoire que la pièce).

```
chemin2=chemin1.replace("FCStd","txt")
file=open(chemin2, "r")
```

- Extraire les données géométriques dans le fichier G-code : s'est la même méthode utilisée dans le programme présenté précédemment.

```
for bloc in file:
    for mot in bloc.split():
        if 'X' in mot:
            x = mot[1:]
        if 'Y' in mot:
            y = mot[1:]
        if 'Z' in mot:
            z = mot[1:]
        if 'I' in mot:
            i = mot[1:]
        if 'J' in mot:
            j = mot [1:]
       if 'K' in mot:
           k = mot[1:]
```

- Si **G0** est trouvée dans la ligne : Définir les coordonnées du point de départ du déplacement suivant (les coordonnées qui se trouvent dans cette ligne reçoivent les coordonnées de départ du déplacement suivant).

```
if  'G0'  in  bloc.split() :
    Xi = x;  Yi = y;  Zi = z
```

- Si **G1** est trouvée dans la ligne : Créer une ligne ; pour créer une ligne on a besoin de deux points (point de départ et point d'arrivé). On utilise le point de la ligne précédente comme point de départ et le point de la ligne présente comme point d'arrivé, puis on affiche le résultat de la forme la ligne a l'écran, et toujours on défini les coordonnées du point de départ du déplacement suivant.

```
if  'G1'  in  bloc.split() :
    ligne = Part.makeLine ((Xi, Yi, Zi) , (x, y, z))
    Part.show(ligne)
    Xi = x;  Yi = y;  Zi = z
```

- Si **G2** existe dans la ligne : Créer un arc, on peut créer un arc en donnant le rayon, le centre, l'axe de rotation, l'angle de départ et l'angle de la fin.
  Pour mesurer un angle dans FreeCAD on a besoin de créer un vecteur horizontal son début est le centre de l'arc, pour l'utiliser comme une référence de mesure des angles :

```
arc1= Part.makeCircle (rayon, centre, axe de rotation, angle de départ, angle de fin)
```

Avec :

Rayon = rayon de l'arc (cercle).

Centre = coordonnées du centre de l'arc (cercle).

Axe de rotation = axe Z (0 ,0 ,1)

Angle de départ et angle de fin (voir figure III.6).

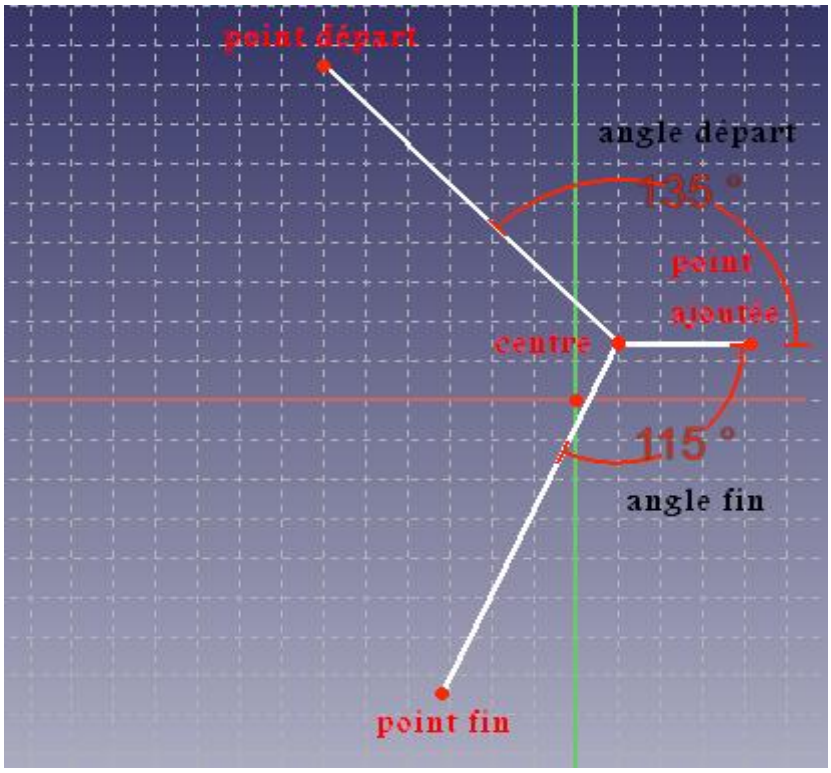


Figure III.6. Différents paramètres de la fonction « arc » dans FreeCAD

Comme se présentée dans la figure III.6, on a besoin d'ajouter un vecteur horizontal son début est le centre de l'arc, pour l'utiliser comme une référence de mesure des angles :

Pour obtenir la valeur de l'angle de départ et de l'angle d'arrivé, on utilise la méthode suivante :

```
centre = Base.Vector(float(i),float(j),float(k))
point départ = Base.Vector(float(Xi),float(Yi),float(Zi))
point arrivée = Base.Vector(float(x),float(y),float(z))
point ajoutée H = Base.Vector(float(i)+15,float(j),float(k))
Vecteur1 = centre.sub(point départ)
Vecteur2 = centre.sub(point arrivée)
Vecteur3 = centre.sub(point ajoutée)
Angle départ = Vecteur1.getAngle(Vecteur3)
Angle arrivée = Vecteur1.getAngle(Vecteur2)
```

Si veut créer un arc **G2** sur l'exemple précédant, avec les coordonnées suivantes : Point de départ (-40, 65, 0) ; Point d'arrivée (-20, -50, 0) ; Centre (10, 15, 0) ; point ajoutée (10+15, 15, 0).

Le résultat sera comme dans la figure suivant (figure III.7) :

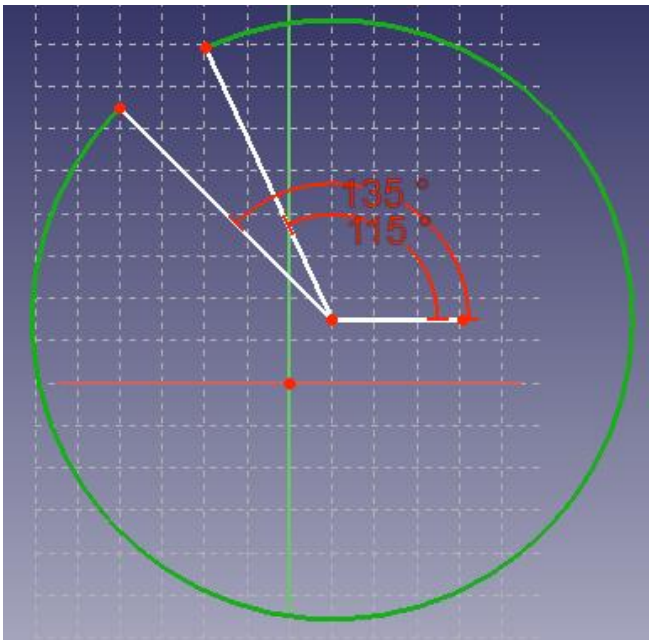


Figure III.7. Dessin d'un « arc » dans FreeCAD sans gérer le sens des angles

L'arc qui convient à ses coordonnées est dans la figure III.8 ci-dessous :

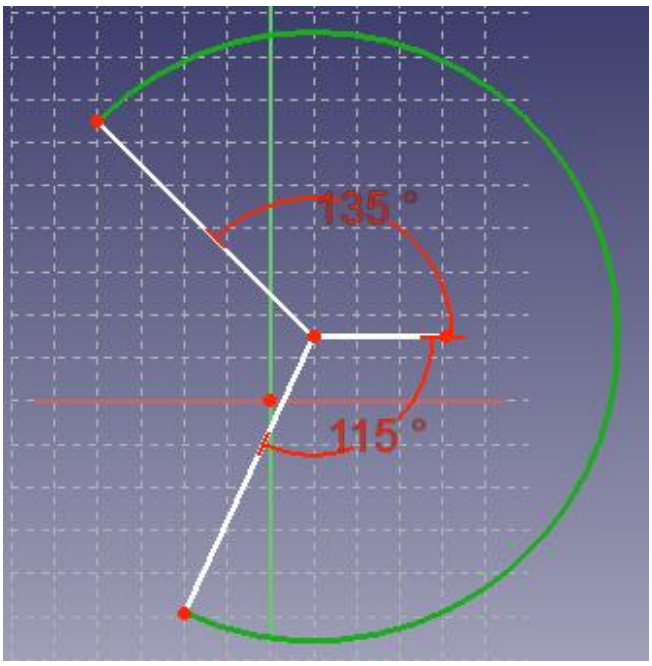


Figure III.8. Dessin d'un « arc » dans FreeCAD avec gestion du sens des angles

La cause de ce problème d'affichage est que la mesure des angles prend deux sens de mesure des angles entre les vecteurs, mais la création des arcs prend un seul sens (sens antihoraire).

La solution de ce problème est de créer un vecteur verticale et changer l'angle de départ et l'angle de fin selon le position de point de départ et le point de fin par rapport au deux vecteurs créer horizontal et vertical :

```
Point ajoutée V = Base.Vector(float(i),float(j)+15,float(k))
Vecteur4 = centre.sub(point ajoutée V)
angle3 = Vecteur1.getAngle(Vecteur4)
angle4 = Vecteur2.getAngle(Vecteur4)
```

- Calculer le rayon de l'arc :

```
rxi =( float(Xi) - float(i))**2
ryj =( float(Yi) - float(j))**2
Rayon = sqrt (rxi + ryj)
```

- Déterminer l'angle de départ et l'angle de fin : selon les vecteurs horizontal et vertical :

```
if float(angle4) <= 90:
  if float(angle3) <= 90:
    angle de départ = angle 2 ; angle de fin = angle 1
  else :
    angle de départ = angle 2 ; angle de fin = 360-angle 1
else :
  if float(angle3) <= 90:
    angle de départ = 360-angle 2 ; angle de fin = angle 1
  else :
    angle de départ = 360-angle 2 ; angle de fin = 360-angle 1
```

- Créer l'arc et mise en forme de l'arc a l'écran : maintenant on peut créer l'arc en utilisant la méthode précédente.
- Définir les coordonnées du point de départ du déplacement suivant

```
Xi = x;  Yi = y;  Zi = z
```

- Si que **G3** est trouvée dans la ligne : Créer un arc ; ce sont les mêmes étapes que dans le cas de G2 en changeant seulement les angles de départ et les angles de fin :

```
if float(angle4) <= 90:
  if float(angle3) <= 90:
    angle de départ = angle 1 ; angle de fin = angle 2
  else :
    angle de départ = angle 1 ; angle de fin = 360-angle 2
else :
  if float(angle3) <= 90:
    angle de départ = 360-angle 1 ; angle de fin = angle 2
  else :
    angle de départ = 360-angle 1 ; angle de fin = 360-angle 2
```

- Afficher l'ensemble du contenu a l'écran :

```
Gui.SendMsgToActiveView("ViewFit")
```

- Placer l'ensemble en vue axonométrique :

```
Gui.activeDocument().activeView().viewAxonometric()
```

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

III.4.2. Personnaliser la barre d'outils

Maintenant, nous voulons créer un nouveau bouton sur la barre d'outils FreeCAD qui sert à lancer la macro sans entrer dans les détails. Ce bouton nous permet de lancer la macro d'une façon très simple avec une seule clique sur le bouton dans la même fenêtre de la pièce active.

