

CHAPITRE III : Application : Génération de trajectoire de contournage

III.1. Présentation du logiciel SolidWorks

III.1.1. Historique

Créé en 1993 par l'éditeur américain éponyme, SolidWorks a été acheté le 24 juin 1997 par la société Dassault Systèmes.

Parmi les plus grandes organisations utilisant SolidWorks, on peut citer les entreprises mondiales : Franckie, équipement d'emballage MMC, AREVA, Axiome ... etc. des entreprises nationales comme : CIT, groupe SNVI, Poval ... etc. Ainsi que des établissements d'enseignement secondaire et universitaire.

III.1.2. Fonctionnement

SolidWorks est un modéleur 3D utilisant la conception paramétrique [6]. Il génère trois (3) types de fichiers relatifs à trois concepts de base : la pièce, l'assemblage et la mise en plan (Figure III.1). Ces fichiers sont en relation. Toute modification à quelque niveau que ce soit est répercutée vers tous les fichiers concernés.

Un dossier complet contenant l'ensemble des relatifs à un même système constitue une maquette numérique. De nombreux logiciels viennent compléter l'éditeur SolidWorks. Des utilitaires orientés métiers (tôlerie, bois, BTP...), mais aussi des applications de simulation mécanique ou d'image de synthèse travaillent à partir des éléments de la maquette virtuelle.

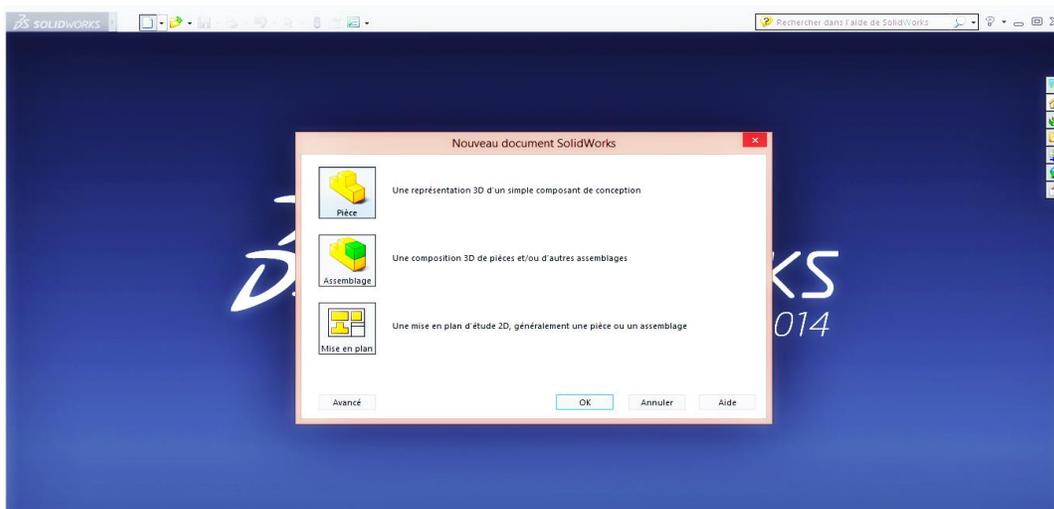


Figure III.1 : L'interface SolidWorks

III.1.3. Les extensions des fichiers

La simple ouverture d'un fichier dans une version ultérieure le rend inutilisable pour toute version antérieure.

Vu leur très faible interopérabilité et le fait que leur contenu soit sauvé sans que l'on utilise la commande de sauvegarde, ces fichiers Solidworks ne doivent pas être considérés comme des sauvegardes à long terme d'un contenu, mais comme une simple extension de la mémoire physique ayant la propriété de rémanence.

Chaque type de fichier possède une extension qui lui est propre. On retrouve :

- .sldprt, pour les fichiers pièce
- .sldasm, pour les fichiers assemblage
- .slddrw, pour les fichiers plan
- .sldprt, pour les fichiers de fond de plan

Certains formats proposés par le logiciel permettent d'envisager une sauvegarde à long terme.

III.2. L'interface de programmations des applications (API)

En informatique, une interface de programmation applicative (souvent désignée par le terme API pour (Application Programming Interface) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

Dans l'industrie contemporaine du logiciel, les applications informatiques se servent de nombreuses interfaces de programmation, la programmation se fait en réutilisant des briques de fonctionnalités fournies par des logiciels tiers. Cette construction par assemblage nécessite pour le programmeur de connaître la manière d'interagir avec les autres logiciels, qui dépend de leur interface de programmation. Le programmeur n'a pas besoin de connaître les détails de la logique interne du logiciel tiers, et celle-ci n'est généralement pas documentée par le fournisseur.

Des logiciels tels que les systèmes d'exploitation, les systèmes de gestion de base de données, les langages de programmation, ou les serveurs d'applications comportent une interface de programmation[8].

III.2.1. SolidWorks API

Une API, ou interface de programmation d'application, est une interface logicielle qui permet aux applications de communiquer entre eux. La création d'API englobe de nombreux outils de programmation qui aident à construire des routines et protocoles. La société SolidWorks offre un ensemble d'outils logiciels 3D – y compris de CAO SolidWorks, SolidWorks Simulation et SolidWorks Sustainability – de créer, de gérer et de publication des données. Les développeurs de logiciels à SolidWorks recommandent une liste d'outils pour développer des applications API compatibles avec le logiciel de l'entreprise de programmation.

III.2.2. Macros SolidWorks

Selon SolidWorks, enregistrement d'une macro SolidWorks sert de la manière la plus accessible pour commencer la programmation avec l'API SolidWorks. Les programmes de SolidWorks contiennent les deux Microsoft Visual Basic pour Applications et Visual Studio Tools for Applications intégré dans le logiciel. Le premier permet aux utilisateurs d'enregistrer et de modifier les macros(figureIII.2), enregistrés en tant que fichiers au format SWP, dans le logiciel SolidWorks tandis que ce dernier fait la même chose pour le code VB.NET et c# dans le logiciel, bien qu'il crée la DLL plutôt que les fichiers SWP.

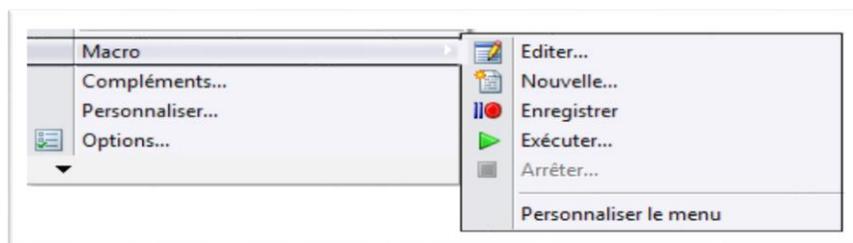


Figure III.2 : Macro SolidWorks

III.2.3. Création d'une Application API avec les Macros

Pour créer un API SolidWorks avec des macros SolidWorks, l'entreprise recommande de planifier soigneusement les actions d'interface utilisateur et en enregistrant ensuite ces actions. SolidWorks suggère alors passer à éditer la macro, suppression des lignes supplémentaires de code et déclarant et lie des variables. Exécution de la macro et le débogage aident à tester la stabilité. Une fois testée, VBA

de Microsoft et Microsoft VSTA se prêtent à la création d'interfaces utilisateur pour l'application. Des applications API SolidWorks autonomes prennent la forme de fichiers EXE, tandis que le composant logiciel enfichable des applications sont des fichiers DLL.

