

Matlab : applications en mécanique.

2A103, TP1

Sorbonne Université.

Licence de mécanique.

1) Mise en place

Vous êtes sous Linux. Créez un répertoire 2A103 puis un sous-répertoire (ex : 2A103/TP1) Lorsque vous utiliserez Matlab, sauvegardez votre script (TP1.m) dans ce sous-répertoire.

2) Prise en main : commandes

Tapez ces commandes dans la Command Window et observez ce qui se passe pour vous construire une compréhension intuitive du fonctionnement de Matlab. Les phrases en bleu décrivent les commandes. Pour plus d'information sur des fonctions vous pouvez utiliser la fonction `help` de Matlab. Les réponses des QCM servent de correction pour vous préparer au C.C de la séance 4.

Calculatrice

```
>> 5
>> 2 + 5
>> a = 10
>> a
```

La variable `a` est créée, et reçoit le scalaire 10

1/ Que fait cette commande `>> disp(a)` ?

Affiche la valeur de `a`
Affiche la taille de `a`
Affiche le type de `a`

```
>> a + 5
>> b = a + 5
>> b = a + 5;
>> c = [1,-2,7,0,10]
>> 2*c
>> d = [1;2;3;4;5]
>> size(a)
>> size(c)
>> whos
```

La valeur contenue dans la variable `a` est additionnée à 5 et le résultat est affiché à l'écran

2/ Quel est le résultat de `>> c(3)` ?

0
7
Erreur

```
>> d(1) + d(2) + d(3) + d(4) + d(5)
>> c(1) = 0
```

3/ Que vaut `c` à présent ?

```
c = 0
c = [0,-2,7,0,10]
c = [1,0,7,0,10]
```

```
>> e = [0, 0; 0, 0]
>> pi
>> cos(2*pi)
>> 2^3
```

Le symbole `^` permet de calculer une puissance

4/ Expliquer ce que fait chacune de ces commandes:

```
>> c*d
>> d*c
>> c+d
```

Messages d'erreur

```
>> toto
>> c(6)
```

Il n'existe pas de variable nommée `toto`, donc je ne peux pas afficher sa valeur à l'écran

5/ Pourquoi y-a-t'il une erreur dans la commande `>> c*d'` ?

`c` est un vecteur colonne `d` est un vecteur ligne
`c` et `d` n'ont pas la même taille
`c` est un vecteur ligne `d` est un vecteur colonne

```
>> f = [0, 0 ; 1]
>> g = 2*(1,2 + 1)
```

"Pour la variable `ind` allant de 1 à 10 par pas de 1, on affiche à l'écran la valeur de cette variable"

Boucles et tests

```
>> for ind = 1:10; disp(ind); end
>> a = 0; for ind = 1:10; a = a + ind; disp(a); end
>> a = 1; for ind = 1:2:10; a = a*ind; disp(a); end
```

6/ Quelle est la dernière valeur affichée ?

- 105
- 3628800
- 945

Tableaux

```
>> a = [0,0; 0,0]
>> a(2,2) = 1
>> a(1,1) = a(2,2)
```

"On construit la matrice `a` avec deux colonnes et deux lignes remplies de zéros"

7/ Que vaut `a` à présent ?

```
[0,0;0,1]
[0,1;0,1]
[1,0;0,1]
```

```
>> b = [a,a]
>> c = [0*a, a, 2*a, 3*a]
```

8/ Quelle est la taille de `c` ?

Tableau de 2 lignes 8 colonnes
Tableau de 6 lignes 2 colonnes
Tableau de 8 lignes 8 colonnes

Fonctions mathématiques

```
>> sin(2.5)^2 + cos(2.5)^2
```

9/ Quel est le résultat de la commande `>> exp(log(3))` ?

```
20.0855
1.0986
3
```

```
>> sqrt(-1)
```

```
>> sin = 2:8;
```

10/ Quel est le résultat de la commande `>>> sin(3)` ?

```
0.1411
4
Erreur
```

```
>> clear sin; sin(3)
```

Fonctions pour tableaux

```
>> x = linspace(0,2*pi,10)
```

```
>> y = 0:0.5:10
```

11/ Quelle est la taille de `y` ?

```
10
20
21
```

On crée le vecteur `x` de taille une ligne et 10 colonnes, qui contient 10 valeurs équiréparties entre 0 et 2π

```
>> max(x)
```

```
>> min(x)
```

```
>> sum(x)
```

12/ Que fait la fonction `prod(x)` ?

Le produit de `x` par 1

La somme des éléments de `x`

Le produit des éléments de `x`

On trace une ligne continue de couleur noire entre les points de coordonnées (0,1) et (2,0), on visualise chacun des deux points avec une astérisque

```
>> sin(x)
```

```
>> x = logspace(-2,2,5)
```

Graphiques

```
>> plot([0,2],[1,0], 'k*-')
```

```
>> xlabel('x'); ylabel('y'), title('Un segment')
```

```
>> xlim([-0.5,3]); ylim([-1,2])
```

```
>> x = linspace(0,4*pi,30); plot(x,sin(x), 'r*--')
```

13/ Quelle est la taille de `x` ?

```
4
10
30
```

```
>> subplot(2,1,1); plot([0 2 1 0 0],[0 0 1.5 1 0])
```

14/ Combien y-a t'il de figures avec cette commande `subplot` ?

