

TP2 : Interface de l'environnement MATLAB

1 Introduction :

L'objectif de ce court document est de vous initier au logiciel MATLAB de la compagnie Mathworks et à la programmation dans cet environnement. L'idée est de vous exposer les bases de cet outil de travail et de vous habilitier à résoudre des problèmes de l'ingénierie par tout. Il intègre à la fois des solutions de calcul, de visualisation et un environnement de développement. Son nom provient de MATrix LABoratory, l'objectif initial était de fournir un accès simplifié aux bibliothèques de fonction des projets LINPACK et EISPACK (dédiées au calcul matriciel et à l'algèbre linéaire).¹

2 MATLAB comme calculette :

Comme tout langage de programmation, MATLAB dispose de fonction de calculs mathématique.

Nous en voyons ici quelques exemples d'utilisation.

Si l'on regarde ce que donne l'exemple d'addition :

```
» 3+2
```

```
ans
```

```
5
```

On peut remarquer que lorsqu'aucune variable d'affectation n'est spécifiée pour stocker le résultat de l'opération, MATLAB stocke le résultat dans une variable appelée ans (diminutif pour anser).

Pour éviter cela, on peut spécifier le nom d'une variable pour stocker le résultat. On pourra alors réutiliser le résultat de l'opération plus tard. Par exemple :

```
» x=3+2
```

```
x =
```

```
5
```

3 Priorité des opérateurs :

Les opérations sont évaluées en donnant la priorité aux opérations selon l'ordre suivant :

¹Présence aux T.P. : La présence aux T.P. est obligatoire. Toute absence devra être valablement justifiée et donnera lieu à un rattrapage.

1. ()
2. ^
3. * /
4. +-

Étape 1 Analyser le problème.

Étape 2 Faire le profil du programme.

Étape 3 Mettre en œuvre le programme.

- a. Coder le programme.
- b. Tester et déboguer le programme.
- c. Documenter le programme.

4 Analyse du problème :

Dans cette étape, on doit:

- a. Déterminer les données qui vont être produites (sorties) comme solution du problème. Dans cette sous-étape, on doit définir les variables qui sont exigées pour représenter la sortie.
- b. Déterminer les données (entrées) nécessaire pour produire les sorties. Dans cette sous- étape, on doit définir les variables exigées pour représenter les entrées.
- c. Développer un algorithme pour obtenir les sorties des entrées en un nombre fini étapes. Un algorithme est défini comme une séquence organisée d'opérations pour résoudre un problème en un nombre fini d'étapes. Dans le cas des problèmes numériques, l'algorithme est constitué d'une série de calculs nécessaire pour obtenir les sorties. Mais, dans le cas des problèmes non numériques, les algorithmes peuvent inclure des opérations variées de manipulations de texte et graphique avec les opérations numériques.

Le but principal de cette étape est d'être sûr que le problème soit bien compris. N'importe quelle erreur commise dans l'analyse peut être transférée au programme.

5 Profil du programme :

Dans cette étape, on fait le profil du programme utilisant des phrases symboliques. Chaque phrase correspond à une simple opération du programme. Pour un simple programme, il est possible de produire le code symbolique en listant les différentes tâches que le programme doit exécuter, dans l'ordre avec lequel elles seront exécutées. Mais pour les programmes complexes, il est nécessaire d'organiser la procédure du profil. Pour cela, on utilisera la méthode d'organisation appelée la conception de haut en bas ou descendante.

Pour concevoir un programme utilisant l'approche descendante, on divise le programme en un nombre de tâches, la liste des tâches est un profil du programme appelé module principal. Dans le module principal, la tâche est seulement identifiée par son nom. Il n'y a pas d'indication de comment la tâche est actuellement accomplie. Ce niveau de détail est laissé pour l'étape suivante dans la conception du programme. La division, du programme en tâches, constitue la conception initiale du programme.

6 Ecriture du programme :

La phase finale de la conception du programme est d'écrire le code source pour le programme. Dans cette étape, le langage symbolique ou pseudo-code pour les modules est traduit en instructions d'un langage de programmation, tels que PASCAL, C, ou Matlab etc...

Comme partie du code source, on doit inclure la documentation sous forme de commentaire qui décrit ce que les différentes sections du programme sont supposées faire. En plus, le code source peut inclure un code de débogage pour tester opération du programme et permettre de trouver les erreurs (bogues) de programmation.

Une fois que le programme fonctionne correctement, on doit éliminer le code de débogage. Mais la documentation devra rester comme une partie permanente du code source pour faciliter la maintenance et la modification du code source à un autre programmeur.

7 Exemple de conception d'un programme :

Problème :

Ecrire un programme qui calcule la surface d'un triangle dont les dimensions sont spécifiées par l'utilisateur.

Analyse du problème :

Un triangle est spécifié d'habitude par deux dimensions, la hauteur et la base. Celles-ci sont les quantités fournies par l'utilisateur. Définissons les variables correspondantes : Hauteur = hauteur

du triangle,

Base = base du triangle.

Le problème demande l'écriture d'un programme pour calculer la surface du triangle.

Définissons une variable :

La sortie : Aire = surface du triangle.

Remarquer qu'on a utilisé des noms complets pour les variables (Aire au lieu de A, Hauteur au lieu de H, etc.).

En mathématiques, il y a une tendance à utiliser des abréviations. Mais en programmation, il serait préférable d'utiliser des noms descriptifs de variables, aux dépens de formules plus longues et peut être plus complexes. En utilisant des noms descriptifs de variables, le code sera plus documenté.

Résumé de l'analyse :

Entrées du programme : Hauteur, Base.

Sortie du programme : Aire.

Algorithme : Obtenir Hauteur et Base de l'utilisateur.

Calculer la surface utilisant la formule $Aire = 0.5 * Hauteur * Base$.

Afficher la valeur d'Aire.

Le programme doit accomplir trois opérations:

Obtenir les dimensions du triangle de l'utilisateur.

Calculer la surface.

Afficher le résultat.

Ceci correspond à un profil du programme constitué de trois tâches:

Programme triangle :

Module principal: Triangle.

Module 1: Obtenir les dimensions.

Module 2: Calculer la surface.

Module 3: Afficher le résultat.

Le profil ci-dessus donne le pseudo-code pour le module principal, mais il ne contient pas les détails des différentes tâches. Ceux-ci sont décrits par le pseudo-code des sous-modules suivants :

Module 1: Obtenir les dimensions

Afficher 'Entrer les dimensions:'

Afficher 'Hauteur = '

Afficher 'Base = '

Entrer la valeur de Hauteur

Entrer la valeur de Base

Module 2: Calculer la surface

Aire = 0.5 * Hauteur * Base Module 3: Afficher le résultat Afficher 'La surface est égale'

Afficher la valeur d'Aire

La conception exige deux niveaux de détail, le niveau module principal et le niveau sous-module.