III.2.4. Langages de programmation

N'importe quel langage de programmation qui prend en charge l'architecture de Microsoft pour les applications compatibles Windows, le Component Object Model ou COM, s'adapte à la création d'applications API SolidWorks autonomes et composant logiciel enfichable. Ces langages de programmation incluent Visual c# .NET, Visual C++ 6.0, Visual Basic C++ et Visual Basic .NET.

III.2.5. Applications compatibles VBA Microsoft

Comprenant Microsoft VBA (Figure III.3) intégré dans le logiciel signifie que SolidWorks logiciel est activée en VBA. Logiciels compatibles VBA interagit avec d'autres applications qui sont également compatibles VBA, afin que les programmeurs puissent utiliser VBA pour créer une application SolidWorks qui s'attache à des objets actifs dans d'autres programmes. Exemples d'autres programmes compatibles VBA Microsoft Access, Microsoft Excel et Microsoft Visio. Lorsque les applications VBA attachent aux instances en cours d'exécution dans d'autres programmes compatibles VBA, ils peuvent récupérer des données qui peuvent à son tour être utilisées avec le logiciel SolidWorks [7].

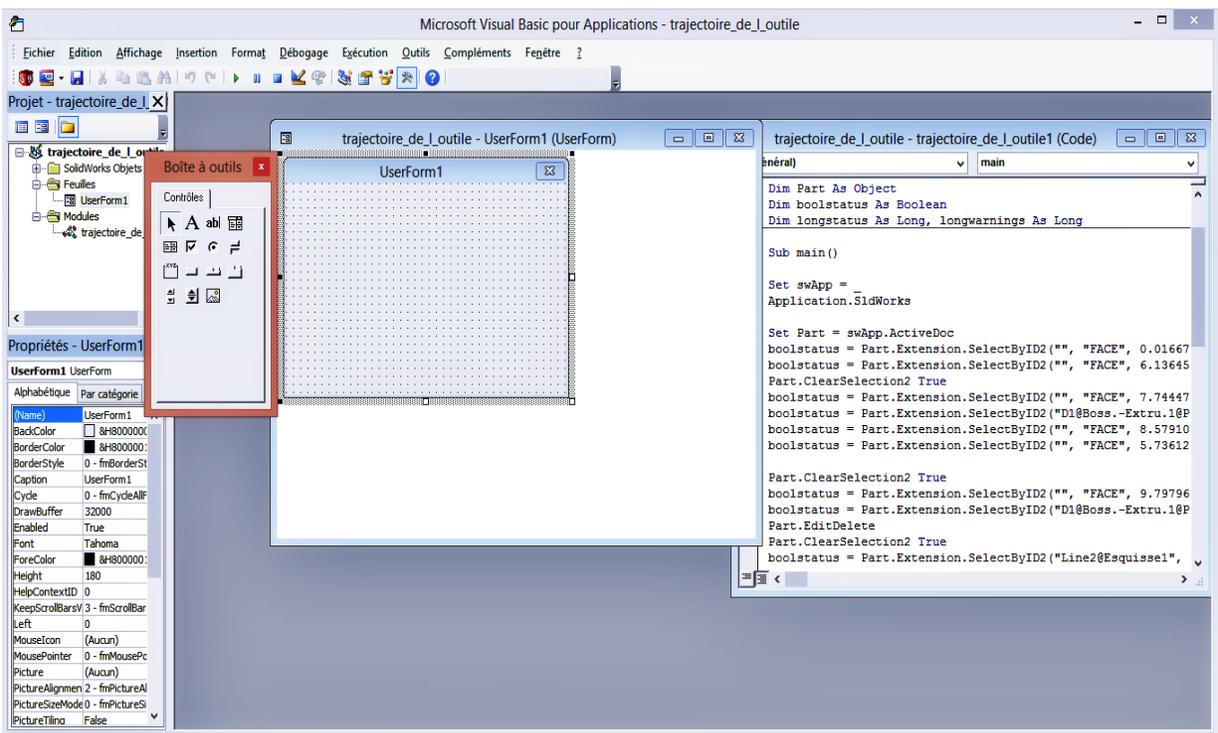


Figure III.3 : L'interface Microsoft Visual Basic pour Applications (VBA)

III.3. PRESENTATION DE TRAVAIL REALISE

Ce travail vous présente l'interface de programmation SolidWorks Application (API), SolidWorks VBA macro enregistreur et Microsoft Visual Basic pour Applications (VBA) Integrated Development Environment (IDE). Cela vous permet d'automatiser et de personnaliser le logiciel SolidWorks et d'intégrer votre logiciel avec le logiciel SolidWorks.

La façon la plus rapide et plus simple pour commencer à programmer avec l'API SolidWorks est d'enregistrer une macro. Ensuite, vous pouvez modifier la macro pour répondre aux besoins de votre travail. Notre travail sous forme de tutoriel ou didacticiel fournit des instructions sur l'enregistrement, l'édition et l'exécution d'une macro écrite en VBA dans le logiciel SolidWorks.

Créer un programme VBA API SolidWorks en enregistrant les opérations effectuées pour créer une trajectoire de l'outil (opération de contournage) sur une pièce. L'enregistreur de macros SolidWorks enregistre chaque opération effectuée lors de l'enregistrement.

III.3.1. Les opérations pour créer une trajectoire de l'outil (opération de contournage) :

On commence par ouvrir la pièce (figure III.4) pour faire les opérations effectuées pour créer une trajectoire de l'outil (opération de contournage) sur cette pièce (figure III.5).

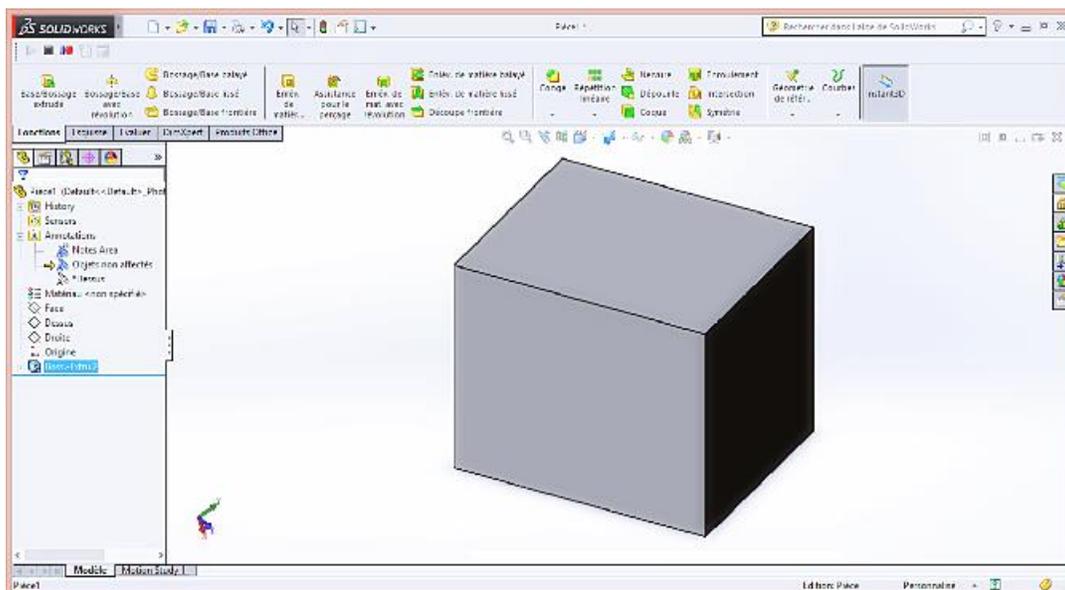


Figure III.4: Ouvrir une pièce dans SolidWorks

Cliquez sur **Enregistrer \ Pause Macro**  (barre d'outils Macro) pour commencer l'enregistrement de la macro.

