

Laboratoire de test et analyse

GNG1503 – Génie de la Conception

Objectif

Pour pratiquer l'évaluation de métriques et l'interprétation de résultats de tests ainsi que d'apprendre les bases de l'environnement du logiciel Matlab et résoudre des problèmes mathématiques.

Contexte

MATLAB est un langage de haute performance pour l'informatique technique. Il intègre le calcul, la visualisation et la programmation dans un environnement facile à utiliser où les problèmes et les solutions sont exprimés dans une notation mathématique familière. Les applications typiques comprennent: Mathématiques, calcul, développement d'algorithmes, modélisation, simulation, prototypage, analyse de données, exploration, visualisation, graphisme scientifique et technique, développement d'applications et la construction d'interface graphique par l'utilisateur.

MATLAB est un système interactif dont l'élément de données de base est un tableau qui ne nécessite pas de dimensionnement. Cela vous permet de résoudre de nombreux problèmes de calcul technique, en particulier ceux qui ont des formulations matricielle et vectorielle, dans une fraction du temps qu'il faudrait pour écrire un programme dans un scalaire sans langage interactif tel que C ou FORTRAN.

Le nom MATLAB est synonyme de matrice de laboratoire. MATLAB a été initialement écrit pour fournir un accès facile au logiciel matriciel. MATLAB a évolué sur une période de plusieurs années avec l'apport de nombreux utilisateurs. Dans les environnements universitaires, il est l'outil d'enseignement standard pour les cours d'introduction et avancés en mathématiques, en génie et en sciences. Dans l'industrie, MATLAB est l'outil de choix pour la recherche, le développement et l'analyse à haute productivité.

Le langage MATLAB est un langage matriciel/tableur de haut niveau avec des instructions de flux de commande, des fonctions, des structures de données, des entrées / sorties et des fonctions de programmation orientée objet. Il permet à la fois de réaliser «des petits programmes» pour créer rapidement des programmes, et «des grands programmes» pour créer des programmes d'application complets et complets.

Pré-laboratoire

Avant d'arriver au laboratoire, l'étudiant devrait revoir et se familiariser avec le laboratoire et les procédures. Il est **fortement** recommandé de lire la première partie de l'Appendice I pour comprendre la base de MATLAB.

Téléchargements

Pour ce laboratoire, vous devrez accéder au logiciel MATLAB. Si vous ne l'avez pas installé sur votre ordinateur, vous pouvez y accéder à distance via la plate-forme universitaire.

Vous trouverez ci-dessous un lien qui vous guidera tout au long du processus d'accès à MATLAB.
<https://www2.uottawa.ca/notre-universite/technologies-information/a-distance/lab>

Vous pouvez trouver toutes les informations sur MATLAB dans le site officiel de MATLAB:
https://www.mathworks.com/help/matlab/learn_matlab/

Questions de Révision

Qu'est-ce que le terme MATLAB représente?

Quelles sont les capacités de MATLAB?

Quel est l'élément mathématique de base que MATLAB utilise?

Comment est appelé la fenêtre où tu tapes des commandes?

Comment tu effaces l'espace de travail?

IMPORTANT: Pendant le laboratoire, vous allez prendre des captures d'écran pour chaque partie. Coller ces captures d'écran dans un document Word (ou autre) à mesure que vous avancez à travers le manuel de laboratoire pour que vous puissiez créer un document PDF de toutes vos captures d'écran de votre travail.

SCÉNARIO: Un nouveau pont piétonnier doit être conçu pour relier l'île Cummings au parc Riverain sur la rivière Rideau. La distance qu'il doit parcourir est d'environ 60 m et doit accueillir la largeur d'un véhicule d'entretien. Il doit pouvoir supporter au moins 3 kPa de charge piétonnière [1]. En supposant les pires conditions d'Ottawa, il doit également être capable de résister à des pressions de vent horaires de 0,41 kPa et à une charge de neige de 2,4 kPa [2]. La Ville d'Ottawa insiste sur le fait que le pont doit être joli et soit les couleurs des 67's d'Ottawa¹. Selon les normes des ponts piétonniers, la déviation verticale due à la charge des piétons et la déviation horizontale due à la charge du vent ne doivent pas dépasser 1/360 de la longueur de la portée [1].

Pour les besoins de ce lab, mettez tout à l'échelle par un facteur de 240 et supposez un facteur de sécurité de 1,5.

Par exemple, si nous voulons connaître la charge verticale minimale que le pont doit pouvoir supporter:

$$\begin{aligned} charge_v &= \text{piéton} + \text{neige} = 3 + 2.4 = 5.4kPa \times 1.5 \text{ sécurité} = 8.1kPa \div 240 \\ &= 33.75Pa \text{ (petite échelle)} \\ force_v &= charge_v \times aire = 33.75Pa \times \left(\frac{60m \text{ long} \times 4m \text{ large}}{240} \right) = 33.75N \text{ (petite échelle)} \end{aligned}$$

¹ <https://teamcolorcodes.com/ottawa-67s-color-codes/>

Partie A- Plan de test

1. Sur la base du scénario ci-dessus, une spécification de conception et une valeur cible ont été développées (voir le tableau suivant). **Remplissez** le tableau suivant avec deux autres spécifications de conception basées sur votre compréhension du problème. L'une des deux spécifications de conception doit être non-fonctionnelle.
 - a. Ne pas utiliser la longueur du pont comme spécification de conception.

Tableau 1: Spécifications de conception et valeurs cibles

ID	Spécifications de conception	Relation (=, <, >)	Valeur	Unité	Méthode de vérification
1	Capacité de charge verticale	>	33.75	N	Test (déflexion < 0.7mm)
2					
3					

2. Sur la base des trois spécifications de conception, prévoyez 3 tests à effectuer (un pour chaque spécification). **Remplissez** le tableau suivant. Remarque: assurez-vous de regarder les matériaux et équipements disponibles dans le laboratoire à utiliser pour ces tests.

