

1) Mise en place

Vous êtes sous linux. Créez dans votre répertoire d'utilisateur le répertoire matlab1. Créez un document Openoffice Impress (l'équivalent opensource de Microsoft Powerpoint) en format paysage («landscape» en anglais). Mettez un titre, le nom des deux étudiants du binôme, le nom de votre encadrant de TP.

Calculatrice

```
>> 5
>> 2+5
>> a=10 ← "la variable a est créée, et reçoit le scalaire
>> a      10, puis la valeur de a est affichée à
>> disp(a) l'écran"
>> a+5 ← "la valeur contenue dans la variable a est additionnée à
>> b=a+5; 5 et le résultat est affiché à l'écran"
>> b=a+5;
>> c=[1,-2,7,0,10]
>> 2*c
>> d=[1;2;3;4;5]
>> size(a)
● >> size(c)
>> whos
● >> c(3)
>> d(1)+d(2)+d(3)+d(4)+d(5)
>> c(1)=0
>> e=[0, 0; 0, 0]
>> pi
● >> cos(2*pi)
>> 2^3
```

Messages d'erreur

```
>> toto ← "il n'existe pas de variable nommée toto, donc je ne peux pas
>> c(6)  afficher sa valeur à l'écran"
● >> c+d
● >> f=[0, 0 ; 0]
```

Boucles et tests

```
>> for ind=1:10; disp(ind); end ← "pour la variable ind allant de 1 à 10 par
>> a=0; for ind=1:10; a=a+ind; disp(a); end pas de 1, on affiche à l'écran la
>> a=1; for ind=1:2:10; a=a*ind; disp(a); end valeur de cette variable".
```

Tableaux

```
>> a=[0,0; 0,0] ← "On construit la matrice a avec deux colonnes et deux
>> a(2,2)=1 lignes remplies de zéros"
● >> a(1,1)=a(2,2)
● >> b=[a,a]
>> c=[0*a, a, 2*a, 3*a]
```

Fonctions mathématiques

```
>> sin(2.5)^2+cos(2.5)^2
● >> exp(log(3))
>> sqrt(-1)
```

Fonctions pour tableaux

```
>> x=linspace(0,2*pi,10) ← "On crée le vecteur x de taille une ligne et 10
>> y=0:0.5:10 colonnes, qui contient 10 valeurs équiréparties
>> max(x) entre 0 et 2pi.
>> min(x)
>> sum(x)
```

```
● >> prod(x)
>> sin(x)
>> x=logspace(-2,2,5)(0)
```

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

Graphiques

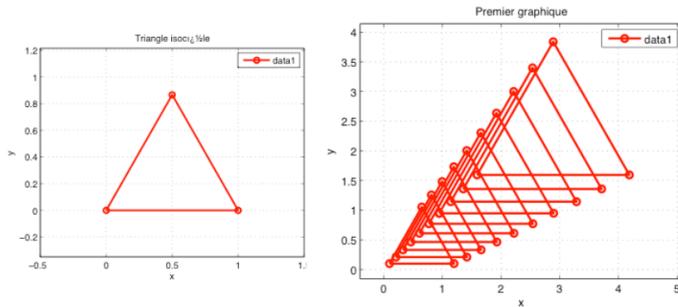
```
>> plot([0,1],[1,0],'k*--') ← "On trace une ligne continue de couleur noire entre les points
>> xlabel('x'); ylabel('y'), title('joli graph') de coordonnées (0,1) et (1,0), on visualise chacun des deux
>> xlim([-1,2]); ylim([-1,2]) points avec une astérisque"
● >> x=linspace(0,4*pi,30); plot(x,sin(x),'r*--')
```

2) Prise en main

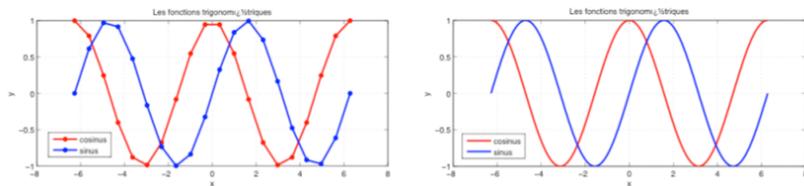
Tapez ces commandes à l'invite et observez ce qu'il se passe pour vous construire une compréhension intuitive du fonctionnement. Pour les commandes notés ● inspirez vous des exemples pour rédiger une phrase qui décrit ce que font les commandes. Ces phrases sont rédigées dans votre compte rendu.