Cliquez sur **Géométrie de référence**  (barre d'outils Fonctions) pour créer un plan, et choisissez **plan**.

Sélectionnez la face de la pièce comme une référence.

Régler la **distance**  10.00mm.

Régler la direction de plan, cochez sur  **inverser**.

Cliquez .

Sélectionner le plan que vous créez.

Cliquez sur **Esquisse**  (barre d'outils Esquisse) pour insérer un croquis.

Sélectionnez la face.

Cliquez sur **Décaler les entités**  (barre d'outils Esquisse).

Régler la valeur  10.00mm.

Cliquez .

Cliquez sur .

Créez un nouveau plan parallèle au plan 1, et entrez la **distance**  10.00mm.

Cliquez .

Cliquez sur .

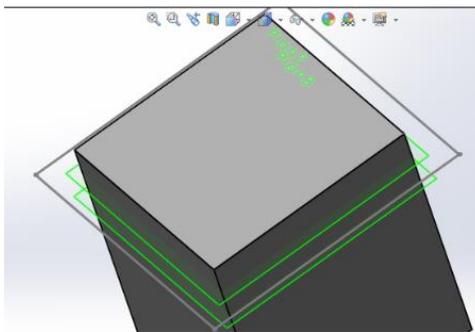


Figure III.5 : Les opérations pour créer une trajectoire de l'outil

Cliquez sur **Arrêter Macro**  (barre d'outils Macro) pour arrêter l'enregistrement de la macro.

Dans la boîte de dialogue Enregistrer sous, accédez au dossier où vous souhaitez enregistrer la macro.

Assurez-vous que Enregistrer sous le **type** est **SW VBA Macros (*.swp)**.

Tapez **trajectoire de l'outil** dans le **nom du fichier**.

Cliquez sur **Enregistrer**.

III.3.2. Modifier la macro pour répondre aux besoins de notre travail à faire

Cliquez sur **Editer la Macro**  (barre d'outils Macro).

Ouvrez **trajectoire de l'outil.Swp**.

Le Visual Basic Editor et la fenêtre ouverte Code. La fenêtre de code contient le code de l'API SolidWorks généré pour la macro (figure III.6).

Cliquez sur **Ouvrir**.

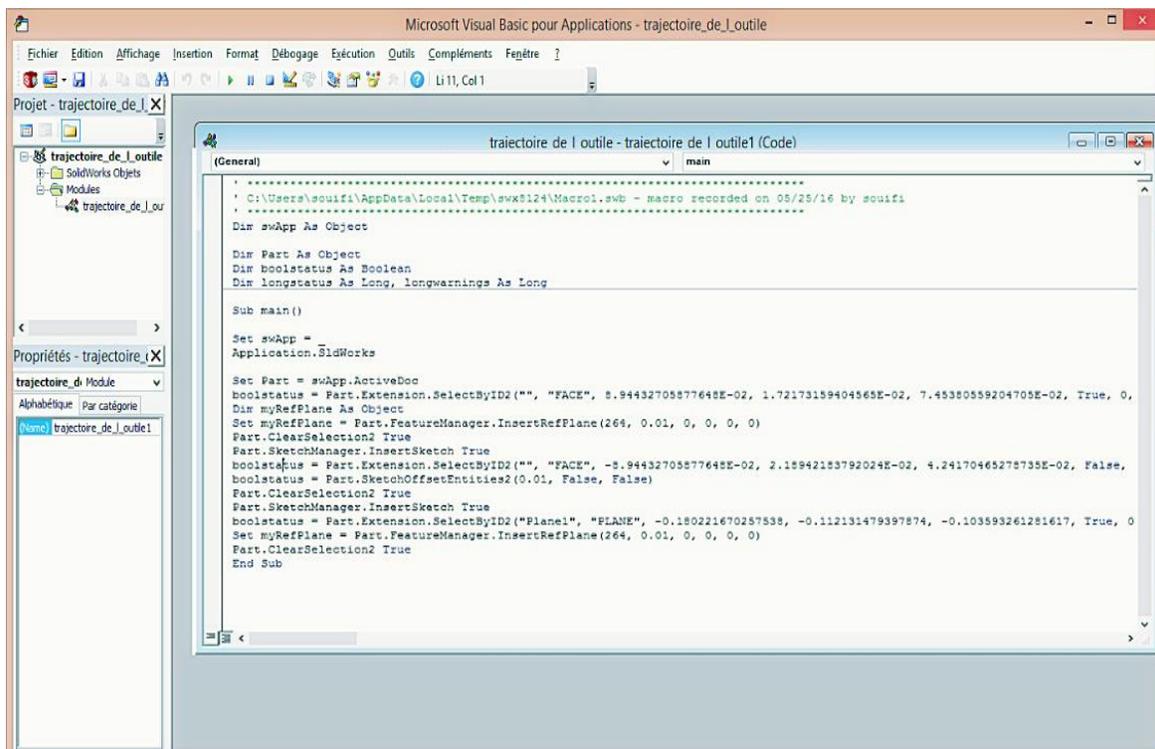


Figure III.6 : Microsoft Visual Basic pour Application (trajectoire de L'outil)

Pour éviter des problèmes comme des variables mal nommés et d'améliorer les performances de notre programme, nous devons explicitement déclarer et lier des variables tôt.

De déclarer explicitement toutes les variables, le type *Option Explicit* au sommet du programme, avant

Dim SwAppAs Object.

Option Explicit

Dim SwAppAs Object

Certaines variables sont déclarées automatiquement dans une macro SolidWorks. Nous pouvons supprimer toutes les variables ne sont pas utilisées dans le programme.

Supprimez ces déclarations de variables.