Tableau 2: Plans de test

Test ID	Objectif (Pourquoi)	Description de la méthode de test et les matériaux nécessaires (Quoi)	Description des résultats à être enregistrés (Comment)	Durée estimée du test (Quand)
1	Valider spec 1			
2	Valider spec 2			
3	Valider spec 3			

Partie B- Collecte de données

Voici le concept qui DOIT être utilisé pour le prototype :

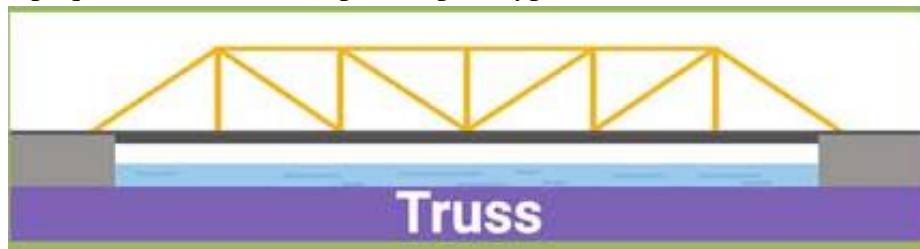


Figure 1 : Concept de pont

1. Basé des tests décrits ci-dessus, **dessinez** à petite échelle une conception détaillée du prototype (basé sur le concept montré). Vous aurez peut-être besoin de plusieurs prototypes pour tester différents aspects, mais idéalement, vous pouvez en utiliser un pour les 3. Montrez-le a votre AE ou GP pour approbation.
 - a. **Important** : la construction du pont ne doit pas dépasser 300mm x 300mm x 150mm.
2. **Construisez** le(s) prototype(s) à partir des matériaux disponibles dans le laboratoire (*ne passez pas plus de 45 minutes sur cette étape*).
3. **Prenez** une photo du ou des prototypes.
3. **Effectuez** les 3 tests décrits dans le Tableau 2 en incrémentant lentement les valeurs avant d'atteindre la valeur cible et notez-les dans le tableau suivant. Vous pouvez ajouter plus de points de données si vous voulez. **Notez** vos observations et vos résultats à chaque étape.

Tableau 3: Données enregistrées

Incrément	Test 1		Test 2		Test 3	
	x	y	x	y	x	y
1						
2						
3						
4						
5						

Partie C- Analyse de données

Partie C.1- Concepts généraux et syntaxe de base MATLAB

Lire les parties suivantes dans Annexe II:

- Comprendre l'interface MATLAB
- Travailler avec les variables de MATLAB
- Commandes de base
- Travailler avec les opérations matricielles et scalaires

1. **Écrire** trois matrices X, Y et Z. X est une matrice 3x3, Y est une matrice 3x4 et Z est une matrice 4x3. **Imprimez-les** sur la fenêtre de commande (Command Window).
2. **Prenez** une capture d'écran de la fenêtre de commande. Vous allez l'utiliser pour votre rapport de laboratoire.
3. **Effacer** le contenu de la fenêtre de commande.
4. **Écrire** le code suivant:

```
x = 1:2:30
```

```
y = 1:3:30
```

```
z = 1:4:30
```

Question 1. Expliquez ce qui se passe dans chaque cas.

5. **Effacer** le contenu de la fenêtre de commande.
6. Considérez trois matrices A = 'Faculté', B = 'de' et C = 'génie'. Trouvez un moyen **d'imprimer** sur la fenêtre de commande le texte: "Faculté de génie". Pensez au laboratoire de chiffrier où vous deviez combiner un prénom et un nom.

*Notez bien: les espaces fonctionnent seulement au début d'une chaîne de texte et non à la fin.

7. **Prenez** une capture d'écran de la fenêtre de commande. Vous allez l'utiliser pour votre rapport de laboratoire.
8. **Effacer** le contenu de la fenêtre de commande.
9. **Écrire** deux matrices Y et Z où:

```
Y =      2      1      4      Z =      2      3      6
      3      2      1      1      5      7
      1      4      3      3      4      1
```

10. En utilisant MATLAB, **effectuez** les opérations suivantes:

```
T = Y+Z
```

$$U = Y-Z$$

$$V = Y*Z$$

$$W = Y.*Z$$

11. **Prenez** une capture d'écran de la fenêtre de commande. Vous allez l'utiliser pour votre rapport de laboratoire.

Question 2. Quelle est la différence entre V et W?

Question 3. Quelle est la valeur dans la matrice T correspondant à la ligne 3 colonne 2? Écrivez votre réponse, ainsi que la syntaxe MATLAB utilisée pour accéder à cet index.

Question 4. Qu'est-ce que le fait de taper W(:,2) dans la fenêtre de commande et d'appuyer sur entrer vous donne? Expliquer ce que le deux-point fait lors de l'accès aux indices.

Partie C.2- Résultats de test

Lire les parties suivantes dans Annexe II:

- Affichage et modification des programmes
- Création de graphiques de base
- Ajout d'annotations

1. **Ouvrir** un nouveau script.
2. **Écrire** le code suivant:

$$x = -20:2:20$$

$$y = (x.^2)$$

$$\text{plot}(x,y)$$

3. **Exécutez-le** et observez le graphique.
4. **Ajoutez** des informations à votre graphique: axe des x, axe des y et le titre.
5. **Prenez** une capture d'écran de la fenêtre de commande. Vous allez l'utiliser pour votre rapport de laboratoire.
6. **Entrer** les données qui ont été collectés dans la partie B dans des variables pour le premier test en écrivant les données x et y dans 2 matrices de longueur N,, N étant le nombre d'incrémentés utilisés dans la collecte de données (ex: Test1_x = [1; 2; 3; 4;...; N] and Test1_y = [1; 2; 3; 4;...; N]).

7. **Ajuster** les étiquettes de tracé en conséquence.
8. **Exécutez** le code.
9. **Prenez** une capture d'écran du graphique.
10. **Répétez** les étapes 6 à 9 pour les deux autres tests.

Question 5. Expliquez les résultats du graphique et si les tests ont réussi.

Question 6. Que représentent les résultats par rapport à vos spécifications cibles et quelles modifications pourriez-vous devoir apporter pour améliorer votre prototype?

Soumission

Soumettez ce manuel de laboratoire comme un PDF imprimé avec les réponses aux questions sur les lignes fournies en téléchargeant sur Brightspace avant la date limite. Vous devez aussi inclure dans votre soumission le document PDF avec vos captures d'écran. Tous les fichiers téléchargés devraient être soumis comme une seule soumission avec plusieurs téléchargements, plutôt que des soumissions séparées.