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 20 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 (voir notes de cours et aide (fonction help)



Reproduisez la figure: tracez un cosinus avec 20 points en abscisse (puisque'il y a peu de points, on voit les segments de droites). Dans un second sous graphique, tracez avec beaucoup de points de abscisse. Pour créer les points d'abscisses, utilisez la fonction linspace.



5) Compte-rendu

Sauvez votre compte-rendu au format pdf.

Connectez vous sur internet à l'adresse australe.upmc.fr. Vérifiez que vous avez un onglet LA207. Si vous ne l'avez pas, dites-le à votre encadrant, il va vous inscrire.

Dans «devoirs évalués», remettez votre compte-rendu, un par étudiant (et non pas un seul par binôme).

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 $r = a\theta$, ou a est un paramètre fixe, par exemple 1.3.

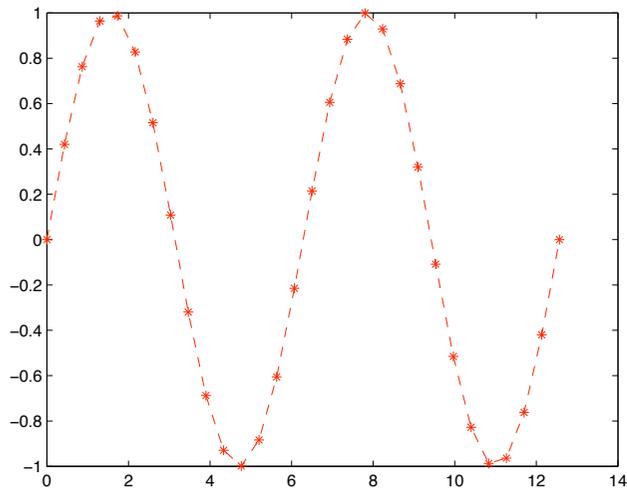
θ varie de 0 à 2π pour un tour de spirale, et par exemple de 0 à 20π pour dix tours de spirale.

6) Animations Pour aller plus loin

Nous pouvons animer nos graphiques avec une boucle for et la fonction drawnow: par exemple je trace $\sin(x+p)$ ou p est une valeur que je fais augmenter progressivement dans une boucle for. A chaque itération de la boucle for, j'utilise drawnow pour que le graphique soit immédiatement affiché à l'écran.

Pour les spirales, vous pouvez les faire tourner en définissant de la même manière $x=r \cos(\theta+b)$, $y=r \sin(\theta+b)$.

La courbe de la question 8



Rédactions:

Q2)

(3): On fait appel à la fonction "size", qui affiche à l'écran le nombre de lignes et le nombre de colonnes du tableau c

(4): On affiche à l'écran la valeur numérique mémorisée dans la troisième case du tableau c. (S'il y a une seule ligne ou bien une seule colonne, on n'a pas besoin de mettre les deux indices c(3,1))

(5) On affiche à l'écran la valeur du cosinus de 2 pi, qui est égale à 1.

Q3)

(1): `??? Error using ==> plus, Matrix dimensions must agree.`

On a essayé d'additionner deux tableaux qui n'ont pas la même taille.

(2) `??? Error using ==> vertcat, CAT arguments dimensions are not consistent.`

On a essayé de construire un tableau qui n'a pas le même nombre de colonne pour chaque ligne, ce qui est impossible puisqu'un tableau doit être rectangulaire. Vertcat signifie "concaténation verticale".

Q4)

(1) On initialise la variable "a" avec la valeur 2, puis on fait une boucle de 1 à 10 par pas de 2. A chaque itération, on multiplie "a" par l'indice "ind" de la boucle et on affiche la valeur de "a" à l'écran.

Q5)

(1) On remplace la valeur mémorisée dans la case de la première ligne première colonne par la valeur mémorisée dans la case de la seconde ligne seconde colonne.