Pour créer un nouvel bouton dans FreeCAD, il nécessaire de suivre les étapes suivantes :

- Cliquez sur **Menu → Outils → Personnaliser**

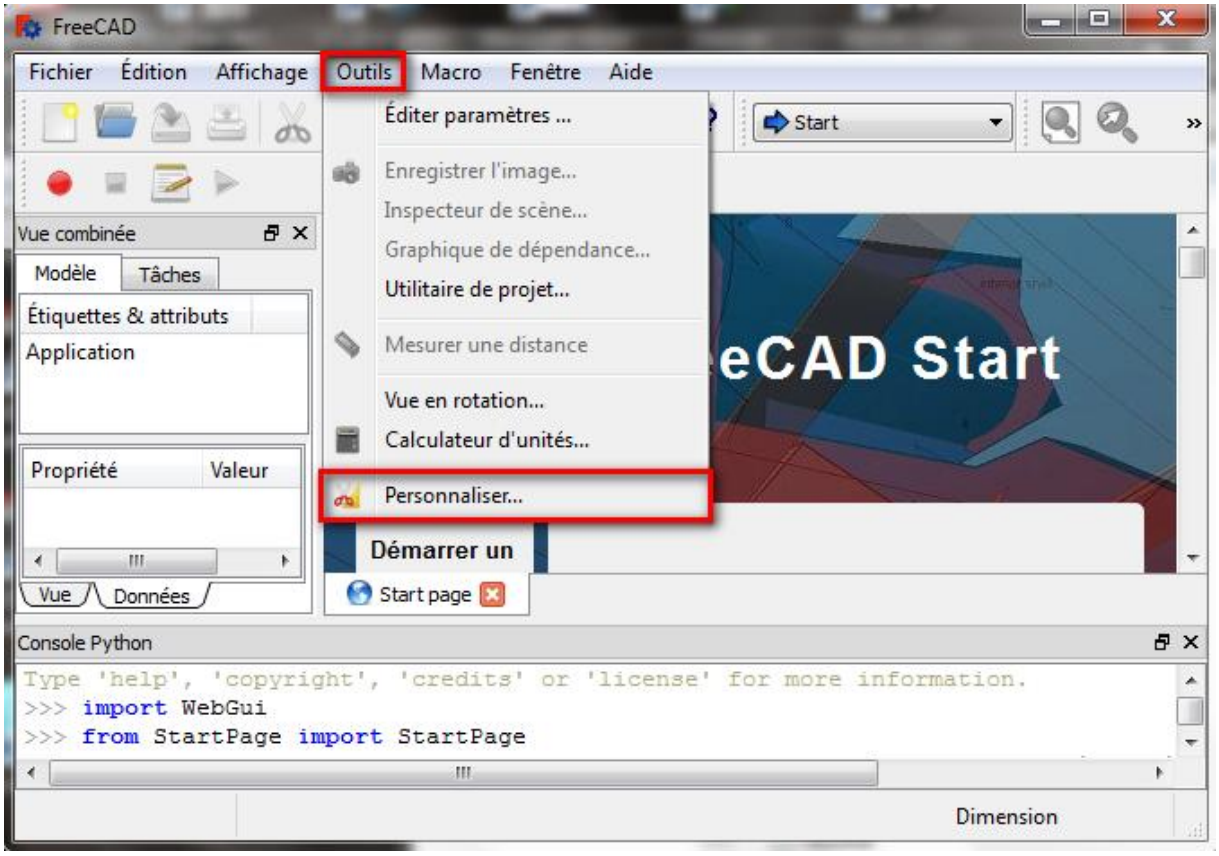


Figure III.9. Création de la Bouton

- La fenêtre "Personnaliser" s'affiche

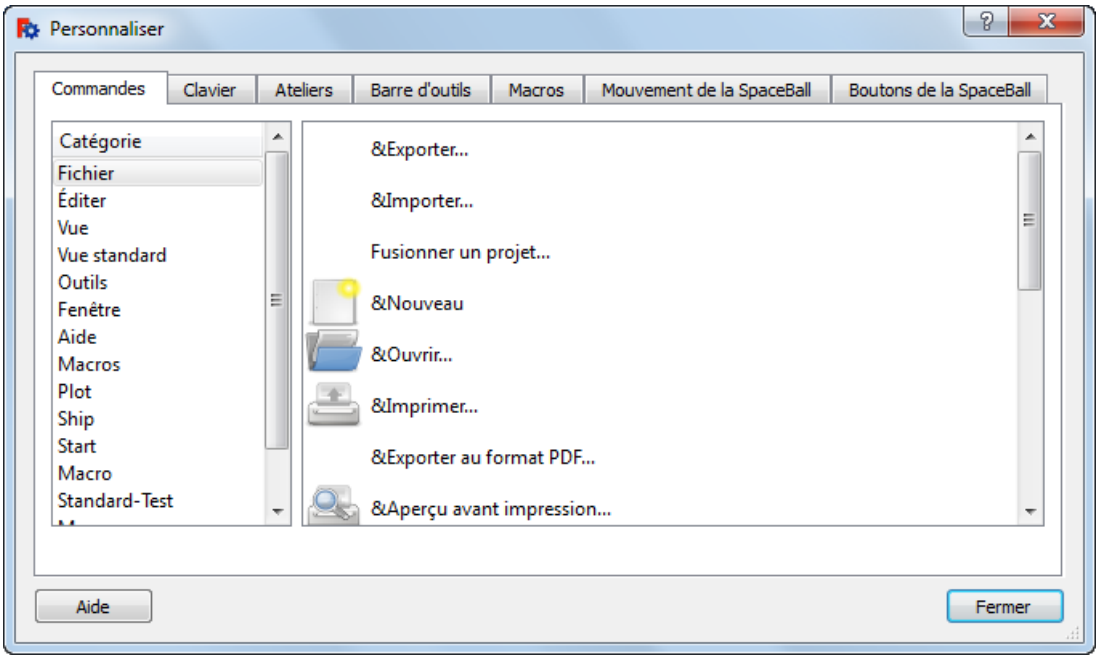


Figure III.9.a

- Sélectionnez l'onglet Macro
- Cliquez sur le bouton .... pour sélectionner un icône pour votre macro

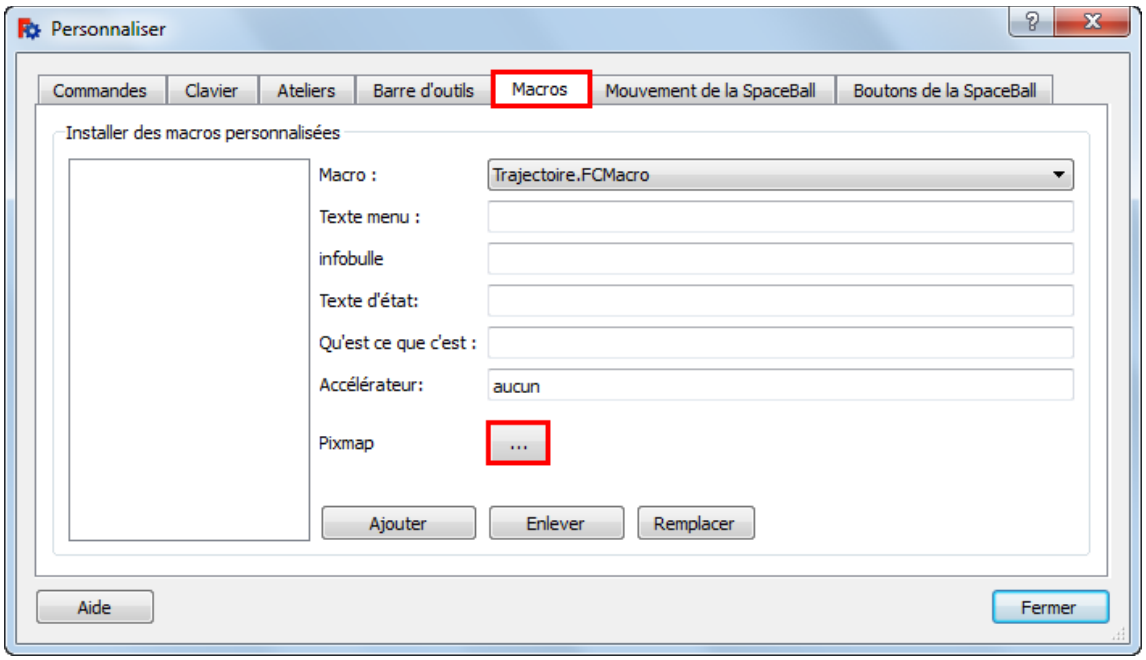


Figure III.9.b

- Cherchez et sélectionnez dans la liste un icône pour votre macro.
- Sélectionnez votre icône et cliquez sur le bouton OK pour valide.

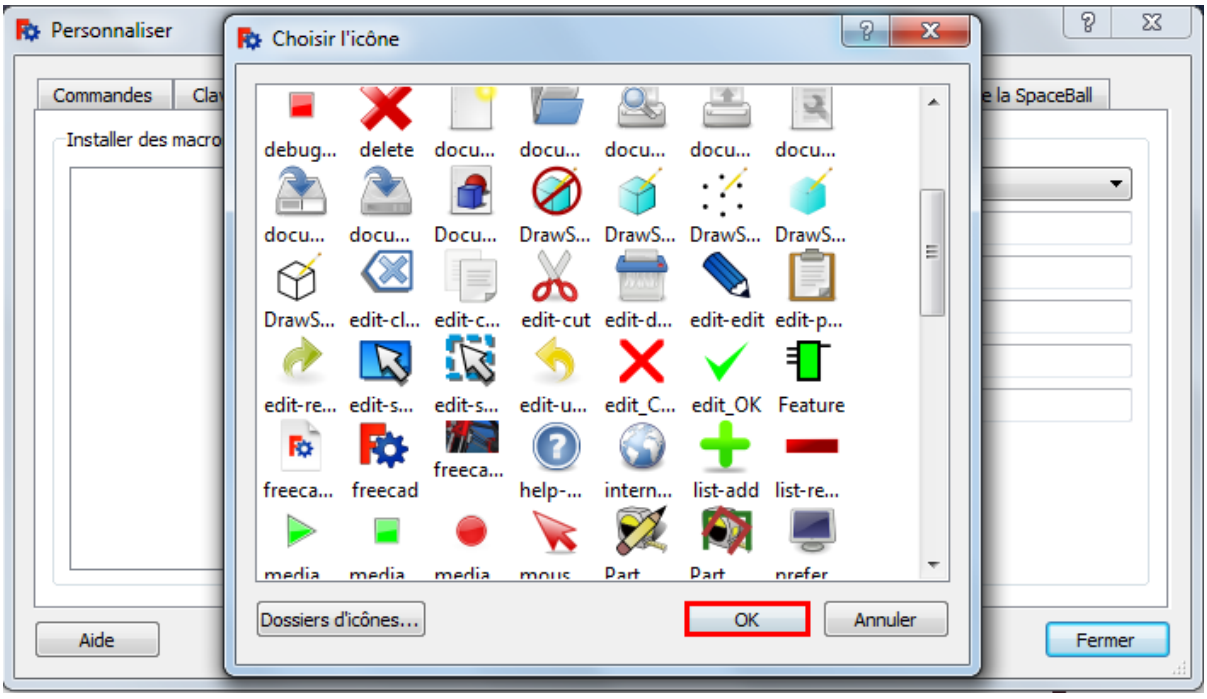


Figure III.9.c

- L'icône sélectionné est maintenant affiché, cliquez sur le bouton trois petits points ....