Références

- [1] A. de la Chevrotière and L. Gérin, "A Review of Two Codes For the Design of Pedestrian Bridges," 2021. [Online]. Available: https://maadigroup.com/wp-content/uploads/2022/10/MAADI_Group_Review_of_2_codes_R3.pdf. [Accessed 18 2 2023].

[2] Dlubal, "Load zones for snow, wind, earthquake," [Online]. Available: <https://www.dlubal.com/en/load-zones-for-snow-wind-earthquake/snow-nbc-2020.html#¢er=45.427105896042356,-75.64542997176247&zoom=13&marker=45.4321656075918,-75.671007516928>. [Accessed 18 2 2023].

Annexe I : Tests quantitatifs et qualitatifs

Les tests quantitatifs et les tests qualitatifs sont deux approches distinctes utilisées pour recueillir et analyser des données dans la recherche, l'évaluation et divers domaines. La principale différence entre eux réside dans le type de données qu'ils génèrent et les méthodes utilisées pour l'analyse des données.

Tests quantitatifs :

Les tests quantitatifs sont conçus pour collecter et analyser des données numériques. Ils visent à quantifier les variables et à fournir des informations statistiques permettant de comprendre les modèles, les relations et les tendances au sein des données. Les données obtenues à partir des tests quantitatifs sont généralement exprimées sous forme numérique et peuvent faire l'objet d'une analyse statistique.

- Exemples de tests quantitatifs :
 - Test de vitesse : L'ingénieur mesure le temps nécessaire à la décapeuse pour parcourir une distance donnée, par exemple d'un bout à l'autre du chantier. Les données recueillies sont la vitesse moyenne de la grappe, exprimée en kilomètres ou en miles par heure.
 - Expériences : Dans la recherche expérimentale, les données recueillies au moyen de mesures (par exemple, la taille, le poids, le temps) sont quantitatives et peuvent être analysées statistiquement pour déterminer s'il existe des différences significatives entre les groupes expérimentaux.
 - Test d'efficacité énergétique : L'ingénieur enregistre la quantité de carburant consommée par le racleur au cours d'une période de fonctionnement déterminée. Il mesure la consommation de carburant en litres ou en gallons par heure, ce qui donne une mesure quantitative de son efficacité.

Tests qualitatifs :

Les tests qualitatifs sont utilisés pour collecter des données non numériques afin de mieux comprendre les pensées, les opinions, les sentiments ou les comportements des personnes. Ces tests s'attachent à comprendre le contexte, les significations et les interprétations des données. Les données qualitatives se présentent souvent sous la forme de textes, d'images, d'enregistrements audio ou vidéo.

- Exemples de tests qualitatifs :
 - Étude d'observation : L'ingénieur observe la décapeuse en action sur le chantier, en notant la qualité de ses manœuvres, de sa manipulation des charges et de son adaptation aux différents terrains. Il peut également observer le langage corporel de l'opérateur pour évaluer son aisance et sa confiance.
 - Analyse comparative : L'ingénieur compare les performances de la décapeuse à des modèles similaires ou à d'autres types d'engins de terrassement utilisés sur le

chantier. Cette évaluation qualitative permet d'identifier les avantages et les limites uniques de la gratte.

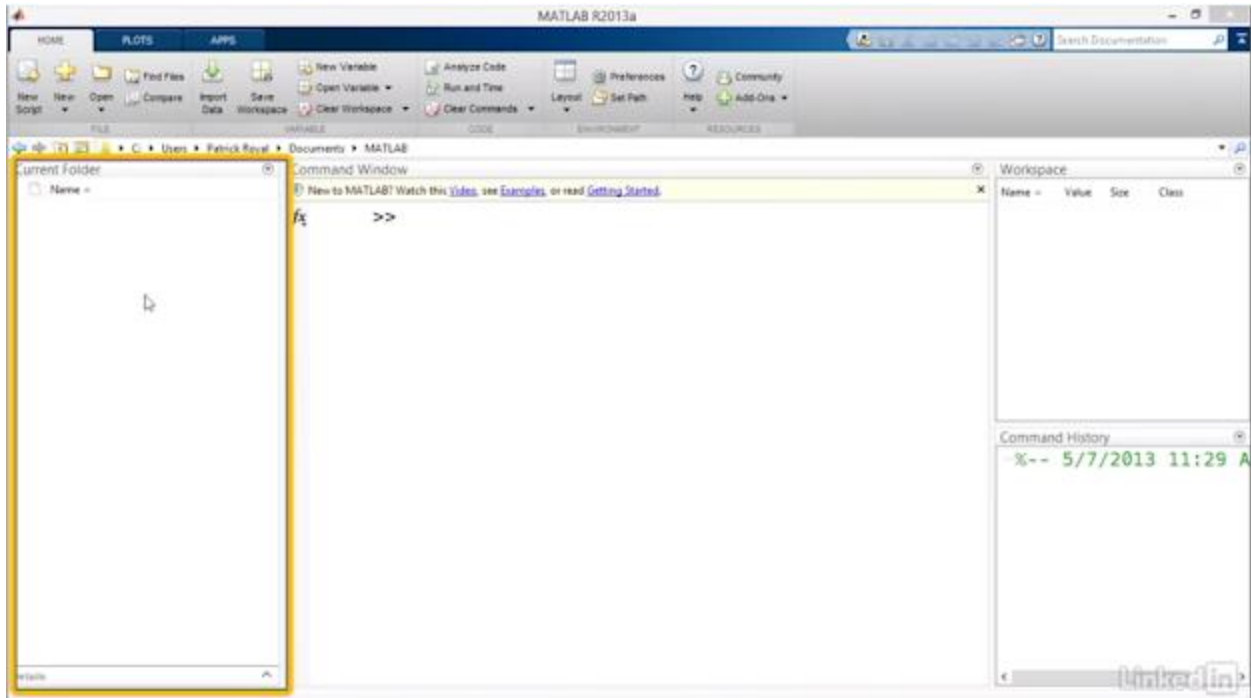
- Analyse de contenu : Analyse d'un contenu textuel ou visuel afin d'identifier des thèmes, des modèles et des significations, comme l'analyse de réponses à des enquêtes ouvertes ou de contenus médiatiques.