(2) On construit le tableau "b" en concaténant "a" côte à côte avec lui-même.

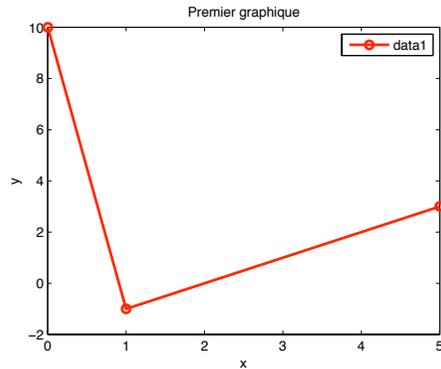
Q7)

(1) On construit un tableau à une ligne d'éléments linéairement répartis entre 0 et 10, avec un pas de 0.5 (c'est très similaire à la fonction linspace, sauf que ici on donne le pas, plutôt que le nombre d'éléments)

(2) On affiche à l'écran le produit de toutes les cases du tableau "x".

Q8)

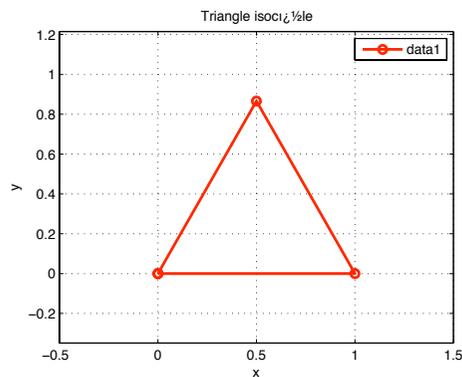
On construit un tableau de 30 éléments linéairement répartis entre 0 et 4 pi, puis on trace la courbe des points qui ont les valeurs des cases de x en abscisse et les valeurs des cases de sin(x) en ordonnée. La courbe est tracée en rouge avec une ligne hachurée, et des marqueurs en astérisque.



```
% definition des abscisses et des ordonnees
x=[0,1,5];
y=[10,-1,3];

% on trace
plot(x,y,'r-o','linewidth',2);
xlabel('x'); ylabel('y'); title('Premier graphique');
legend('data1')
```

Script

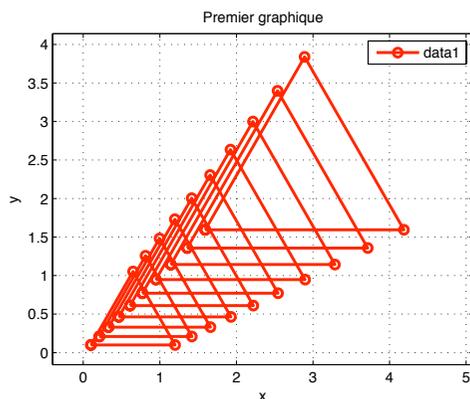


```
% definition des abscisses et des ordonnées des points
x=[0 1 0.5 0];
y=[0 0 sin(pi/3) 0];

% on trace
plot(x,y,'r-o','linewidth',2);
xlim([-0.5,1.5]); ylim([-0.5,1]);
xlabel('x'); ylabel('y'); title('Triangle isoc?le'); legend('data1')
axis equal
grid on
```