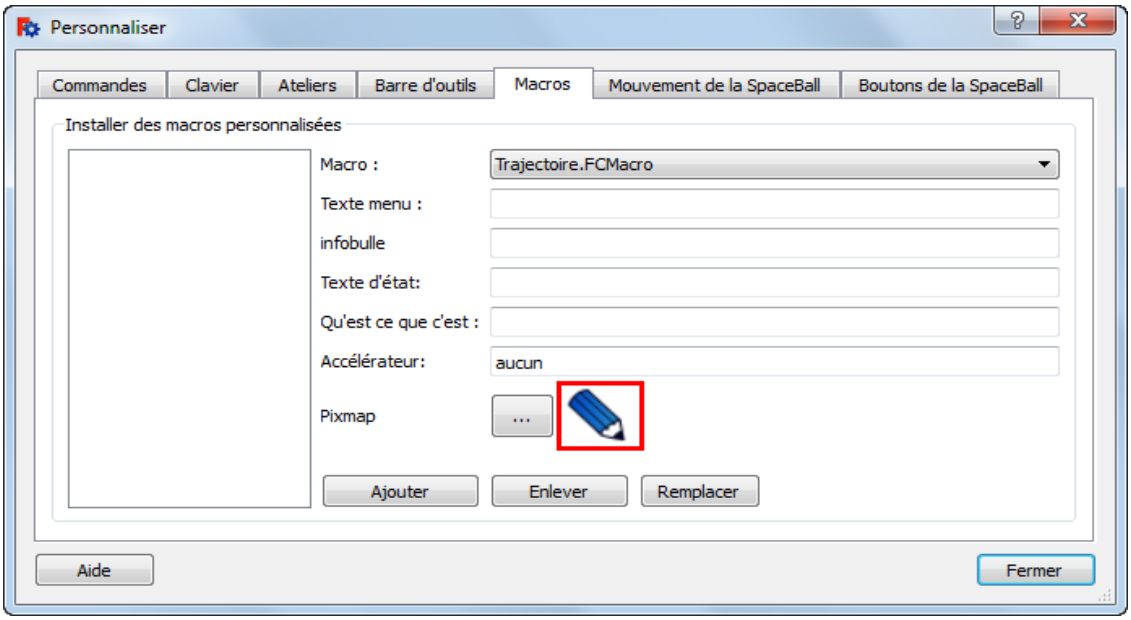


Figure III.9.d

- Choisissez votre macro et spécifiez un texte de menu (ce texte apparaîtra comme l'étiquette dans le menu); remplissez 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 votre bouton qui sont facultatifs mais que vous pouvez remplir.
- Cliquez sur le bouton Ajouter

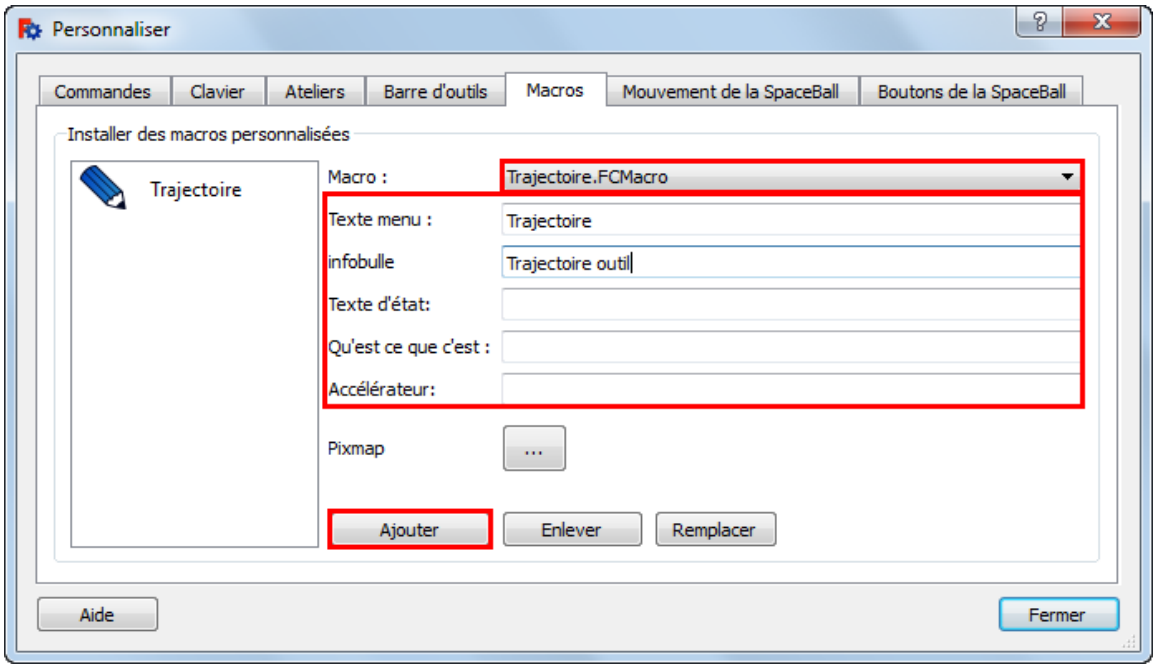


Figure III.9.e

- Votre bouton est maintenant créé :

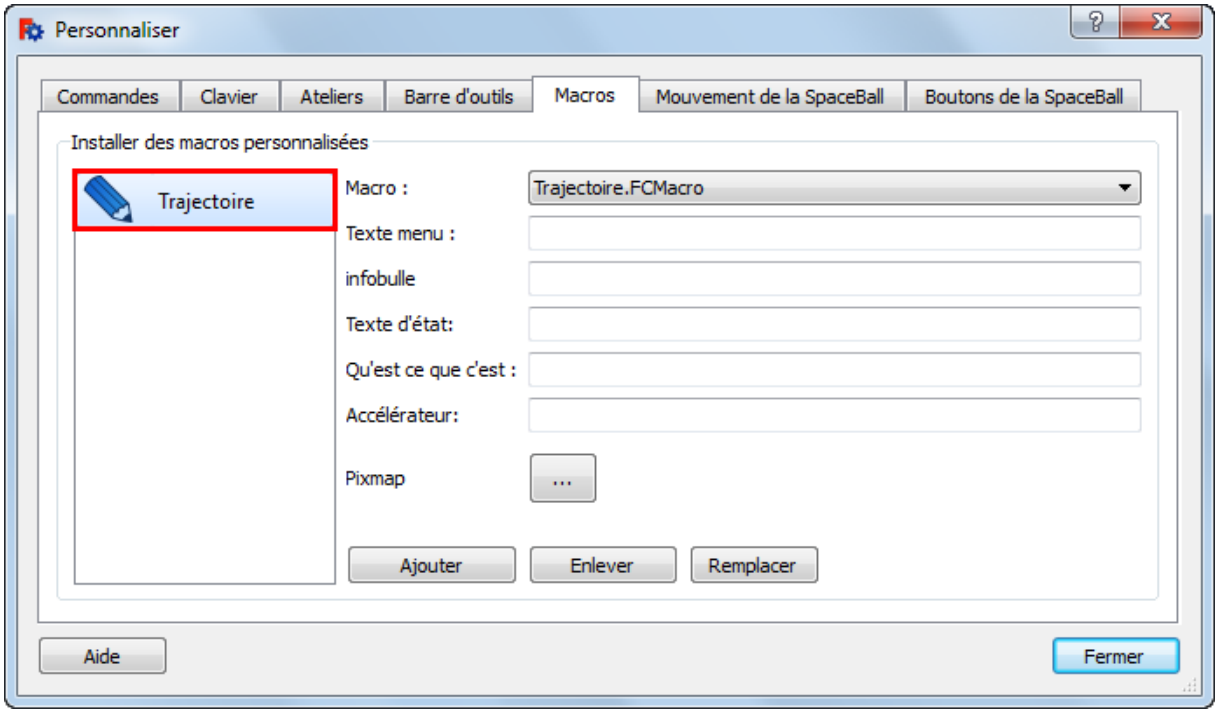


Figure III.9.f

- Nous allons 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** (ici **Start**).