Annexe II : Documentation MATLAB

Ce document contient toutes les informations nécessaires pour répondre aux questions du laboratoire.

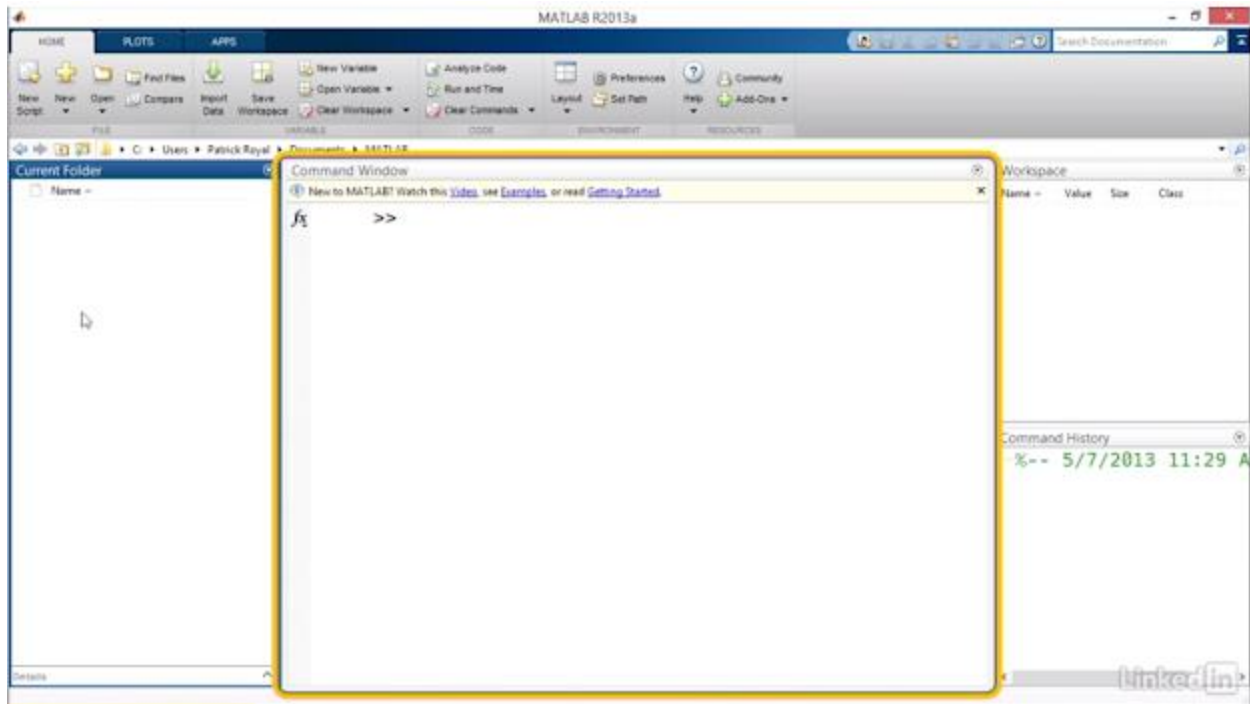
Comprendre l'interface MATLAB

Lorsque vous ouvrez MATLAB pour la première fois, vous verrez que l'interface est divisée en quatre fenêtres. Sur la gauche, la fenêtre du dossier actuel (Current Folder).



C'est là que tous les fichiers MATLAB pertinents sont placés dans le dossier où vous vous trouvez. Mais c'est aussi là que toutes les fonctions et scripts que vous créez seront stockés. Il est important de vous assurer que vous êtes dans le bon dossier lorsque vous exécutez des scripts et des fonctions car MATLAB considérera le dossier actif dans le but d'exécuter des fonctions. Par défaut, MATLAB définira le dossier actuel dans un dossier nommé MATLAB dans votre dossier Documents.

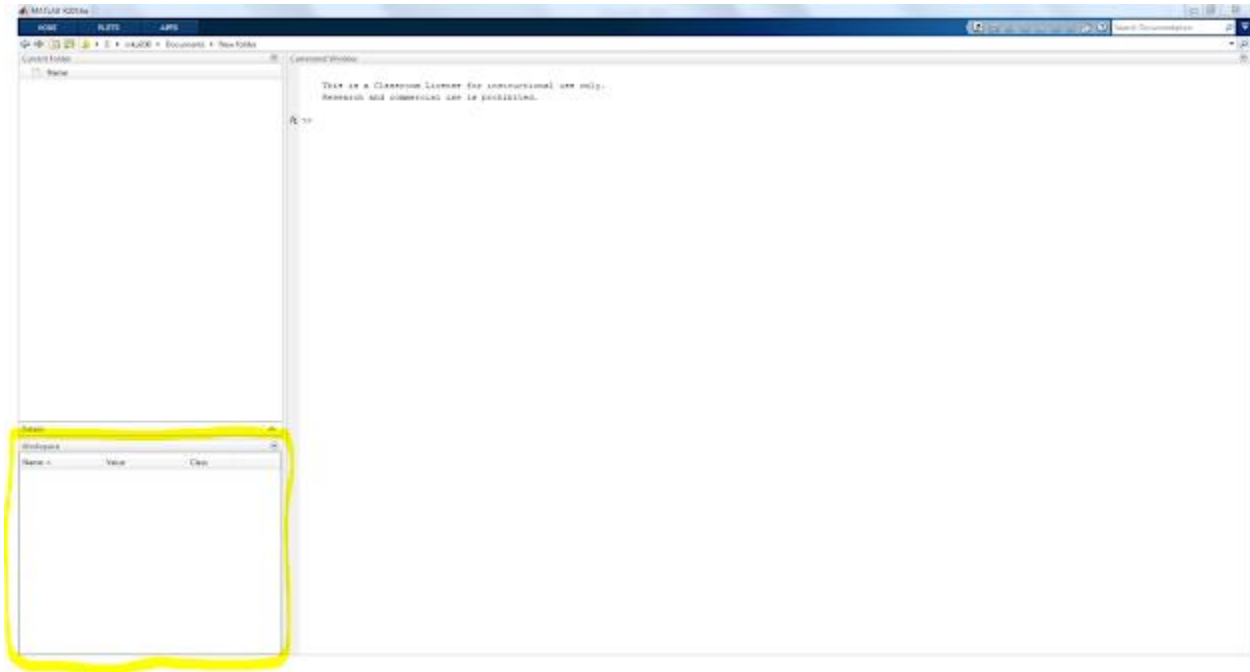
Au milieu de l'écran se trouve la fenêtre de commande.



