

Université Mouhamed Boudiaf de M'sila.

Faculté de Technologie.

Matière : TP MÉTHODES NUMÉRIQUES.

2^{ème} Année Licence (LMD) - 2019/2020.

TP N° 1 : RAPPELS ET HISTORIQUES SUR MATLAB (COMMANDES STRUCTURÉES DANS MATLAB)

1 Introduction :

MATLAB est une abréviation de Matrix LABoratory. Écrit à l'origine, en Fortran, par C. Moler, MATLAB était destiné à faciliter l'accès au logiciel matriciel développé dans les projets LINPACK et EISPACK. La version actuelle, écrite en C par the MathWorks Inc., existe en version professionnelle et en version étudiant. Sa disponibilité est assurée sur plusieurs plateformes : Sun, Bull, HP, IBM, compatibles PC (DOS, Unix ou Windows), Macintosh, iMac et plusieurs machines parallèles.

MATLAB est un environnement puissant, complet et facile à utiliser destiné au calcul scientifique. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant calcul numérique et visualisation. C'est un environnement performant, ouvert et programmable qui permet de remarquables gains de productivité et de créativité. MATLAB est un environnement complet, ouvert et extensible pour le calcul et la visualisation. Il dispose de plusieurs centaines (voire milliers, selon les versions et les modules optionnels autour du noyau Matlab) de fonctions mathématiques, scientifiques et techniques. L'approche matricielle de MATLAB permet de traiter les données sans aucune limitation de taille et de réaliser des calculs numériques et symboliques de façon fiable et rapide. Grâce aux fonctions graphiques de MATLAB, il devient très facile de modifier interactivement les différents paramètres des graphiques pour les adapter selon nos souhaits.

L'approche ouverte de MATLAB permet de construire un outil sur mesure. On peut inspecter le code source et les algorithmes des bibliothèques de fonctions (Toolboxes), modifier des fonctions existantes et ajouter d'autres.

MATLAB possède son propre langage, intuitif et naturel qui permet des gains de temps de CPU spectaculaires par rapport à des langages comme le C, le Turbo Pascal et le Fortran. Avec MATLAB, on peut faire des liaisons de façon dynamique, à des programmes C ou Fortran, échanger des données avec d'autres applications (via la DDE : MATLAB serveur ou client) ou utiliser MATLAB comme moteur d'analyse et de visualisation.

MATLAB comprend aussi un ensemble d'outils spécifiques à des domaines, appelés Toolboxes (ou Boîtes à Outils). Indispensables à la plupart des utilisateurs, les Boîtes à Outils sont des collections de fonctions qui étendent l'environnement MATLAB pour résoudre des catégories spécifiques de problèmes. Les domaines couverts sont très variés et comprennent notamment le traitement du signal, l'automatique, l'identification de systèmes, les réseaux de neurones, la logique floue, le calcul de structure, les statistiques, etc...

MATLAB fait également partie d'un ensemble d'outils intégrés dédiés au Traitement du Signal. En complément du noyau de calcul MATLAB, l'environnement comprend des modules optionnels qui sont parfaitement intégrés à l'ensemble :

1. une vaste gamme de bibliothèques de fonctions spécialisées (Toolboxes),
2. Simulink, un environnement puissant de modélisation basée sur les schémas-blocs et de simulation de systèmes dynamiques linéaires et non linéaires,
3. Des bibliothèques de blocs Simulink spécialisés (Blocksets),
4. D'autres modules dont un Compilateur, un générateur de code C, un accélérateur,...
5. Un ensemble d'outils intégrés dédiés au Traitement du Signal : le DSP Workshop.

1.1 Quelles sont les particularités de MATLAB ?

MATLAB permet le travail interactif soit en mode commande, soit en mode programmation ; tout en ayant toujours la possibilité de faire des visualisations graphiques. Considéré comme un des meilleurs langages de programmations (C ou Fortran), MATLAB possède les particularités suivantes par rapport à ces langages :

- la programmation facile,
- la continuité parmi les valeurs entières, réelles et complexes,
- la gamme étendue des nombres et leurs précisions,
- la bibliothèque mathématique très compréhensive,
- l'outil graphique qui inclus les fonctions d'interface graphique et les utilitaires,
- la possibilité de liaison avec les autres langages classiques de programmations (C ou Fortran).