Dim longstatusAs Long, longwarningsAs Long

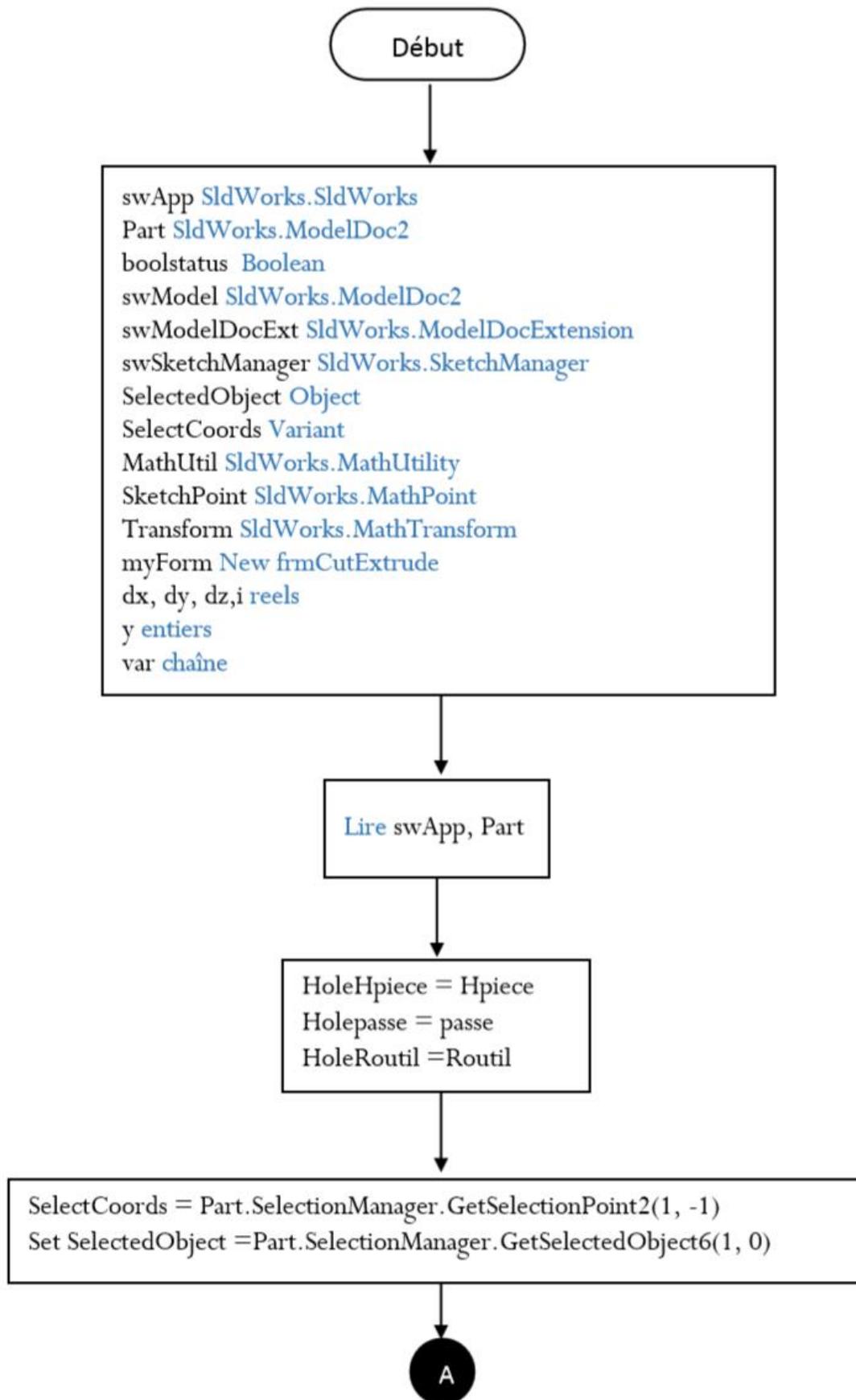
Modifiez les variables *SwAppet Part* pour les rendre tôt liés.