Cette zone contient toutes vos entrées et sorties lorsque vous exécutez diverses fonctions. Chaque fois que vous exécutez un script ou une fonction, toute sortie sera affichée ici. Techniquement, la fenêtre de commande a toutes les mêmes fonctions que l'éditeur de script. Ainsi, il est possible d'écrire des programmes directement sur cette ligne. Cependant, dans la pratique, vous voudrez probablement quelque chose que vous pouvez enregistrer, modifier et exécuter plusieurs fois. À tout moment, si la fenêtre de commande devient trop encombrée, vous pouvez la supprimer en tapant `clc` et en appuyant sur Entrée, ou en cliquant sur la flèche dans le coin supérieur droit de la vignette et en choisissant Effacer la fenêtre de commande (Clear Command Window).



En bas à gauche de l'écran se trouve l'espace de travail (Workspace).



Il contient une liste de toutes les variables actives de votre simulation. Pour modifier les informations disponibles sur vos variables, faites un clic droit sur les colonnes et sélectionnez les informations que vous souhaitez afficher.

L'espace de travail affiche toutes les variables que vous avez utilisées dans une session MATLAB donnée, quel que soit le script ou la fonction dont elles proviennent. Ainsi, après un certain temps, il peut être encombré d'anciennes données. Pour nettoyer l'espace de travail, cliquez sur la flèche déroulante en haut à droite de l'écran de l'espace de travail et choisissez Effacer l'espace de travail (Clear Workspace). Après confirmation, MATLAB supprimera toutes les variables existantes.

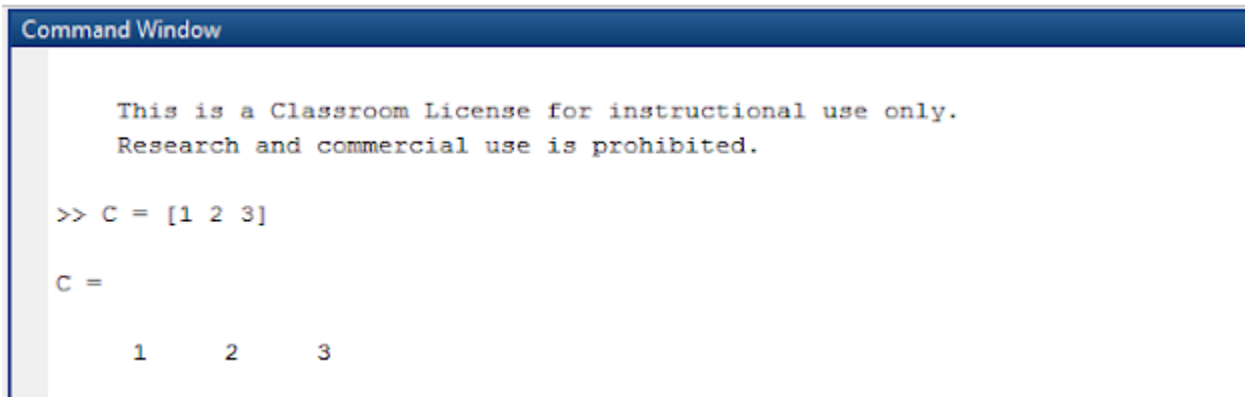
Travailler avec les variables de MATLAB

Voyons comment créer et manipuler des variables MATLAB de base. Pour commencer, nous allons avoir besoin de certaines variables. Pour l'instant, nous allons utiliser la méthode la plus simple de création de variables, en tapant dans la fenêtre de commande. Définir une nouvelle variable dans MATLAB est très simple. Contrairement à d'autres langages de programmation comme C ou Java, vous n'avez pas besoin d'ajouter une instruction de déclaration de variable. En d'autres termes, vous n'avez pas besoin de dire quoi que ce soit à MATLAB, cette variable est un entier ou cette variable est une chaîne. Au lieu de cela, tout ce que vous avez à faire est de dire à MATLAB à quoi correspond votre variable, afin que vous puissiez le faire maintenant.

Dans la fenêtre de commande, tapez $A \text{ égal à } 1$ et appuyez sur Entrée. La fenêtre de commande (Fenêtre de commande) vous donne le résultat $A \text{ est égal à } 1$. L'espace de travail affiche maintenant

une nouvelle variable appelée A avec une valeur de 1 et une taille de 1 par 1. Cela dit simplement que A est un scalaire.

MATLAB prend également en charge les variables qui sont des vecteurs ou des matrices. Pour créer un vecteur dans votre déclaration de variable, placez des crochets autour de votre terme et séparez chaque valeur entre crochets par une virgule ou un espace. Par exemple, $C = [1\ 2\ 3]$ provoque la création de C en tant que vecteur un par trois avec les trois valeurs qui y sont stockées.



```
Command Window

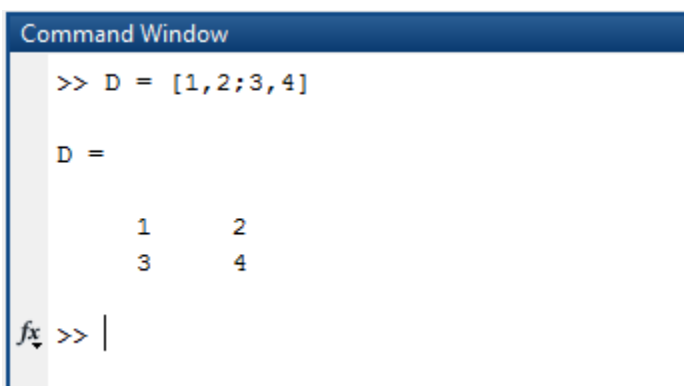
This is a Classroom License for instructional use only.
Research and commercial use is prohibited.

>> C = [1 2 3]

C =

     1     2     3
```