Dans MATLAB, aucune déclaration n'est à effectuer sur les nombres. En effet, il n'existe pas de distinction entre les nombres entiers, les nombres réels, les nombres complexes et la simple ou double précision. Cette caractéristique rend le mode de programmation très facile et très rapide. En Fortran par exemple, une subroutine est presque nécessaire pour chaque variable simple ou double précision, entière, réelle ou complexe. Dans MATLAB, aucune nécessité n'est demandée pour la séparation de ces variables.

La bibliothèque des fonctions mathématiques dans MATLAB donne des analyses mathématiques très simples. En effet, l'utilisateur peut exécuter dans le mode commande n'importe quelle fonction mathématique se trouvant dans la bibliothèque sans avoir à recourir à la programmation.

Pour l'interface graphique, des représentations scientifiques et même artistiques des objets peuvent être créées sur l'écran en utilisant les expressions mathématiques. Les graphiques sur MATLAB sont simples et attirent l'attention des utilisateurs, vu les possibilités importantes offertes par ce logiciel.

1.2 MATLAB peut-il s'en passer de la nécessité de Fortran ou du C ?

La réponse est non. En effet, le Fortran ou le C sont des langages importants pour les calculs de haute performance qui nécessitent une grande mémoire et un temps de calcul très long. Sans compilateur, les calculs sur MATLAB sont relativement lents par rapport au Fortran ou au C si les programmes comportent des boucles. Il est donc conseillé d'éviter les boucles, surtout si celles-ci est grande.

1.3 écriture d'un programme m-file par MATLAB :

En MATLAB, les programmes se terminent par une extension '.m' dans le nom du fichier programme. Aucune compilation n'est à faire avant l'exécution du programme. Au cours de l'exécution, un message d'erreur apparaît et indique les lieux où se trouvent les erreurs. Pour lancer l'exécution du programme, il faut se mettre toujours dans le même répertoire où se trouve ce programme.

2 Commandes structurées dans Matlab (Boucles et structures)

2.1 Manipulation 01 : Multiplication de matrices

Le produit matriciel s'en déduit : le produit de la matrice $A(n \times m)$ par la matrice $B(m \times p)$ est la matrice $C(n \times p)$ telle que l'élément c_{ij} est égal au produit scalaire de la ligne i de la matrice A par la colonne j de la matrice B .

$$c_{ij} = \sum_{k=1}^m a_{ik} * b_{kj}; \text{ avec } i = 1 \dots n \text{ et } j = 1 \dots p$$

Programme 01 : Ecrire Un programme en Matlab (m-file : sur le nom *tp1prog1.m*) qui permet de faire le produit de deux matrices en utilisant la boucle **for**. Refaire le programme en utilisant la boucle **while**.

2.2 Manipulation 02 : somme des n termes d'une série

On donne la série des n termes définie comme suit :

$$S1 = \sum_{k=1}^n \frac{(-1)^k k}{2^k}$$

Programme 02 : Ecrire un programme en Matlab (m-file : sur le nom *tp1prog2.m*) qui permet de faire la somme des n termes de la série précédente $S1$ en utilisant la boucle **while**. On exécute le programme pour $n = 4, n = 20$ et $n = 100$. Refaire le programme en utilisant la boucle **for**.

2.3 Manipulation 03 : Lire des variables et faire la somme des n termes d'une série

Pour lire des variable par Matlab en utilisant la commande *input* et pour afficher un résultat on utilise la commande *fprintf* ou *disp*.

Programme 03 : Ecrivez un programme en Matlab (m-file : sur le nom *tp1prog3.m*) qui permet de demander deux entiers a et b qui affiche le résultat de la somme $S2$ avec : $S2 = \sum_{k=1}^b k^a$ en utilisant la boucle **for**. Refaire le programme en utilisant la boucle **while**.

Exercice : Dans un script Matlab, écrire un simple programme (sur le nom : *tp1exercice.m*) qui fait un test sur un nombre : pair ou impaire ?