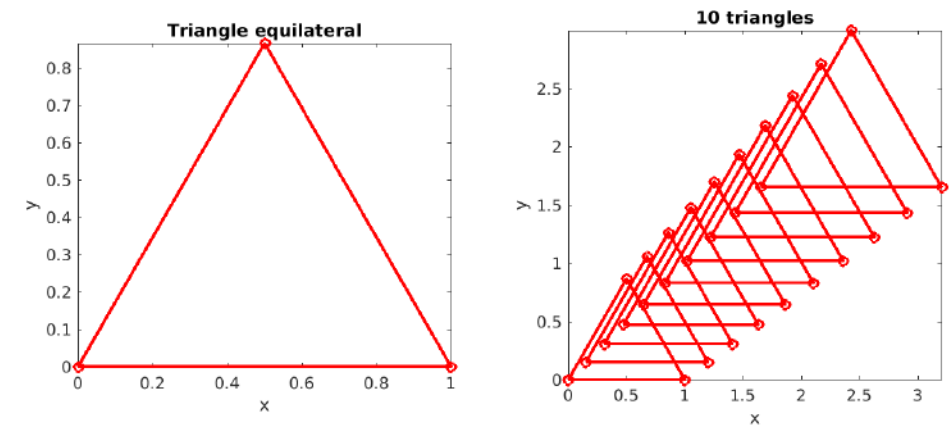
```
2
3
4
```

```
>> axis([-1 2.5 -2 3])
```

```
>> subplot(2,1,2); plot([0.1 0.5 0.9 0.5 0.1],[0.5 0 0.5 1 0.5])
```

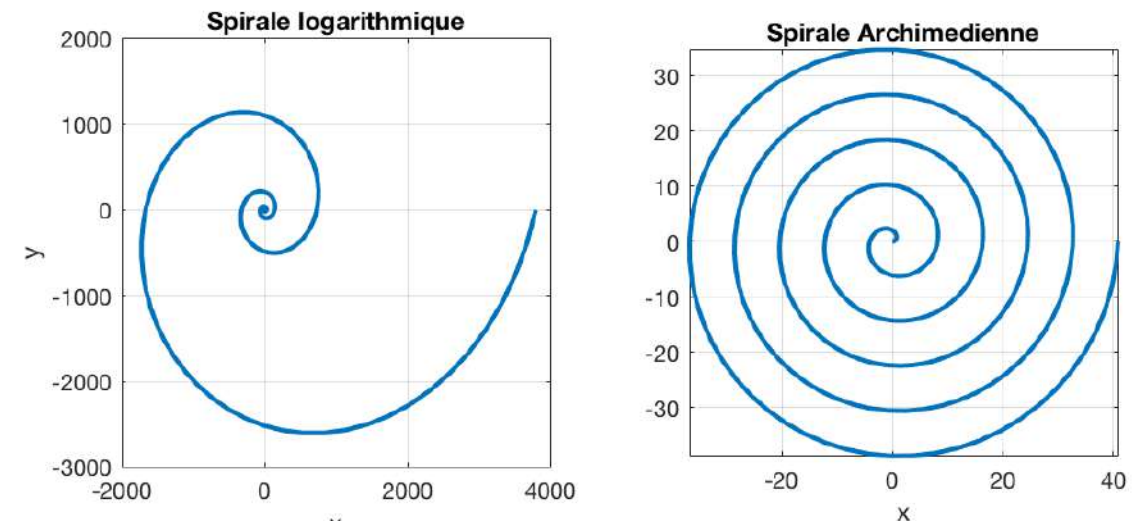
3) Prise en main : graphique

Reproduisez la figure : tracez un triangle équilatéral en choisissant les coordonnées des points à donner à la fonction `plot`. Ensuite, tracez 10 triangles en changeant de position et de taille avec l'aide d'une boucle `for`. Les deux graphiques sont mis dans une seule fenêtre graphique avec la fonction `subplot`.



4) Courbes paramétrées

Reproduisez la figure avec une spirale algébrique (Archimédienne) et une spirale logarithmique, dans deux sous graphiques l'un à côté de l'autre.



On peut définir des spirales avec $x = r \cos(\theta)$ et $y = r \sin(\theta)$, mais ici le rayon r dépend de l'angle θ . Pour la spirale logarithmique, on a la loi $r = a^\theta$, (a puissance θ) et pour la spirale Archimédienne, on a $r = a\theta$, où a est un paramètre fixe, par exemple 1.3.

θ varie de 0 à 2π pour un tour de spirale, et par exemple de 0 à 20π pour dix tours de spirale.