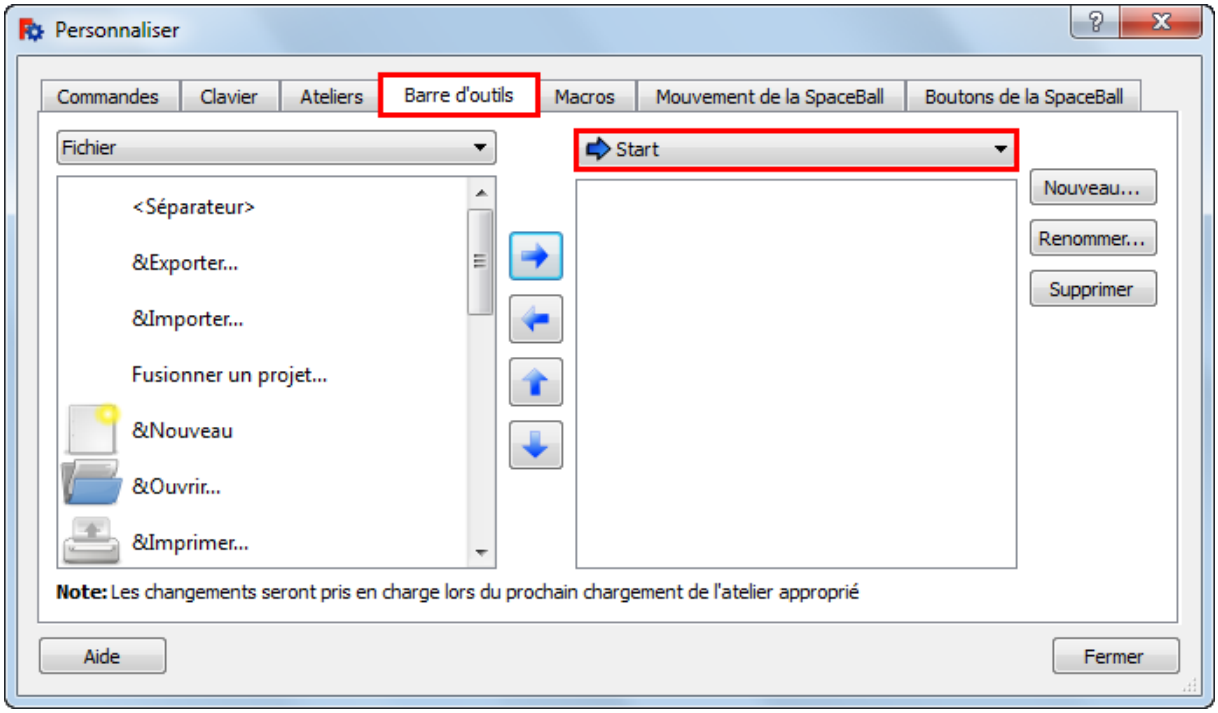


Figure III.10. Création d'une Barre d'outils pour le bouton

- Dans la fenêtre de gauche, sélectionnez l'onglet **Macros** et **Macro.**

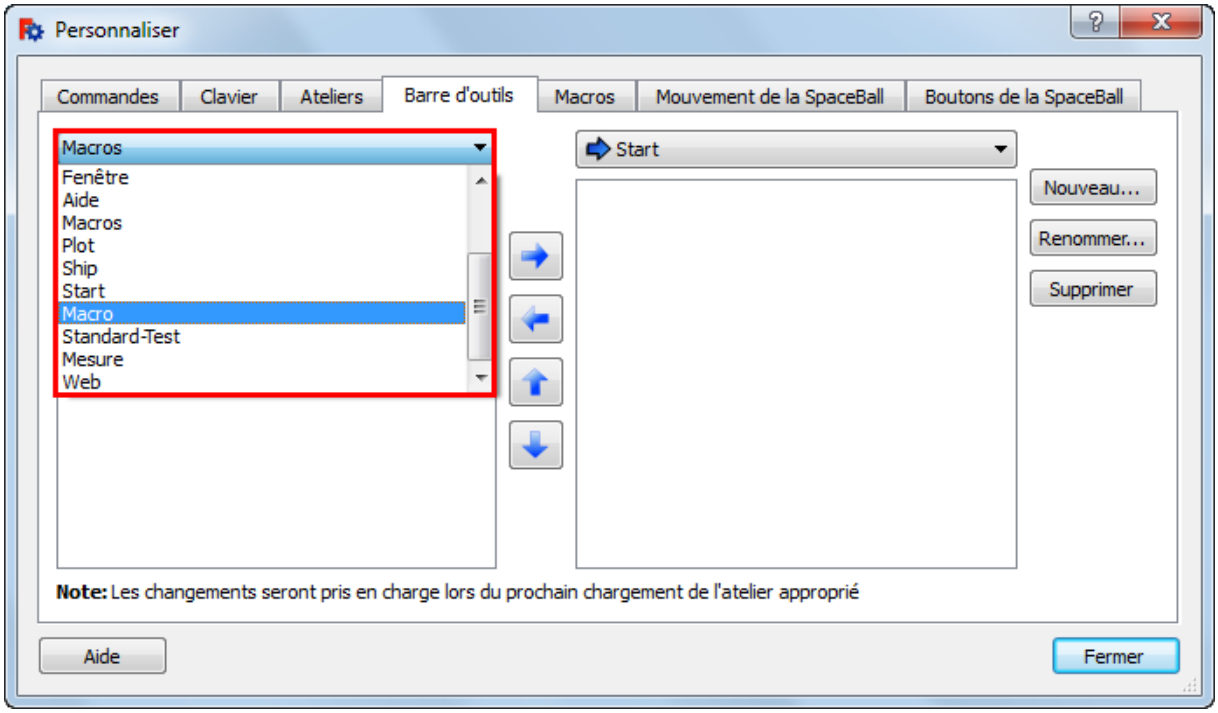


Figure III.10.a

- Maintenant votre macro et son icône apparaissent.

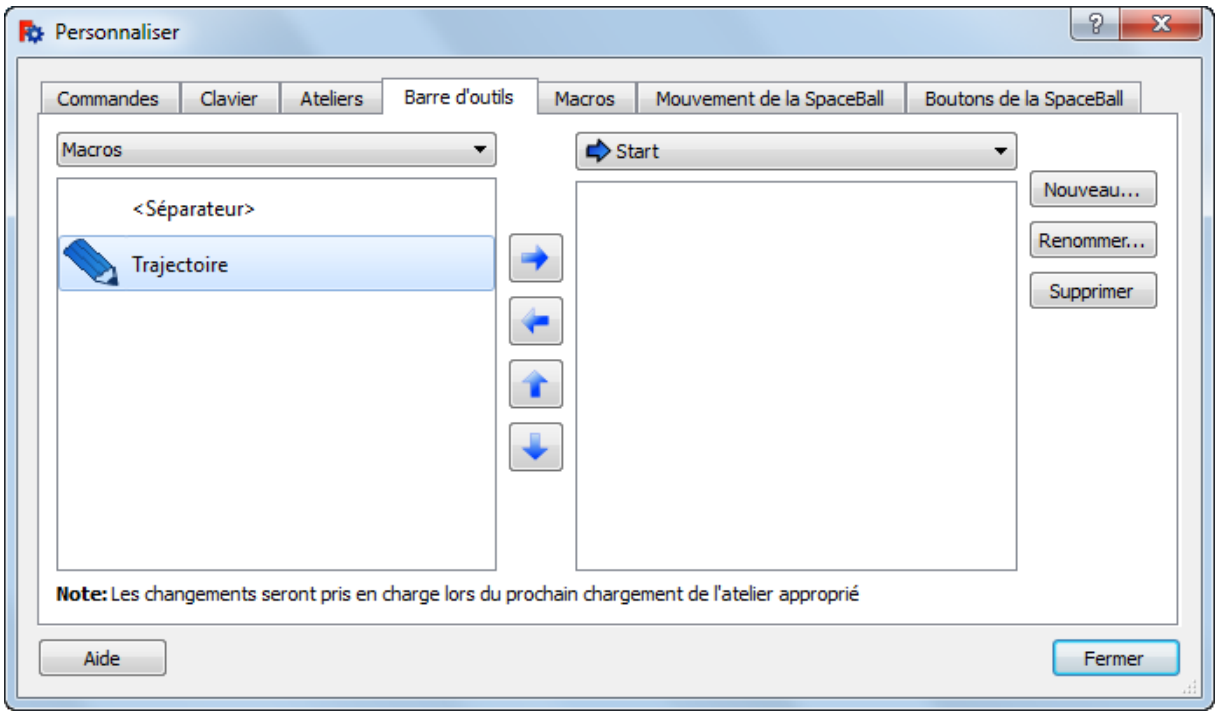


Figure III.10.b

- Cliquez le bouton Nouveau... (choisissez votre barre d'outils ici votre barre d'outils est attachée à l'atelier Start).

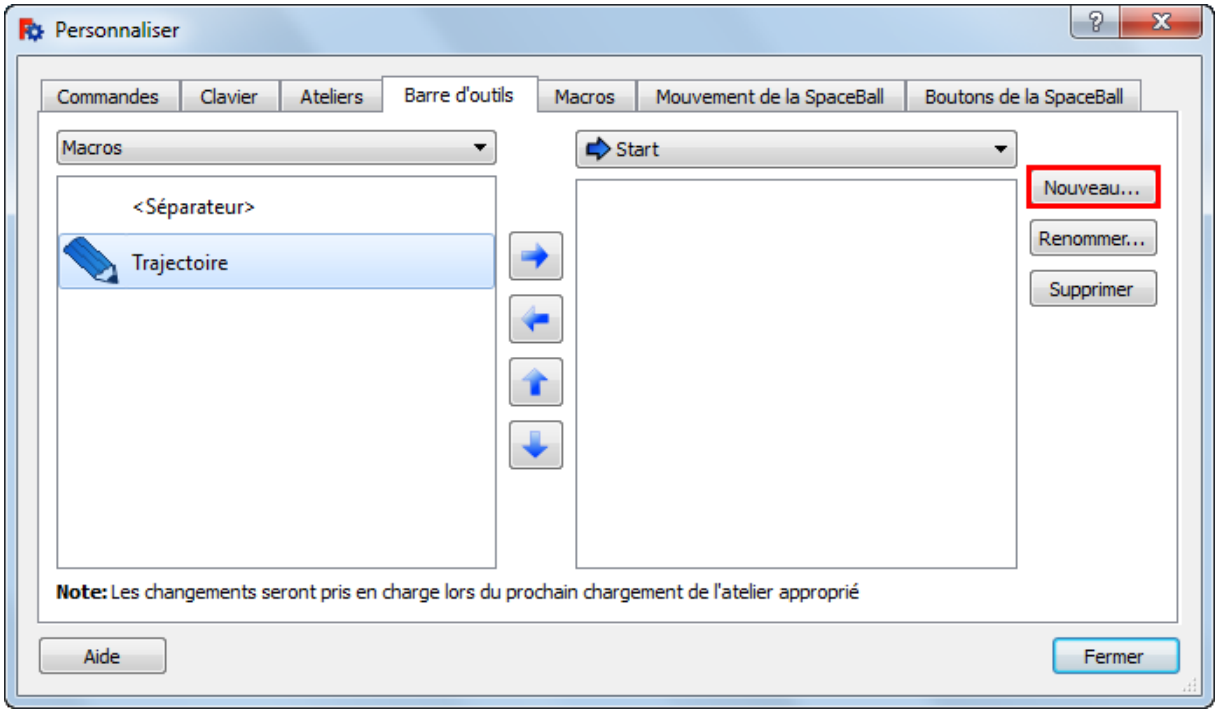


Figure III.10.c

- dans la fenêtre **Nouvelle barre d'outils** entrez le nom de la nouvelle barre d'outils et validez avec le bouton OK

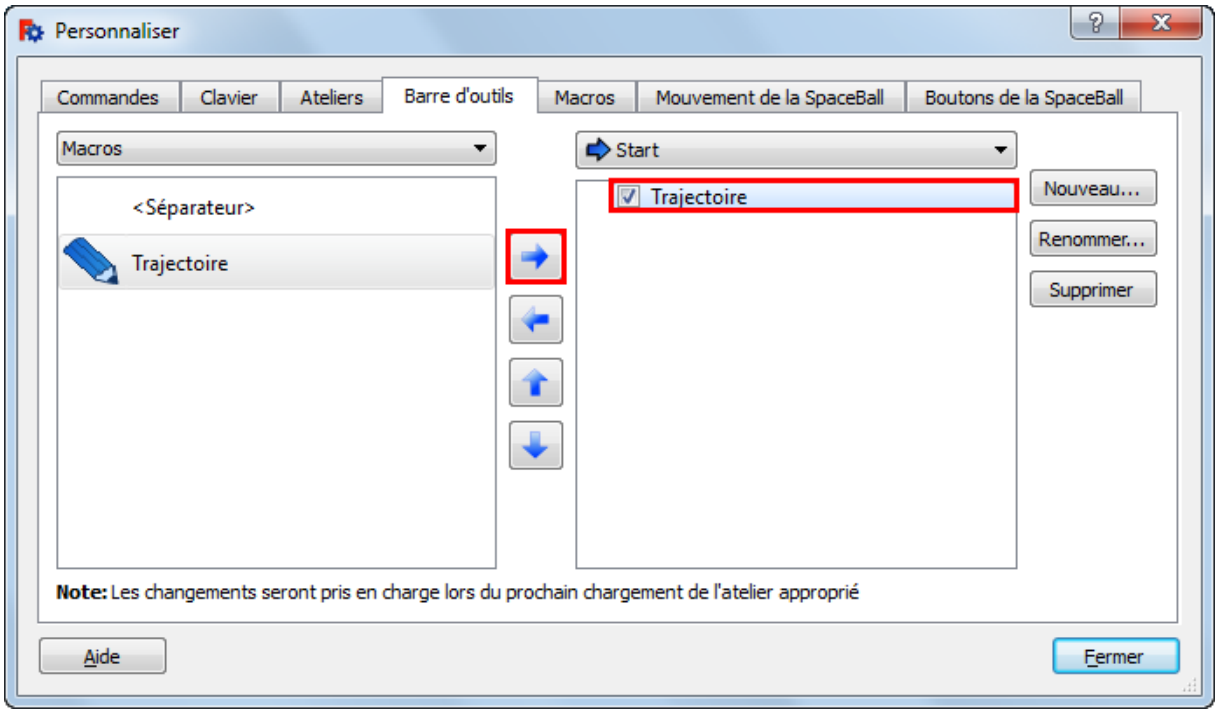


Figure III.10.d

- vous venez maintenant de créer une barre d'outils appelée "Trajectoire" avec un bouton qui s'appelle aussi "Trajectoire".