Pour créer une matrice, le même principe s'applique, sauf que vous utilisez un point-virgule pour séparer les lignes. Par exemple, $D = [1,2; 3,4]$ fait que D soit créé comme une matrice 2 par 2 avec ces valeurs.



```
Command Window

>> D = [1,2;3,4]

D =

     1     2
     3     4

fx >> |
```

Commandes de base

Pour définir des variables sous forme de texte plutôt que de valeur numérique, placez des guillemets simples autour du texte que vous définissez. Si vous voulez que votre chaîne soit variable plutôt que fixe, les fonctions num2str et strcat seront utiles. num2str convertit une valeur numérique en une chaîne, qui ajoute un nombre non prédéterminé à la chaîne. Strcat concatène deux chaînes strictement pour former une chaîne plus longue.

```

>> a = 'Hello'

a =

Hello

>> b = 'World'

b =

World

>> c = strcat(a,b)

c =

HelloWorld

```

Contrairement à Java, MATLAB ne vous oblige pas à placer un point-virgule à la fin de chaque ligne. Si vous le faites, MATLAB l'interprétera comme une commande pour supprimer la sortie pour tout ce qui se passe sur la ligne précédente. C'est une fonctionnalité extrêmement utile, car par défaut, MATLAB affiche les résultats de chaque fonction, équation, définition de variable ou boucle pendant le programme. Pour les scripts plus volumineux, cela peut rapidement submerger la fenêtre de commande et rendre le programme inutilisable.

En général, à moins que vous ne souhaitiez spécifiquement voir la sortie d'une ligne, il est recommandé de terminer chaque ligne par un point-virgule.

```

>> a = 'Hello';
>> b = 'World';
>> c = strcat(a,b);
>> % If we want to see the value of a we just write :
>> c

c =

HelloWorld

```

Travailler avec des opérations matricielles et scalaires

Jetons un coup d'œil aux opérations matricielles de base utilisées par MATLAB. Puisque MATLAB traite toutes les variables comme s'il s'agissait d'une matrice, il est important de distinguer les opérations telles que la multiplication de la matrice par rapport à la multiplication du scalaire. Nous allons pratiquer différents types d'opérations matricielles.

Utilisons les matrices ci-dessous:

```
Command Window
>> a = [1,2,3;5,6,7]

a =

     1     2     3
     5     6     7

>> b = [1,1,1;0,4,5]

b =

     1     1     1
     0     4     5

>> c = [3,1;2,2;0,0]

c =

     3     1
     2     2
     0     0

>> d = [4,4,2;-1,2,4]

d =

     4     4     2
    -1     2     4

>> e = 1:4

e =

     1     2     3     4

>> f = [2,1,4,3]

f =

     2     1     4     3
```

Pour commencer, tapez `a + b` dans la fenêtre de commande. Puisque la matrice `a` et la matrice `b` ont la même dimension, MATLAB l'interprétera comme une addition par morceaux et ajoutera simplement les valeurs correspondantes dans chaque cellule des matrices.

```
Command Window
>> a+b

ans =

     2     3     4
     5    10    12
```

D'autre part, si vous tapez `a + 2`, MATLAB l'interprétera comme une addition scalaire, de sorte qu'il en ajoutera deux à chaque cellule de `A`.

```
Command Window
>> a+2

ans =

     3     4     5
     7     8     9
```

Le même principe s'applique à la multiplication des matrices. Si l'on entre `c * d`, MATLAB voit que ces deux variables sont des matrices, et leurs dimensions internes correspondent (le nombre de colonnes de `c` est égal au nombre de lignes de `d`). Ainsi, il effectuera automatiquement la multiplication matricielle.

```
Command Window
>> c*d

ans =

    11    14    10
     6    12    12
     0     0     0
```

D'autre part si vous tapez `c * 2` MATLAB considère 2 comme une constante et multipliera chaque terme dans `c` par 2.

```
Command Window
>> c*2

ans =

     6     2
     4     4
     0     0
```

La multiplication matricielle est un processus assez simple. Mais que se passerait-il si nous voulions que MATLAB effectue une multiplication par morceaux des entrées de matrice correspondantes. Les matrices a et b ont les mêmes dimensions, mais si vous entrez une seule fois b, MATLAB ne sait pas que nous voulons multiplier les entrées correspondantes. Donc, cela suppose que nous voulons multiplier les matrices et afficher une erreur car les dimensions internes de ces deux matrices ne correspondent pas. Au lieu de cela, nous pouvons ajouter un point devant l'astérisque.

```
Command Window
>> a*b
Error using *
Inner matrix dimensions must agree.

>> a.*b

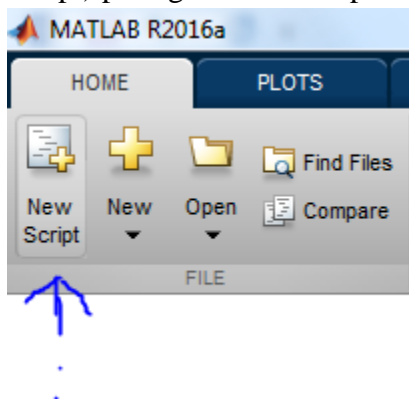
ans =

     1     2     3
     0    24    35
```

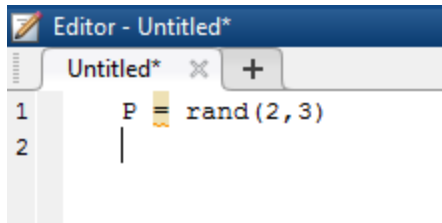
L'ajout d'un point devant toute opération, addition, soustraction, multiplication, division, exposants ou même égalité, indique à MATLAB d'exécuter l'opération par morceaux. C'est un moyen vraiment efficace de gérer de grandes quantités de données à la fois.

Affichage et modification des programmes

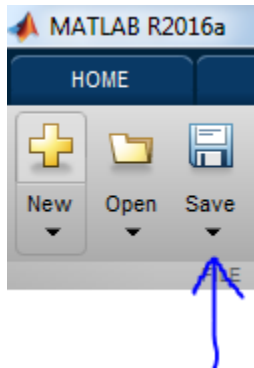
Créer un nouveau programme est facile. Cliquez simplement sur le bouton Nouveau script (New Script) pour générer le script et ouvrir automatiquement la fenêtre de l'éditeur de script.