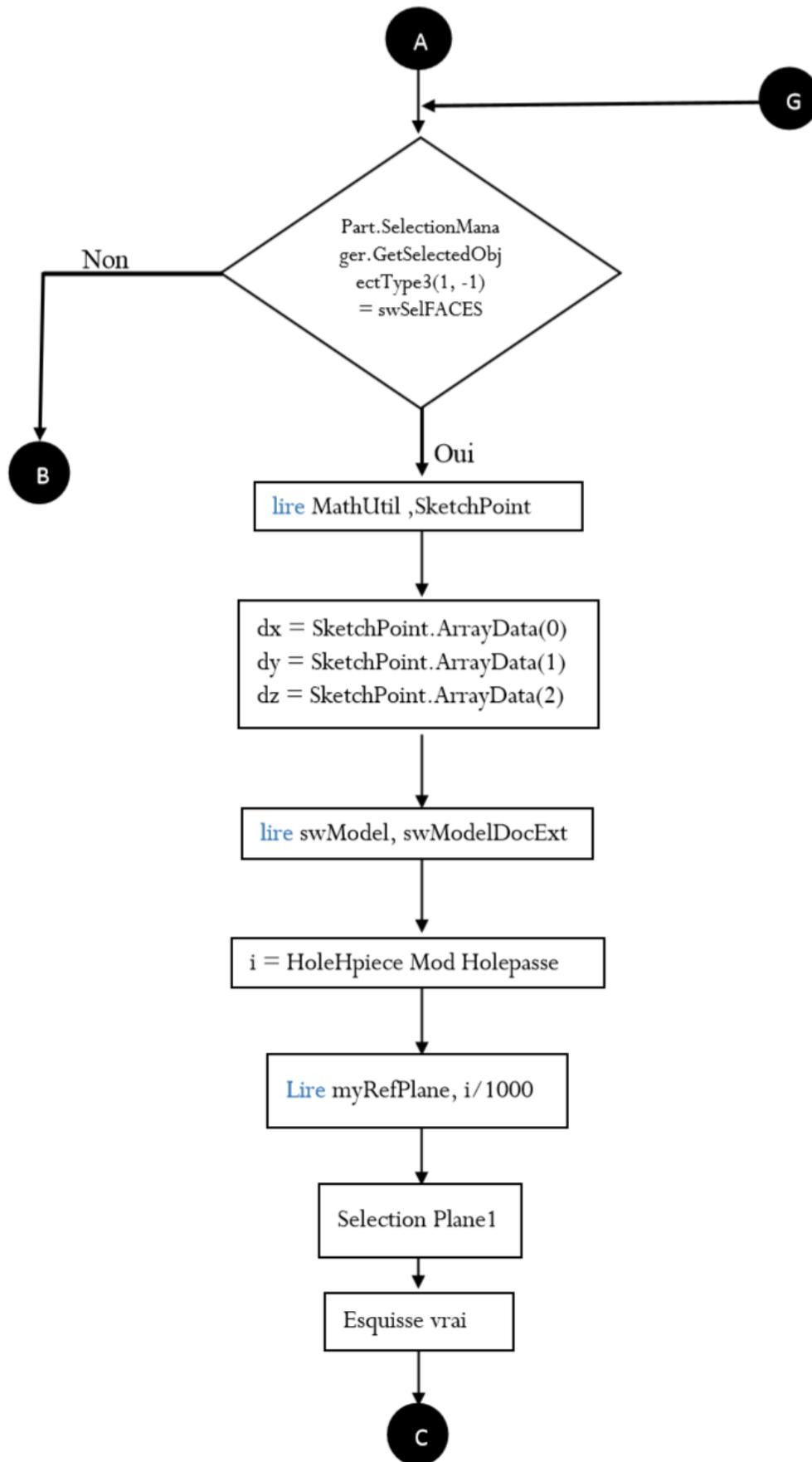
Dim swApp As SldWorks.SldWorks

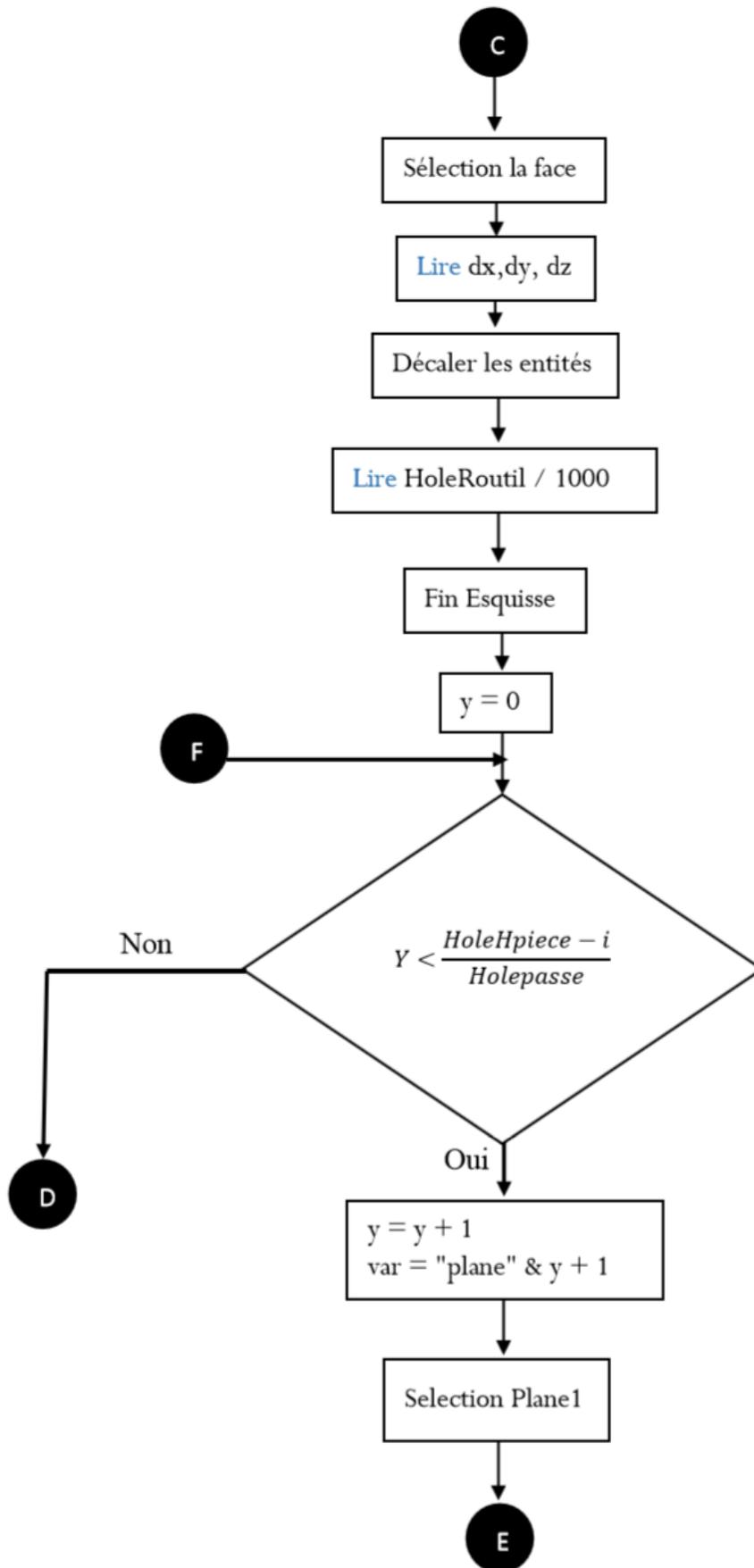
Dim Part As SldWorks.ModelDoc2

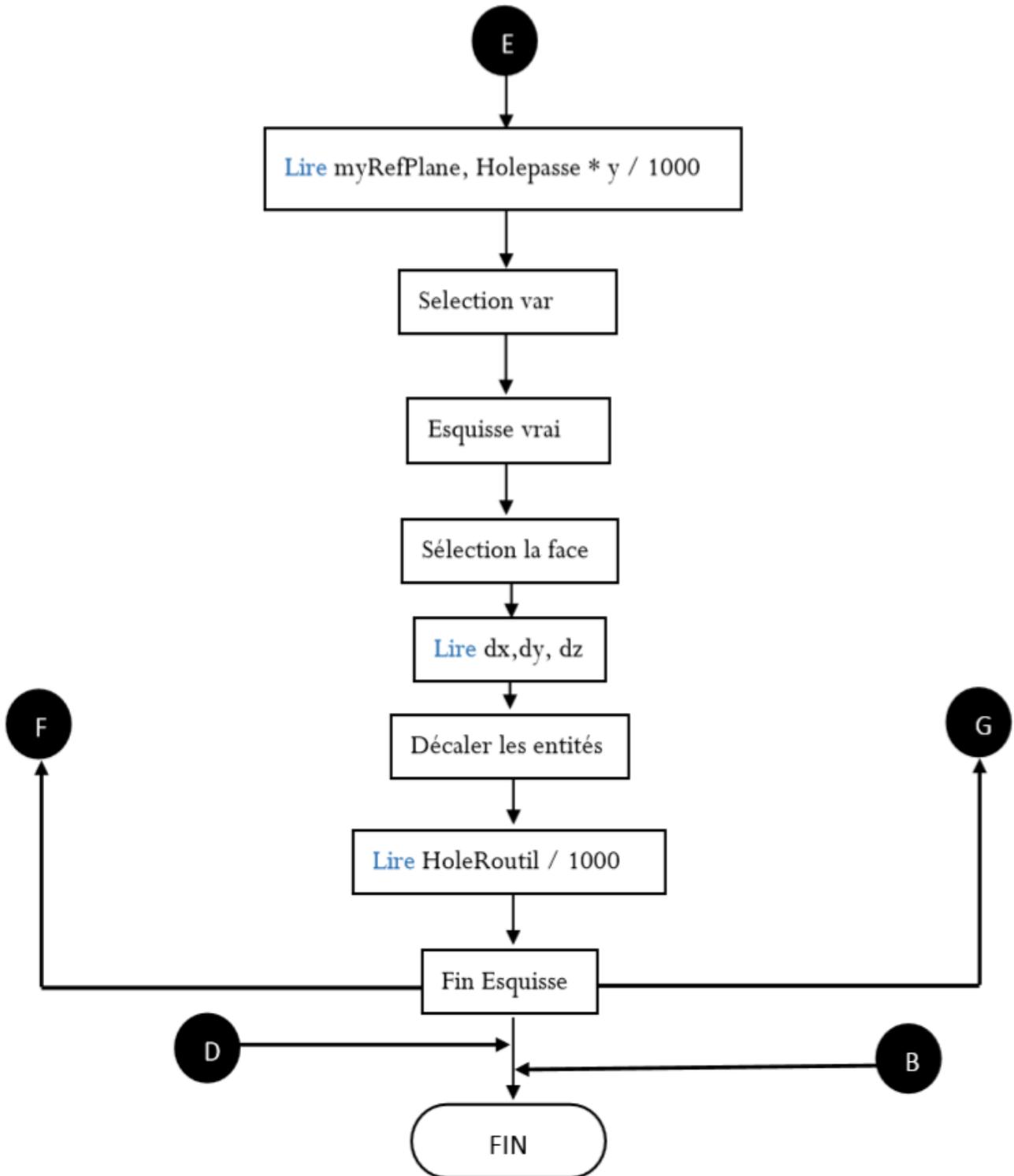
Modifier la macro dans la fenêtre de code de sorte que nous pouvons créer trajectoire d'outil (opération contournage) sur un visage présélectionné dans une pièce quelconque. Pour rendre le programme plus générique, d'éliminer le projet.

Voir organigramme pour obtenir des instructions sur la façon de modifier notre macro.









III.3.3. Réalisation de l'interface utilisateur :

Cette fenêtre va permettre à l'utilisateur de créer l'interface de sa macro. Une boîte à outils permet d'ajouter différents contrôles à cette fenêtre (figure III.7).