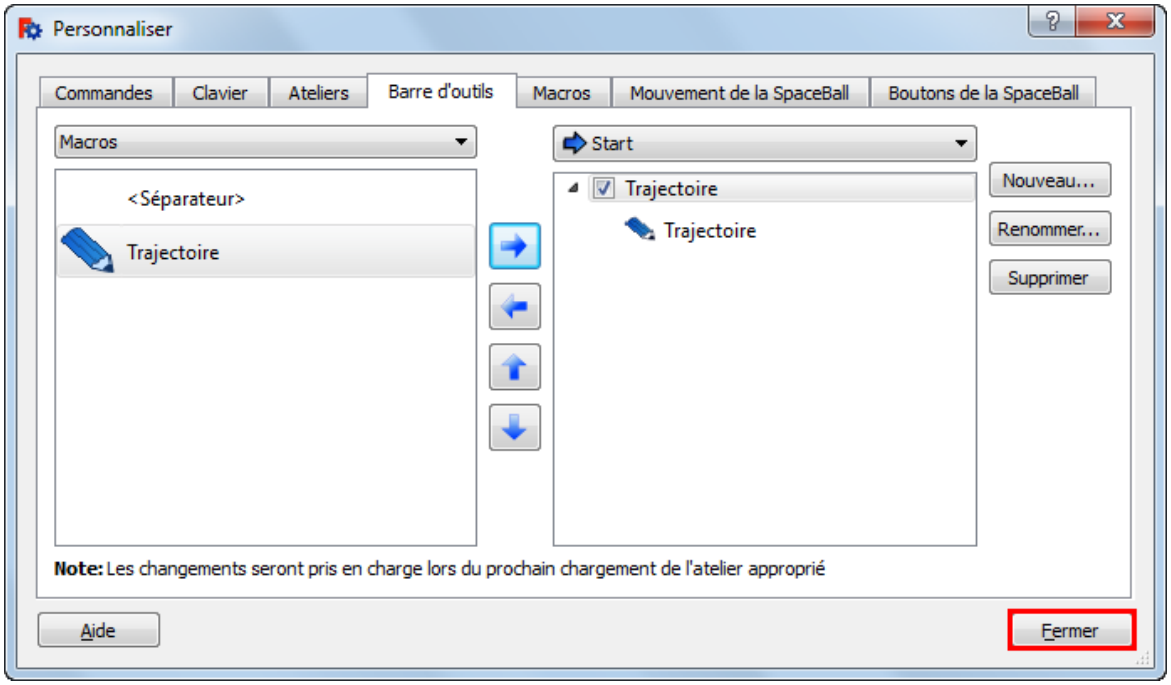


Figure III.11. La Barre d'outils est crée

- votre 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.

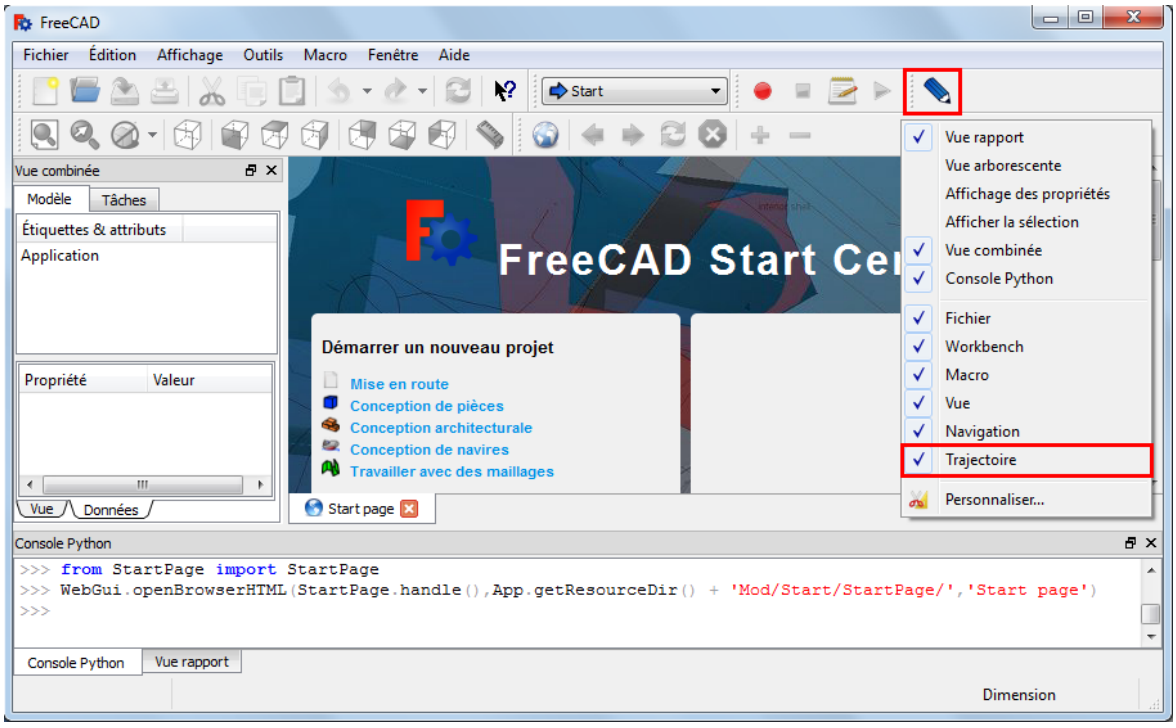


Figure III.12. La barre d'outils et le bouton sont affichés

III.5. Exemples de superposition de trajectoires et leur modèle 3D avec notre programme

On va tester notre programme pour générer la trajectoire sur des exemples des pièces de formes déférentes.

III.5.1. Pièce prismatique

- ouvrir le fichier de la pièce polygone (extension FCSTD).

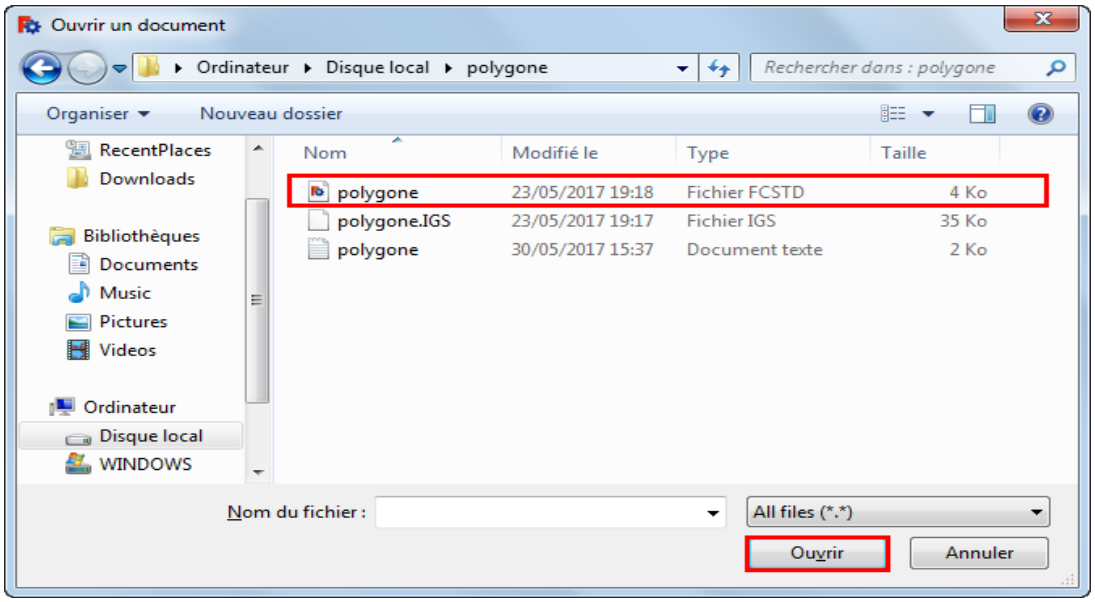


Figure III.13. Dossier de la pièce prismatique

- Cliquer sur le bouton de la macro.

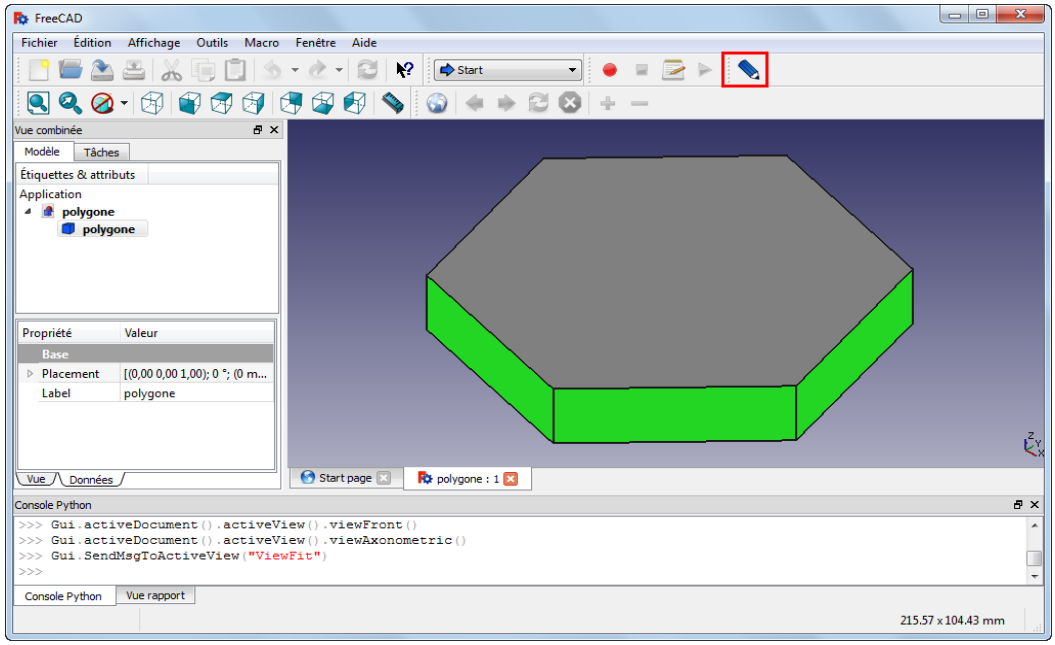


Figure III.14. Pièce prismatique ouverte dans FreeCAD

- La trajectoire est générée comme suit.