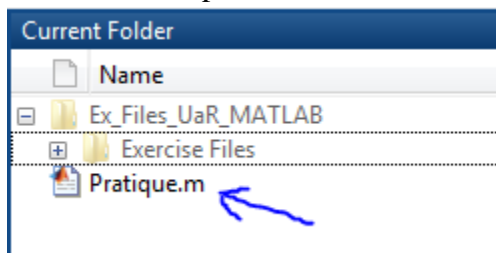
Ici, vous pouvez saisir tout code qui sera exécuté dans le cadre du programme. Par exemple, nous allons simplement créer un script très simple qui génère une matrice aléatoire deux par trois. Donc $P = \text{rand}(2,3)$.



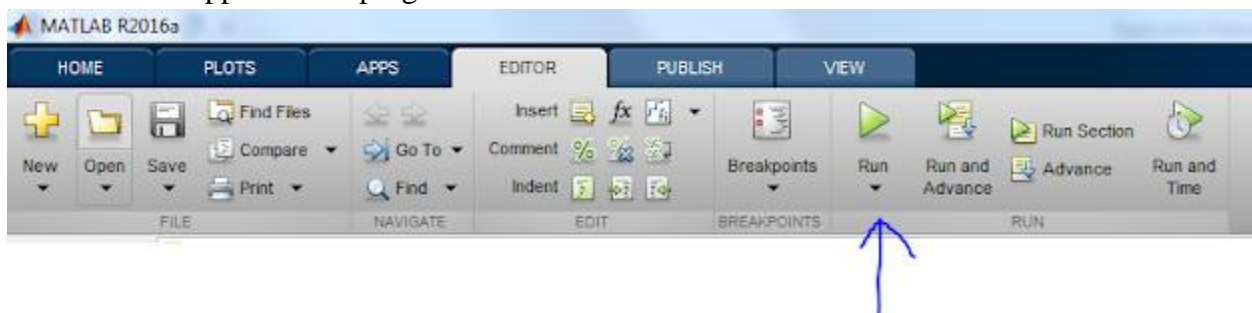
Lorsque vous êtes prêt à enregistrer, cliquez sur le bouton Enregistrer et donnez un nom au programme et il sera automatiquement enregistré dans le dossier actif.



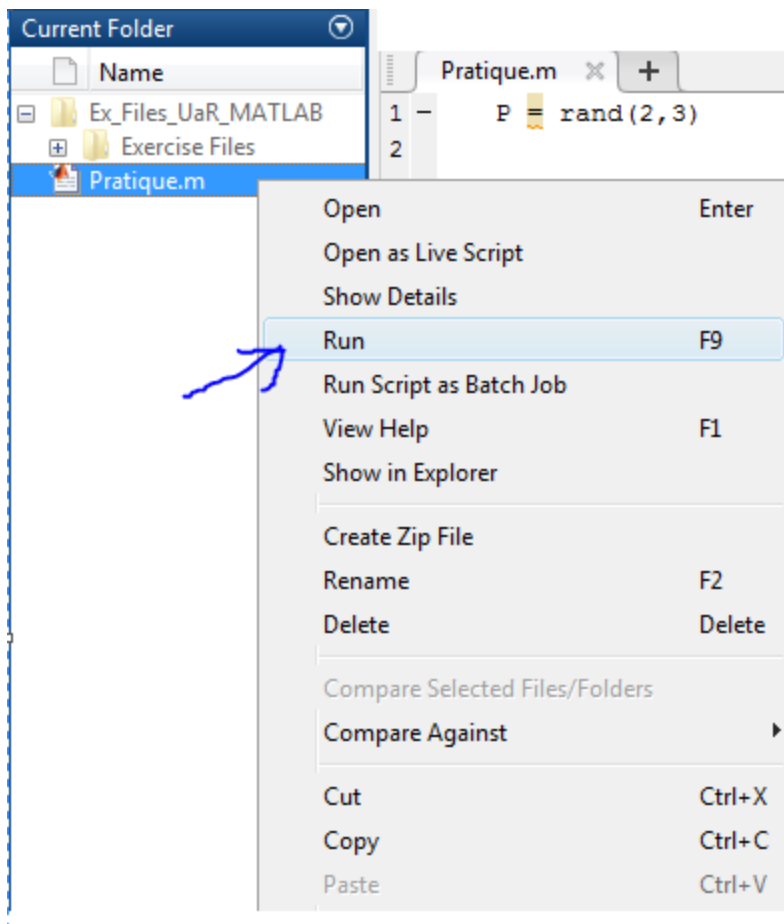
Maintenant, chaque fois que vous voulez revenir au script pour le modifier, double-cliquez simplement sur le nom dans le volet de dossier actuel et cela rouvrira la fenêtre et permettra de modifier le script exactement comme avant.



Pour exécuter le programme, il existe plusieurs options différentes. Vous pouvez d'abord cliquer sur le bouton Exécuter(Run) dans la fenêtre d'édition. Cela exécute le programme immédiatement sans fermer la fenêtre, vous permettant de voir rapidement et facilement comment les modifications apportées au programme affectent la sortie.



Deuxièmement, vous pouvez cliquer avec le bouton droit sur le script à partir de la fenêtre du dossier actuel. Et choisissez Exécuter(Run) ou appuyez sur F9.