Script

Il faut bien penser à remettre le premier point à la fin du tableau de sorte à “boucler la boucle”



```
% plein de triangles isoc?les
hold off

% definition des points du triangle
x=[0 1 0.5 0];
y=[0 0 sin(pi/3) 0];

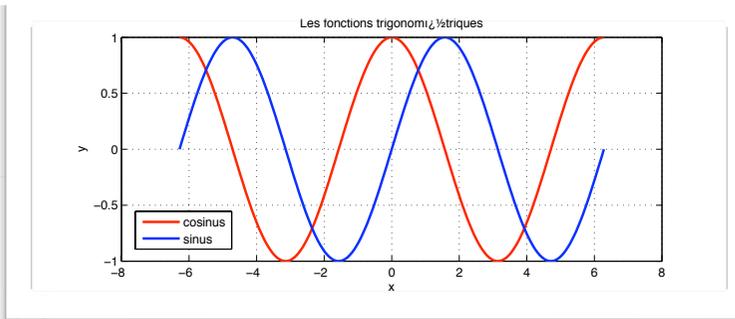
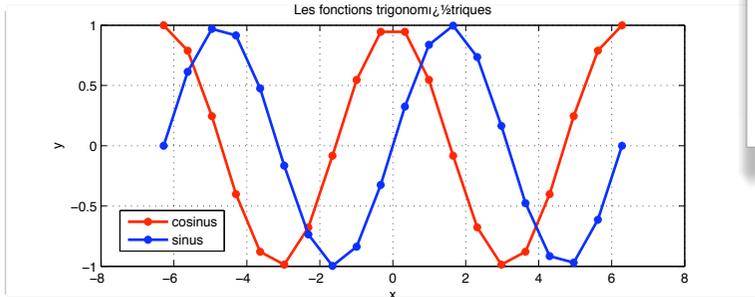
% on fait une boucle
for ind=1:10

    % on translate le triangle, et on l'agrandit
    x=1.1*x+0.1;
    y=1.1*y+0.1;

    % on trace le triangle
    plot(x,y,'r-o','linewidth',2);
    xlabel('x'); ylabel('y'); title('Premier graphique'); legend('data1')
    axis equal
    hold on
end
grid on
```

Script

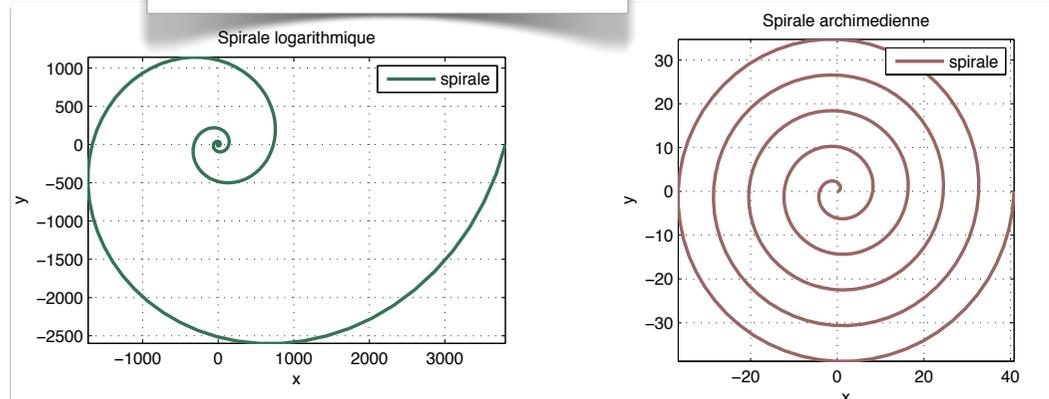
A chaque itération, on déplace le triangle en ajoutant à x et à y une valeur constante, et on l'agrandit en multipliant les tableaux des x et des y par une constante.



```
% definition des abscisses et des ordonnées
x=linspace(-2*pi,2*pi,20);
y1=cos(x);
y2=sin(x);

% on trace
plot(x,y1,'r*-',x,y2,'b*-')
legend('cosinus','sinus');
xlabel('x'); ylabel('y'); title('Les fonctions trigonométriques');
grid on
```

Script



```
%%% la spirale logarithmique
subplot(1,2,1)
th=linspace(0,2*pi*5,300);

% les abscisses et ordonnées
r=1.3.^th;
x1=r.*cos(th);
y1=r.*sin(th);

% on trace
plot(x1,y1,'k-')
axis equal tight
legend('spirale');
xlabel('x'); ylabel('y'); title('Spirale logarithmique'); grid on

%%% la spirale Archimédienne
subplot(1,2,2)
th=linspace(0,2*pi*5,300);

% les abscisses et ordonnées
r=1.3*th;
x2=r.*cos(th);
y2=r.*sin(th);

% on trace
plot(x2,y2,'k-')
axis equal tight
legend('spirale');
xlabel('x'); ylabel('y'); title('Spirale archimédienne'); grid on
```

Script

Ici il faut utiliser le symbole `.*` pour la multiplication de tableau, puisque `r` est un tableau et `cos(th)` aussi: on multiplie chaque case du tableau `r` par la case correspondante du tableau `cos(th)`