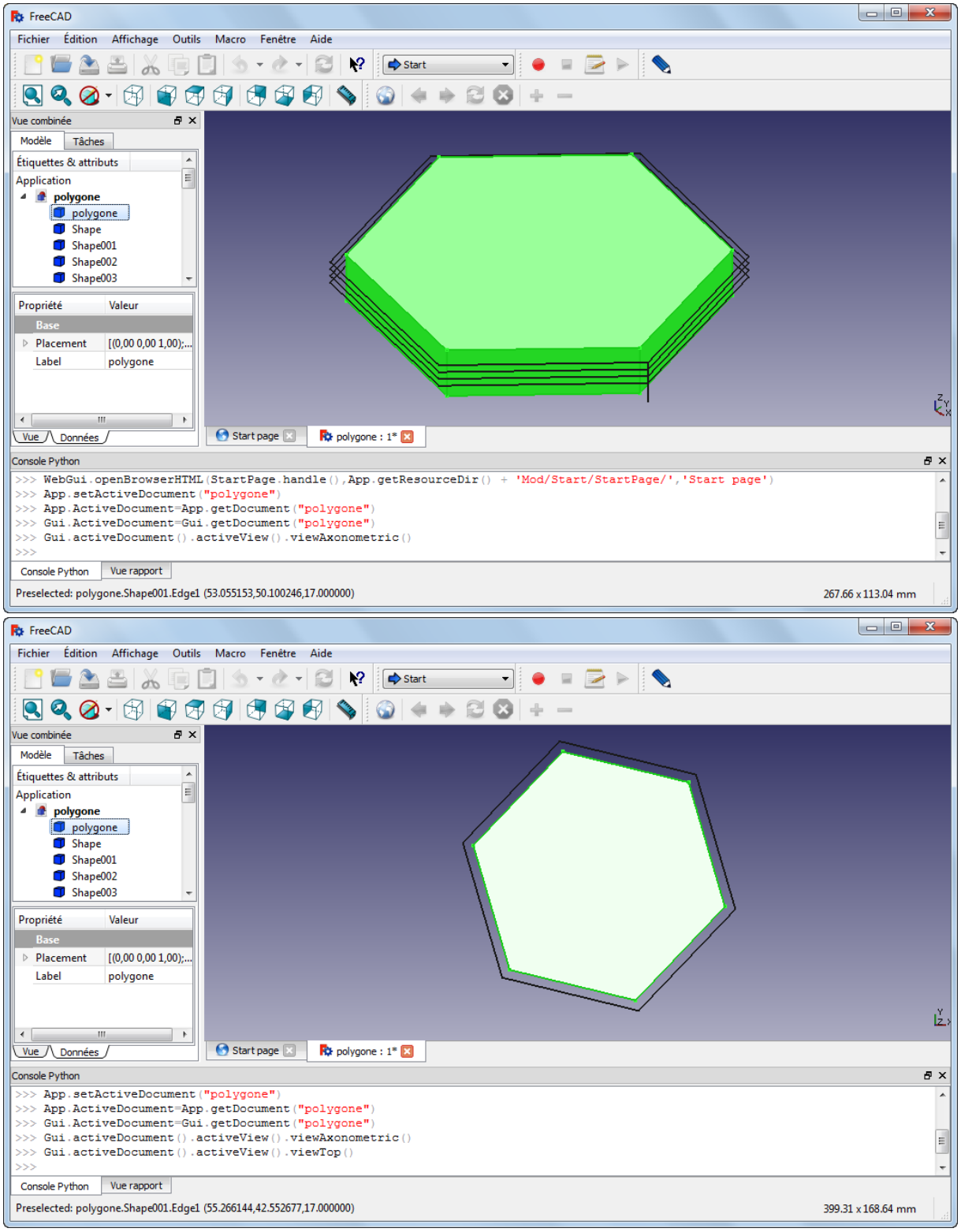


Figure III.15. Trajectoire de la pièce prismatique

Remarque : le G-code de cette pièce génère une trajectoire conforme à la surface usinée en mode contournage (pas de collision avec la pièce).

III.5.2. Pièce de révolution

- ouvrir le fichier de la pièce cylindre (extension FCSTD).

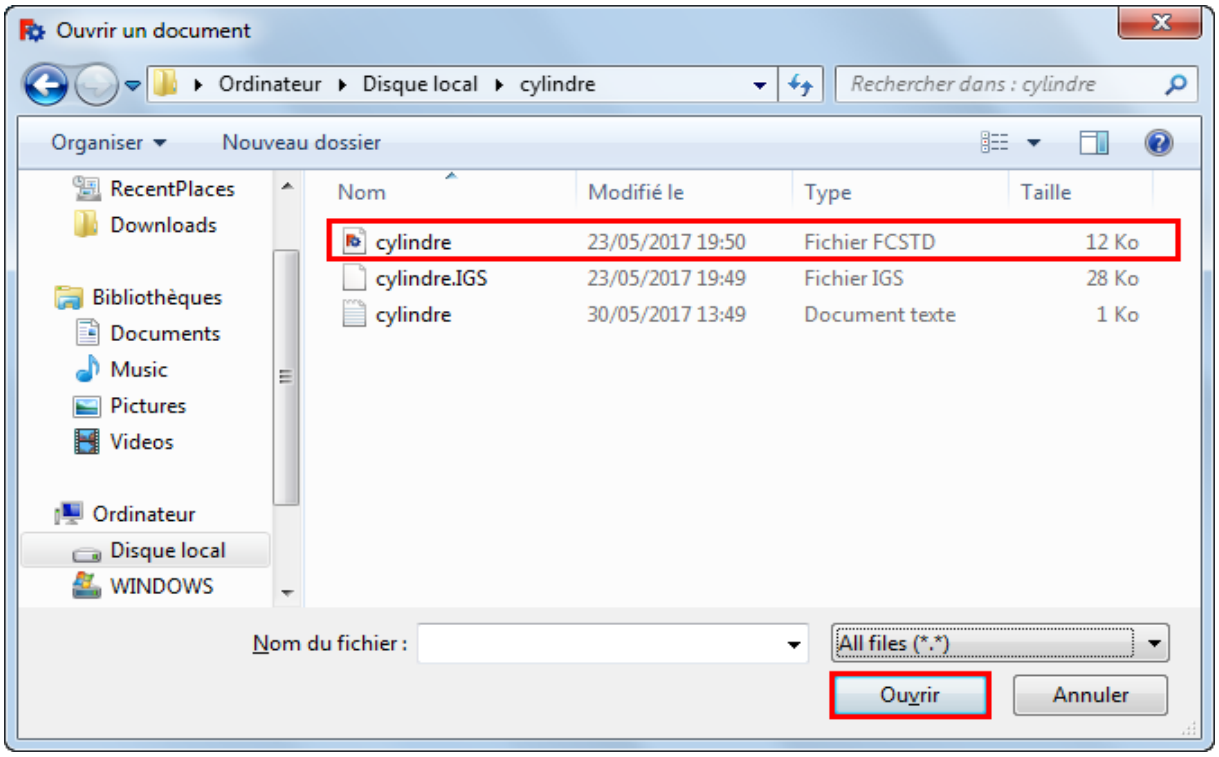


Figure III.16. Dossier de la pièce de révolution

- Cliquer sur le bouton.

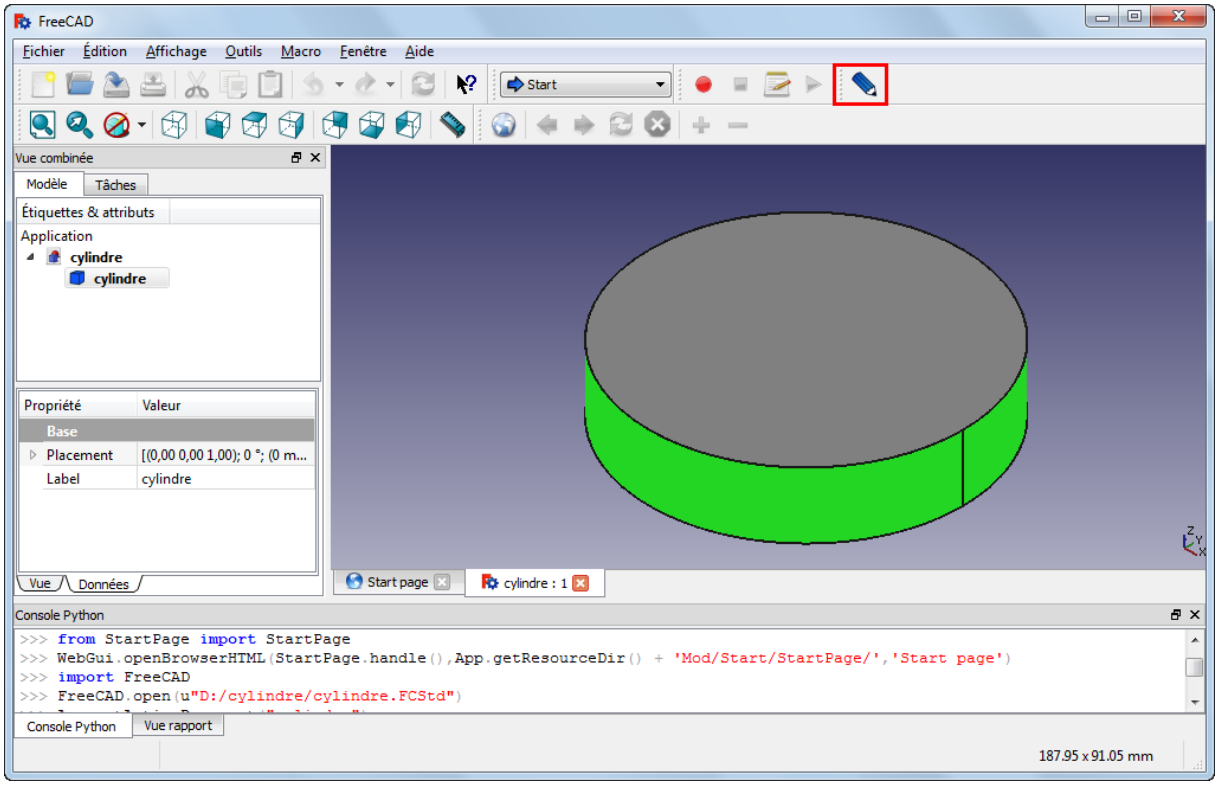


Figure III.17. Pièce de révolution ouverte dans FreeCAD

- La trajectoire est générée comme suit.