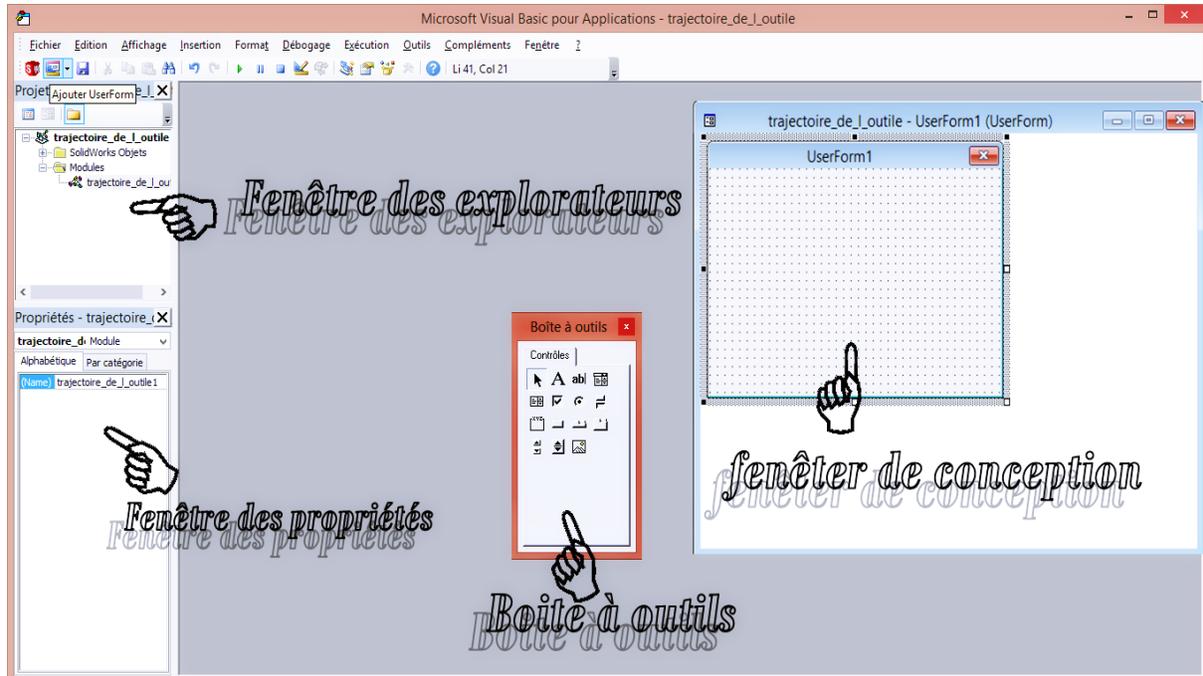


Figure III.7 : Réalisation de l'interface utilisateur

Cliquez sur **Ajouter UserForm** pour ajouter un formulaire vierge appelé **UserForm1** au projet et pour afficher la boîte à outils.

Cliquez sur **Affichage > Fenêtre des Propriétés** pour afficher les propriétés associées à **UserForm1**.

Pour modifier le nom du formulaire, cliquez sur **UserForm1** dans l'**Explorateur de projet**.

Tapez **frmTrajectoireOutil** dans le **(Nom)** rangée dans la **fenêtre des Propriétés**.

Le nom en haut de la fenêtre Propriétés et le nom du module de formulaire dans le changement Explorateur de projet pour **frmTrajectoireOutil**. Toutefois, le nom indiqué sur le formulaire est la propriété **Caption** de la forme, que vous n'avez pas changée, il reste donc UserForm1.

Ajouter trois zones de texte et un bouton de commande, appelés contrôles, à la forme en utilisant la boîte à outils (figure III 8).

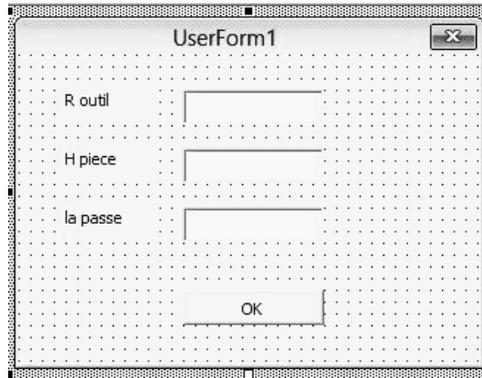


Figure III 8 : Composition de l'interface avec des zones de texte et le bouton de commande

Modifiez les noms des zones de texte et le bouton de commande pour des noms plus significatifs.

Double-cliquez sur le bouton OK sur le formulaire pour ouvrir la fenêtre de code contenant une procédure d'événement pour ce contrôle.

La forme finale de de l'interface utilisateur (Figure III 9).



Figure III 9 : L'interface utilisateur

Fermez La fenêtre de code sans l'enregistrer.

Si vous voulez ajouter un icône de raccourci dans une barre d'outils SolidWorks (Figure III 10), suivi les étapes : Options > Personnaliser > Commandes > Macro > Nouveau bouton de macro.

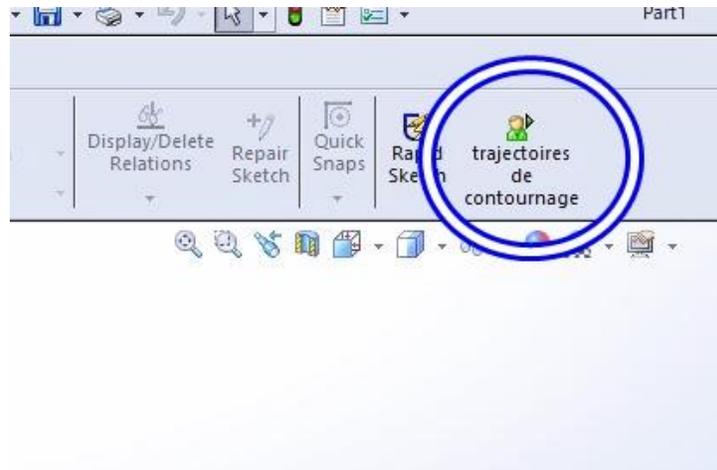


Figure III 10 : Intégration de l'icône de raccourci de notre programme

III.3.4. Exécution du programme

a) La première pièce (figure III.11) :

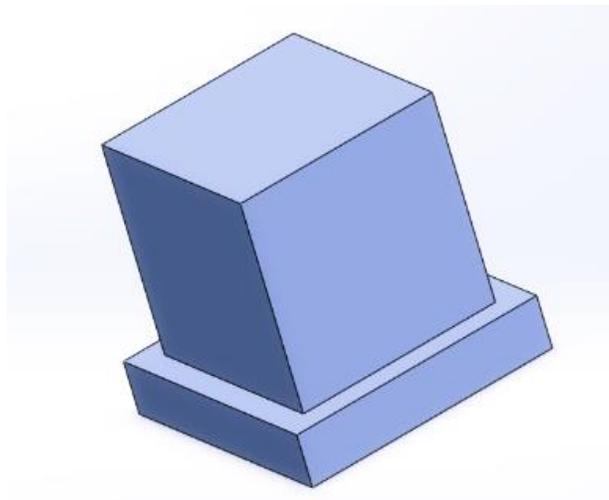


Figure III.11 : la pièce avant l'exécution de programme

Sélectionnez face de pièce et cliquez **trajectoire de contournage**

Entrez la valeur de **R outil**, **H pièce**, et **la passe** nous avons choisi dans cette exemple **R outil=5mm**, **H pièce=40mm**, et **la passe=3mm** (figure III.12).