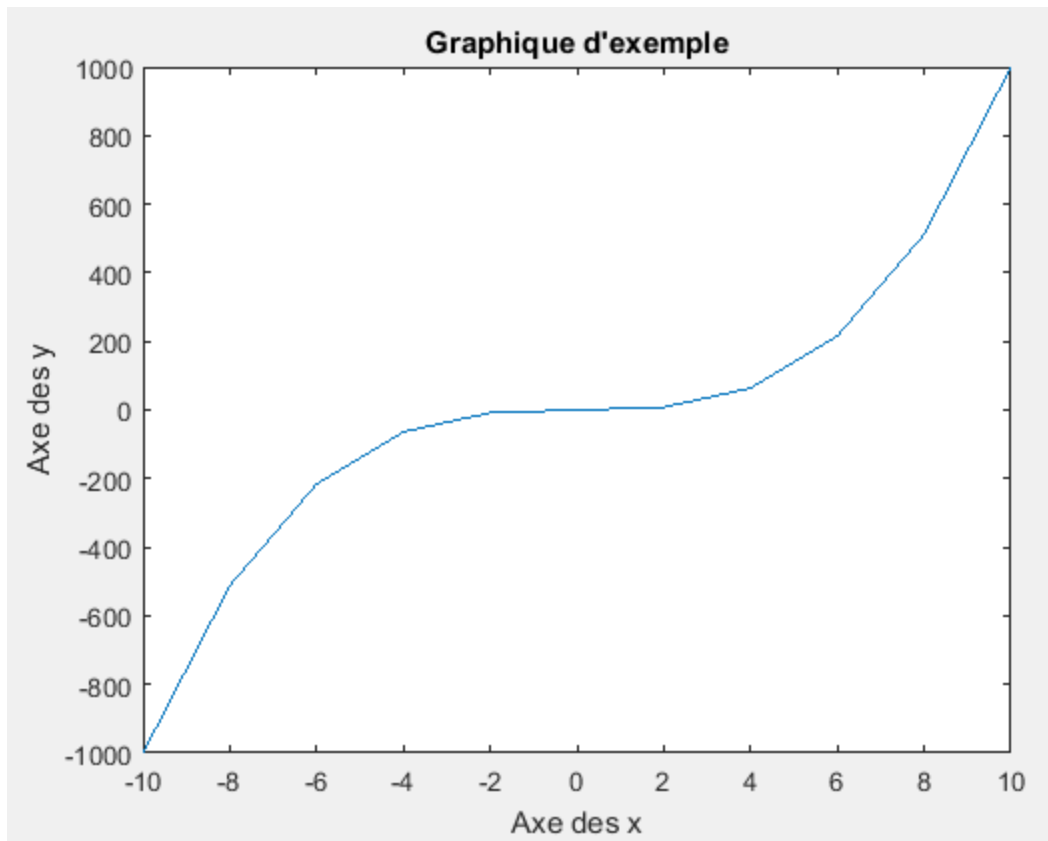
Création de graphiques de bases

Voyons comment afficher les données dans MATLAB. La fonction de base pour dessiner un graphe bidimensionnel dans MATLAB est la fonction «plot». Il existe une grande variété de syntaxe pour cette fonction, vous permettant de modifier de nombreux paramètres au fur et à mesure que vous la créez. Le graphique le plus simple prend un seul vecteur de données et trace les valeurs des données par rapport au nombre d'index des données dans le vecteur: `plot(x, y)`

Ajout d'annotations

Pour ajouter des informations sur un graphique comme le titre, les axes, ... il suffit de taper les fonctions `xlabel`, `ylabel` et `title` suivies des informations à saisir.

```
x = -10:2:10;  
y = (x.^3);  
plot(x,y)  
xlabel('Axe des x');  
ylabel('Axe des y');  
title('Graphique d''exemple');
```



Annexe III: Introduction aux matrices

Une matrice est une disposition rectangulaire de nombres en lignes et en colonnes. Par exemple, la matrice A a deux lignes et trois colonnes.

$$A = \begin{bmatrix} -2 & 5 & 6 \\ 5 & 2 & 7 \end{bmatrix}$$

3 colonnes
↓ ↓ ↓

← ← 2 lignes

Dimensions des matrices

Les dimensions d'une matrice indiquent sa taille: le nombre de lignes et de colonnes de la matrice, dans cet ordre.

Puisque la matrice A a deux lignes et trois colonnes, nous écrivons sa dimension comme 2x3. En revanche, la matrice B comporte 3 lignes et 2 colonnes, c'est donc une matrice 3x2.

$$B = \begin{bmatrix} -8 & -4 \\ 23 & 12 \\ 18 & 10 \end{bmatrix}$$

Éléments d'une matrice

Un élément matriciel est simplement une entrée matricielle. Chaque élément d'une matrice est identifié en nommant la ligne et la colonne dans laquelle elle apparaît.

Par exemple, considérons la matrice G:

$$G = \begin{bmatrix} 4 & 14 & -7 \\ 18 & 5 & 13 \\ -20 & 4 & 22 \end{bmatrix}$$

L'élément G(3,1) est l'entrée dans la troisième ligne et la première colonne qui est -20

Opérations des matrices

- Addition

Pour additionner deux matrices de même dimension, additionnez simplement les entrées dans les positions correspondantes.

Rappelez-vous, les matrices doivent avoir **la même dimension**, ce qui signifie le même nombre de lignes et de colonnes.

Exemple:

$$\begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 2 \\ 8 & 1 \end{bmatrix} = \begin{bmatrix} 3+5 & 7+2 \\ 2+8 & 4+1 \end{bmatrix} \\ = \begin{bmatrix} 8 & 9 \\ 10 & 5 \end{bmatrix}$$

• Multiplication

a) Scalaire par matrice

Le terme de multiplication scalaire se réfère au produit d'un nombre réel et d'une matrice. Dans la multiplication scalaire, chaque entrée dans la matrice est multipliée par le scalaire donné.

$$2 \cdot \begin{bmatrix} 5 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 2 \cdot 5 & 2 \cdot 2 \\ 2 \cdot 3 & 2 \cdot 1 \end{bmatrix} \\ = \begin{bmatrix} 10 & 4 \\ 6 & 2 \end{bmatrix}$$

b) Produit de deux matrices

La multiplication matricielle se réfère au produit de deux matrices. C'est une opération totalement différente. Vous pouvez multiplier deux matrices si et seulement si, le nombre de colonnes dans la première matrice est égal au nombre de lignes dans la deuxième matrice.

Sinon, le produit de deux matrices est indéfini.

Les dimensions de la matrice du produit sont (lignes de première matrice) \times (colonnes de la deuxième matrice)

Par exemple, si nous multiplions une matrice 2×3 par une matrice 3×1 , la matrice produit est 2×1

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \times \begin{bmatrix} B_{11} \\ B_{21} \\ B_{31} \end{bmatrix} = \begin{bmatrix} A_{11} \times B_{11} + A_{12} \times B_{21} + A_{13} \times B_{31} \\ A_{21} \times B_{11} + A_{22} \times B_{21} + A_{23} \times B_{31} \end{bmatrix} = \begin{bmatrix} C_{11} \\ C_{12} \end{bmatrix}$$