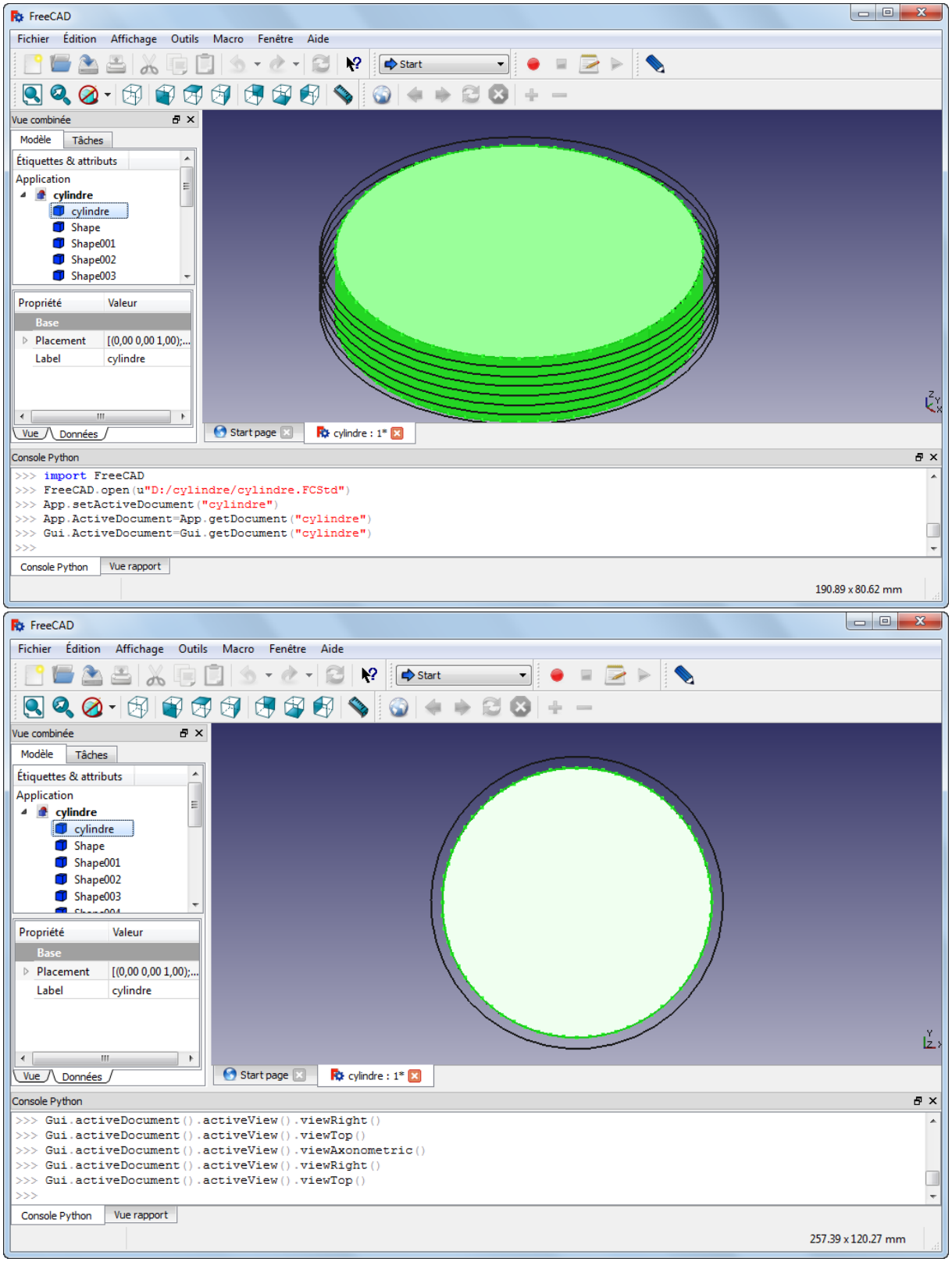


Figure III.18. Trajectoire de la pièce de révolution

Remarque : le G-code de cette pièce génère une trajectoire conforme à la surface usinée toujours en contournage (pas de collision avec la pièce).

III.5.3. Pièce de forme mixte n°1

- ouvrir le fichier de la pièce mixte (extension FCSTD).

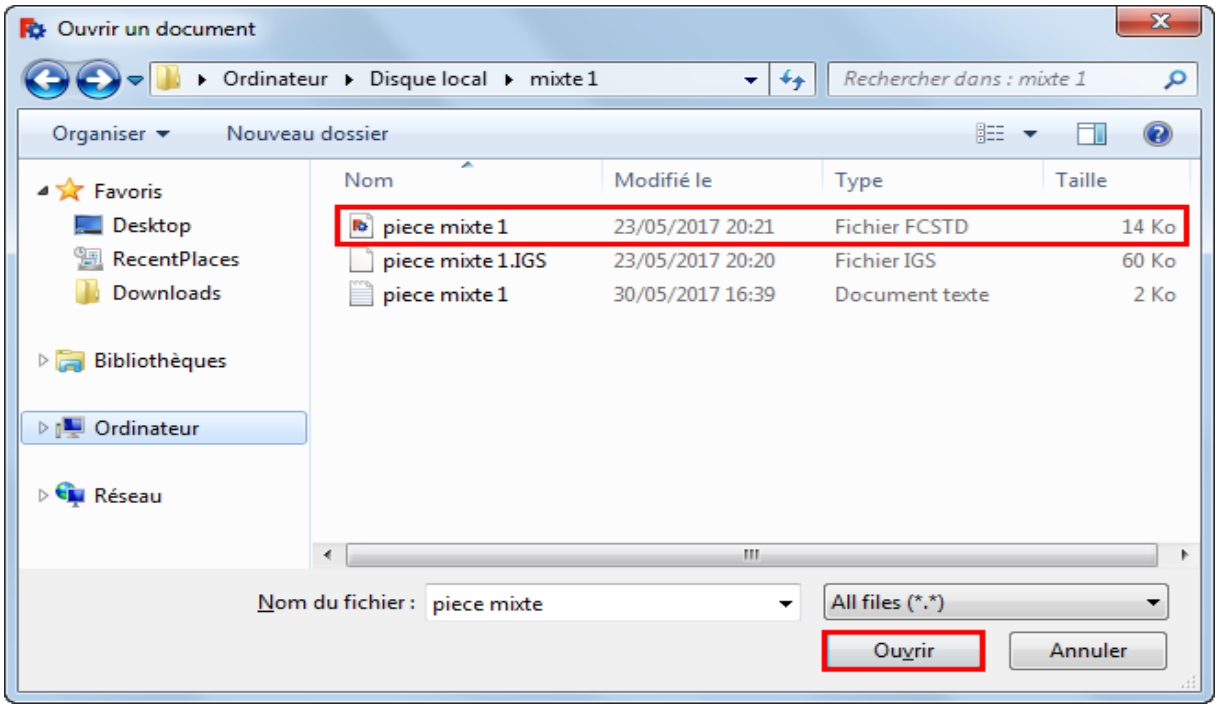


Figure III.19. Dossier de la pièce de forme mixte n°1

- Cliquer sur le bouton.

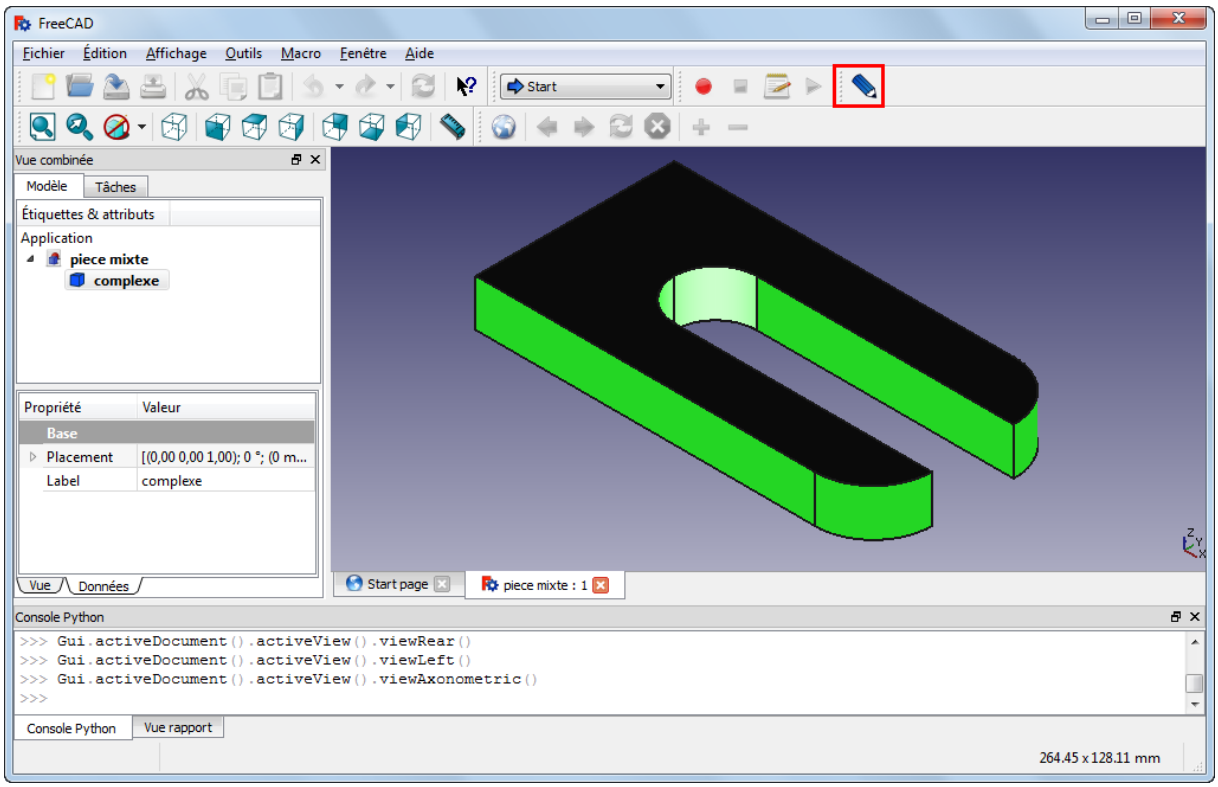


Figure III.20. Pièce mixte 1 ouverte dans FreeCAD

- La trajectoire est générée comme suit.

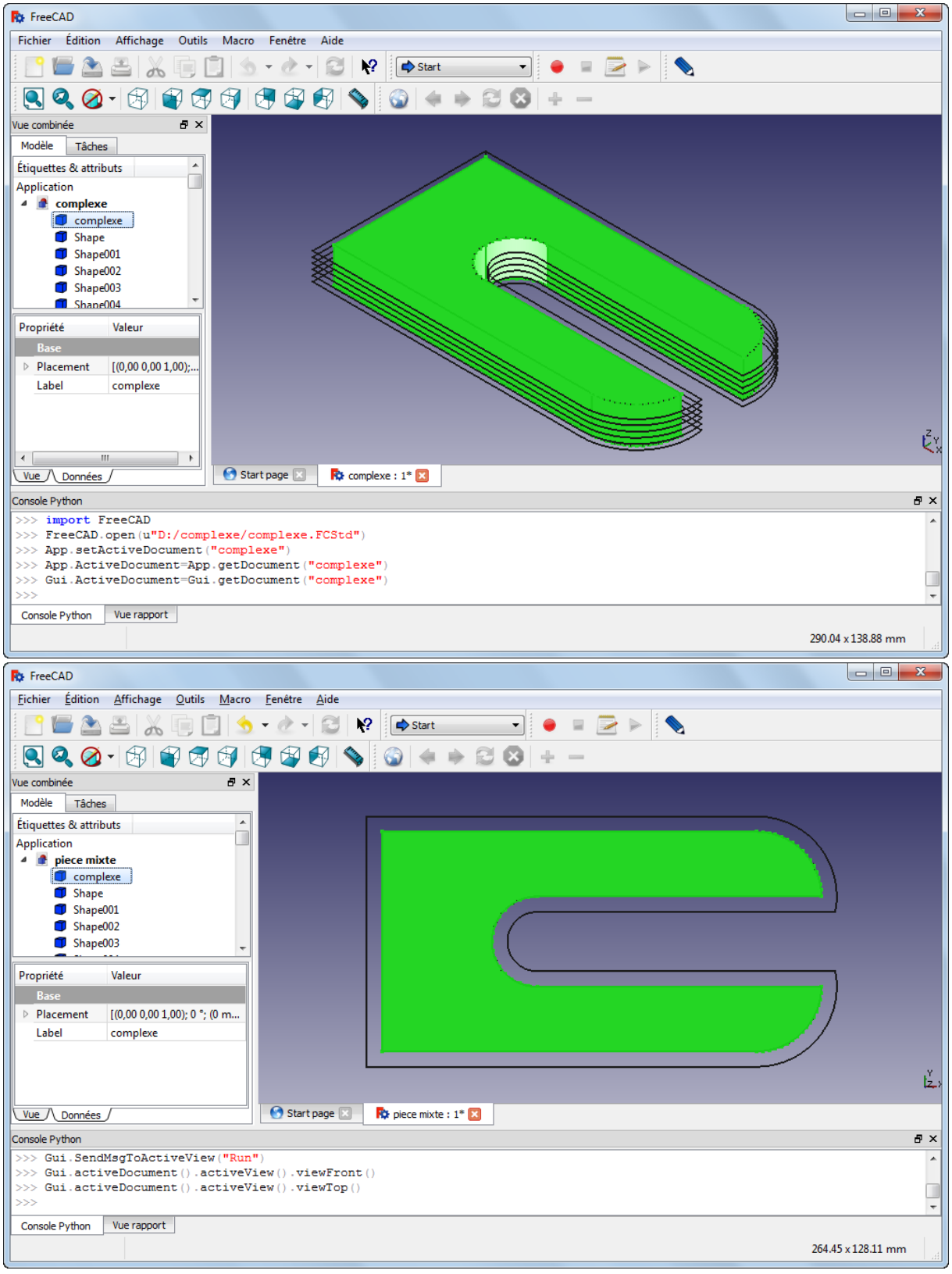


Figure III.21. Trajectoire de la pièce mixte 1

Remarque : le G-code de cette pièce génère une trajectoire conforme à la surface usinée (pas de collision avec la pièce).