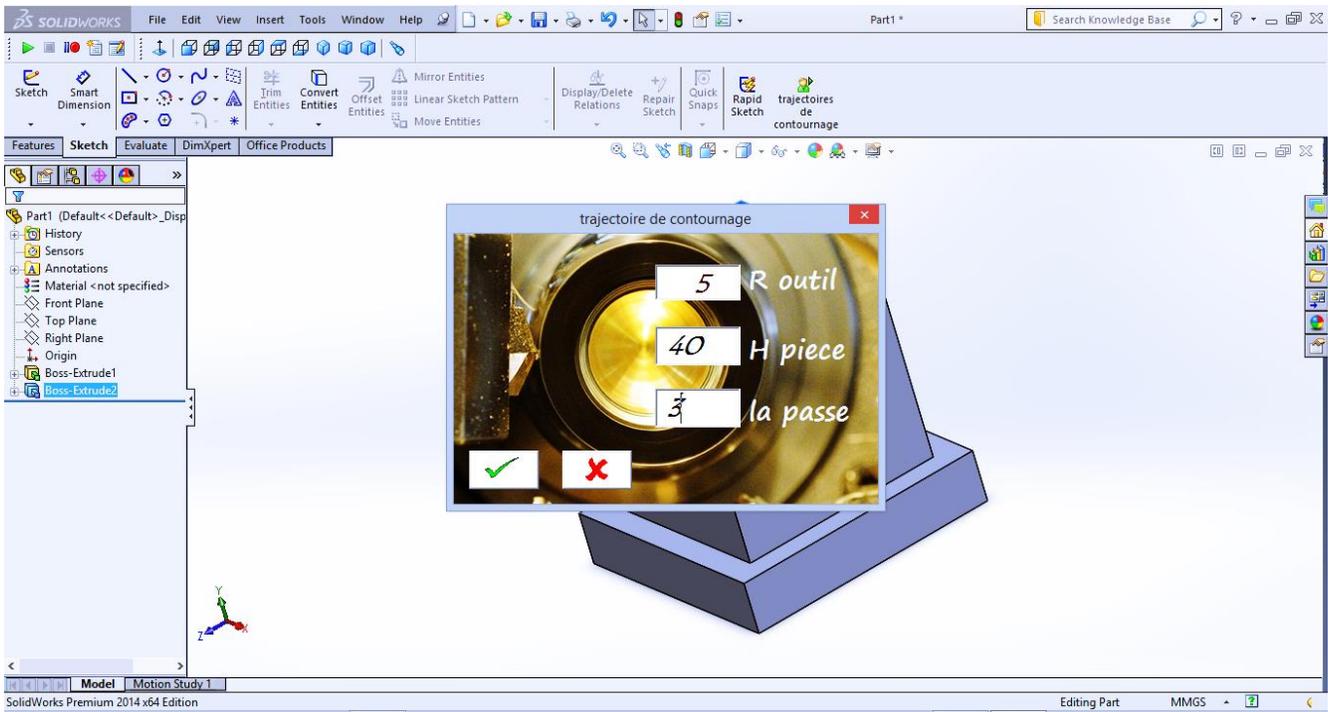


Figure III.12 : déclaration des valeurs de R outil, H pièce, et la passe

Cliquez OK et voir la résultat (figure III.13).

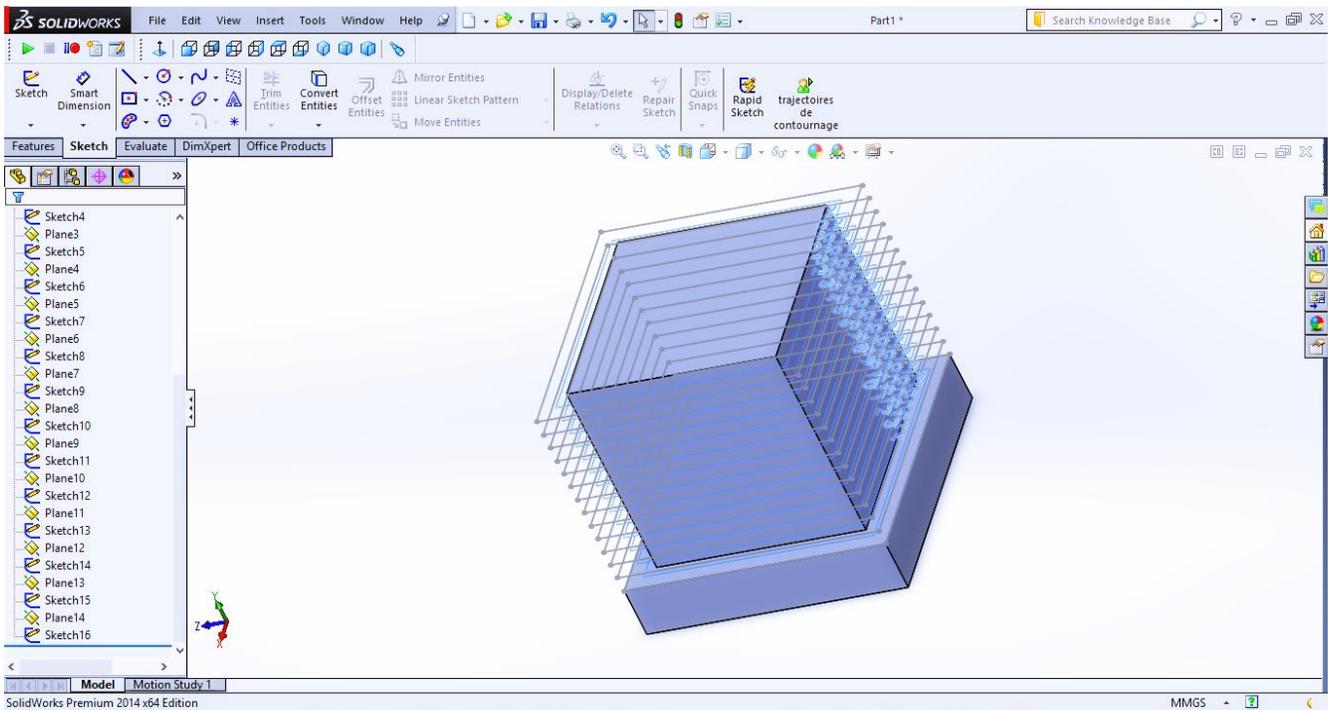


Figure III.13 : trajectoire de contournage sur la première pièce

2^{ème} exemple (figure III.14) :

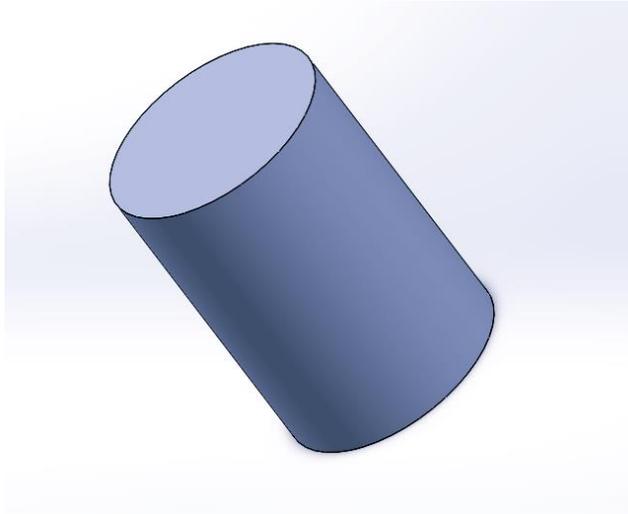


Figure III.14 : 2^{ème} exemple pièce cylindrique

Exécuté le programme et voir la résultat (figure III.15).