III.5.3. Pièce de forme mixte n°2

- ouvrir le fichier de la pièce mixte (extension FCSTD).

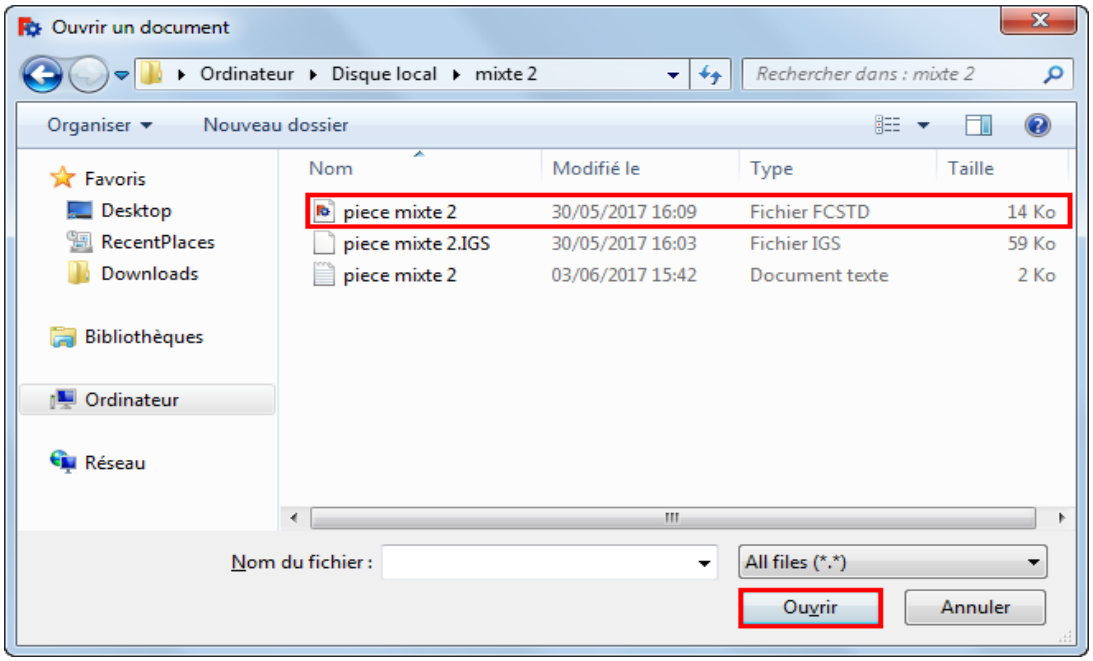


Figure III.22. Dossier de la pièce de forme mixte n°2

- Cliquer sur le bouton.

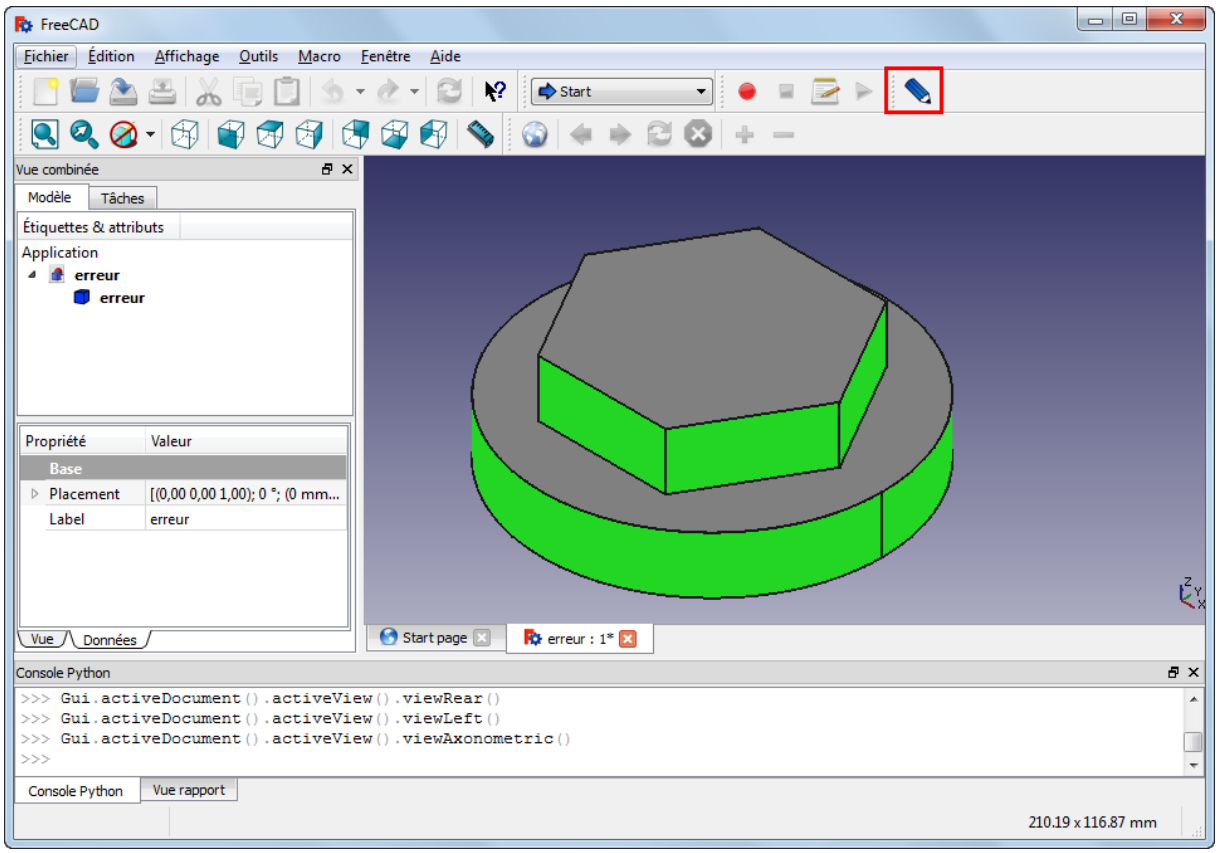


Figure III.23. Pièce mixte 2 ouverte dans FreeCAD

- La trajectoire est générée comme suit.

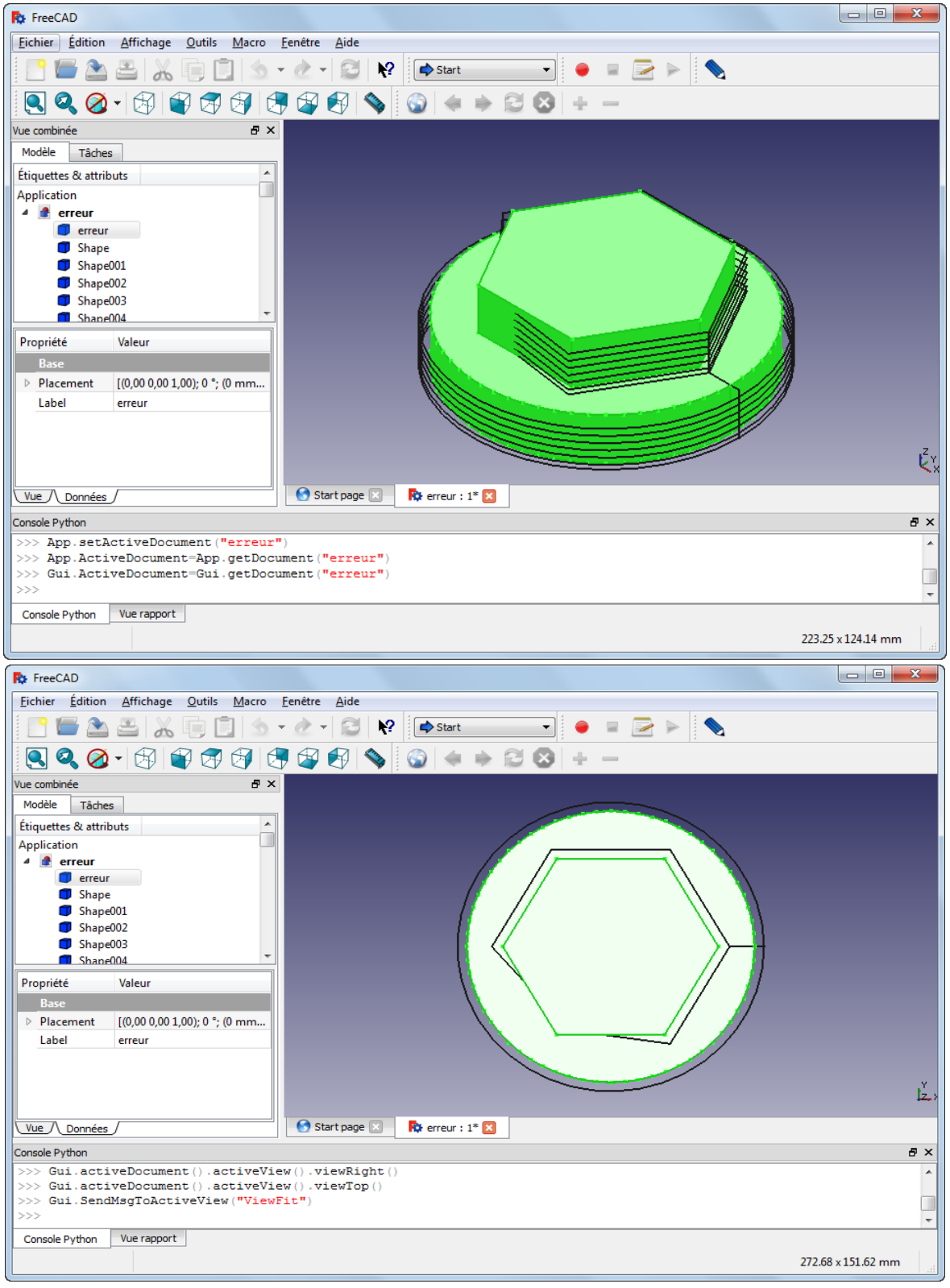


Figure III.24. Trajectoire de la pièce mixte 2

Remarque : le G-code de cette pièce génère une trajectoire qui fait une collision avec la pièce (il y'a une erreur dans le G-code).