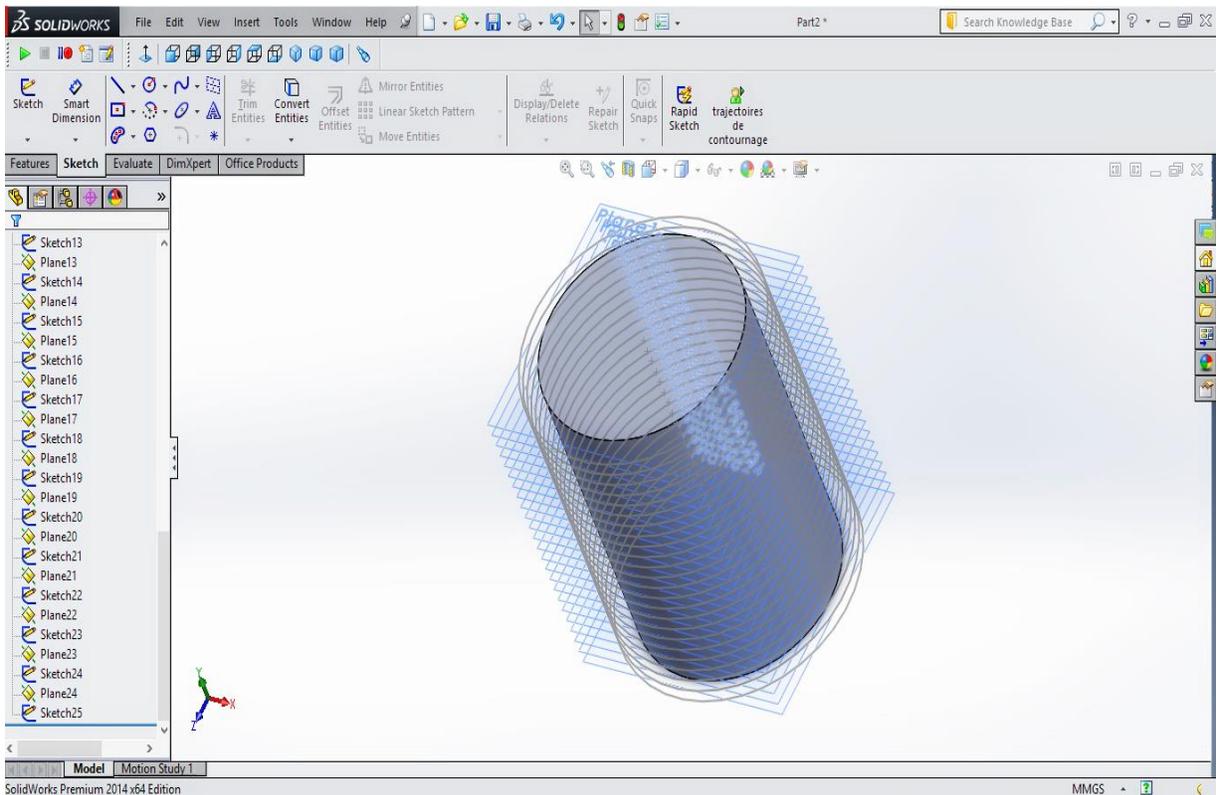


Figure III.15 : Trajectoire de contournage sur la 2^{ème} pièce

3^{ème} exemple (figure III.16) :

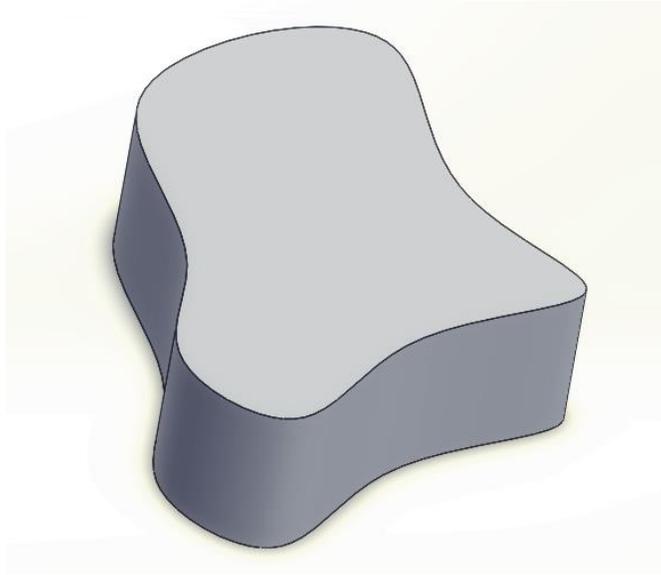


Figure III.16 : 3^{ème} exemple pièce

Exécuté le programme et voir la résultat (figure III.17).

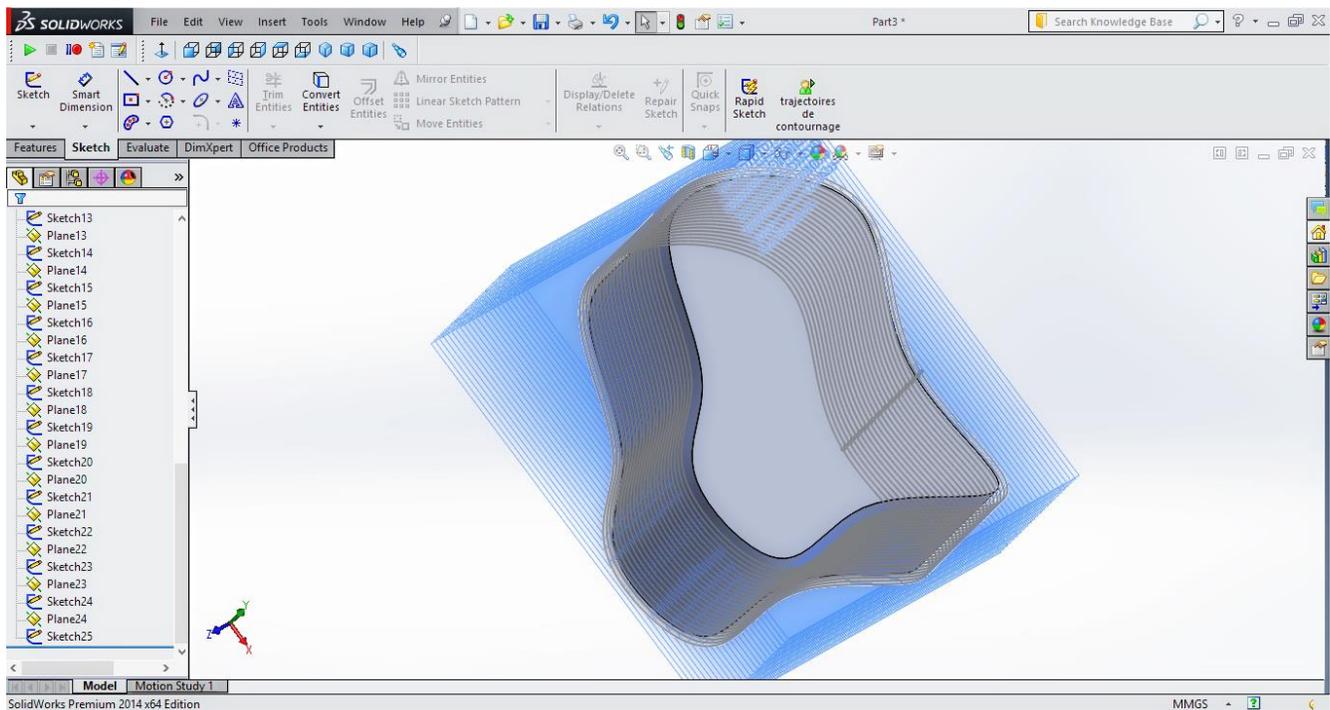


Figure III.17 : Trajectoire de contournage sur la 3^{ème} pièce