Matlab : applications en mécanique. LA207

Université Pierre et Marie Curie.
Licence de mécanique.
Examen final, juin 2012.
Sujet du matin.

Ex1 Compétences générales

1) Tracez la courbe paramétrée («coeur de Raphaël Laporte», cf mathcurve.com) pour le paramètre t variant de 0 à 2π :

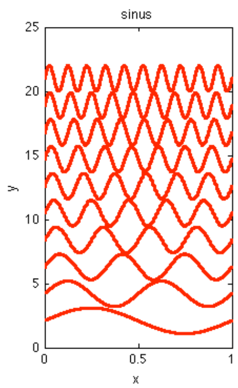
$$\begin{cases} x = \sin^3 t \\ y = \cos t - \cos^4 t \end{cases}$$

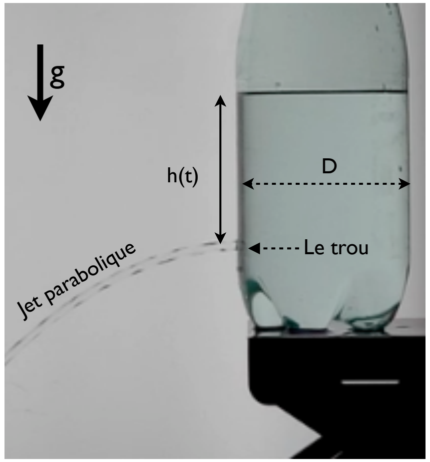
2) Ecrivez un code Matlab qui calcule la somme suivante (expansion de Taylor du cosinus en 0) pour une valeur donnée a=3π/4, et N=10 :

$$S_N(a) = \sum_{k=0}^{N} (-1)^k \frac{a^{2k}}{(2k)!}$$

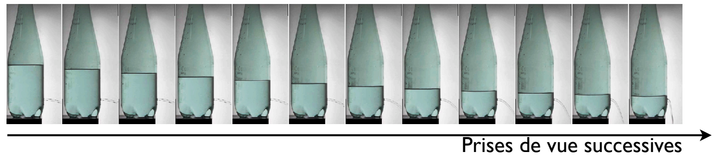
Tracez un graphique qui montre que cette somme tend vers -√2/2 lorsque N devient grand (la valeur de cos(3π/4)).

3) Ecrivez un code matlab qui reproduit qualitativement la figure ci dessous (une fonction sinus dont on fait varier la période et la hauteur) :





Ex2 Toricelli



L'image toricelli.tif représente la vidange d'une bouteille de Badoit qui a été percée d'un petit trou de diamètre d. C'est l'expérience de Toricelli. Avec l'équation de Bernoulli, on montre que la vitesse du jet à un temps donné est égale à √2gh, avec h la hauteur entre le trou et la surface de l'eau, et g=9.81m/s2. En prenant en compte la conservation du débit, on montre que la hauteur h évolue dans le temps comme ceci

$$h(t) = \left(\sqrt{h_0} - \frac{d^2}{D^2}\sqrt{\frac{g}{2}}t\right)^2$$

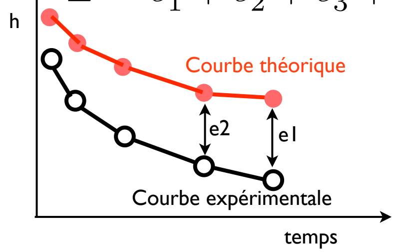
Ou h0 est la hauteur initiale, d le diamètre du petit trou, et D le diamètre de la bouteille. Nous allons étudier ce phénomène en comparant l'expérience à cette théorie.

1) Lisez l'image toricelli.tif et affichez là dans une fenêtre graphique.
2) Calculez la taille d'un pixel de l'image en prenant pour étalon de longueur le diamètre de la bouteille D=8 centimètres.
3) Le vecteur temps: Le film est pris à 30 prises de vue par seconde, et le numéro de chaque prise de vue est affiché sur l'image. Construisez le tableau tvec qui contient les valeurs successives des temps de notre expérience.
4) Avec la fonction ginput, mesurez les valeurs successives de h(t) en mètres: la hauteur entre le centre du petit trou et la surface de l'eau. Vous utiliserez pour la transformation de pixel en mètre la taille d'un pixel que vous avez calculée à la question 2.
5) Tracez le graphique de h (en mètres) en fonction de t (en secondes). Annotez votre graphique: labels et titre. Voici notre courbe expérimentale.

Nous allons maintenant comparer ce graphique expérimental avec la formule théorique.

6) Superposez la courbe expérimentale avec la formule théorique pour d le diamètre du petit trou égal à 3 millimètres. Cette valeur du diamètre est elle trop grande ou trop petite? Justifiez votre réponse en décrivant le graphique.
7) Superposez la courbe expérimentale avec la formule théorique pour 20 valeurs successives de d entre 3 et 6mm. En déduire une valeur estimée du diamètre d.
8) Maintenant une dernière manière pour estimer d: pour chaque valeur successive de d entre 3 et 6mm, calculez l'erreur E au sens des moindres carrés entre la courbe expérimentale et la courbe théorique. Cette erreur est la somme des carrés de l'erreur entre la valeur expérimentale et la valeur théorique. Tracez la courbe de E en fonction de d et déduisez en le diamètre du trou de notre bouteille de Badoit.

$$E = e_1^2 + e_2^2 + e_3^2 + \dots$$

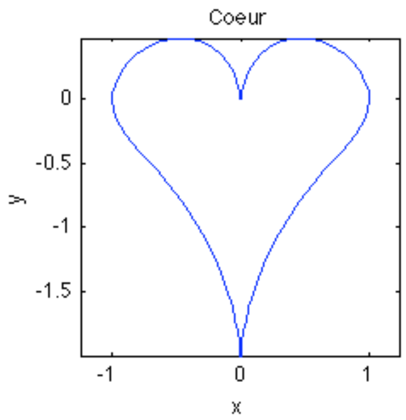


Ex I Compétences générales

1) Tracez la courbe paramétrée («coeur de Raphaël Laporte», cf mathcurve.com) pour le paramètre t variant de 0 à 2π:

$$\begin{cases} x = \sin^3 t \\ y = \cos t - \cos^4 t \end{cases}$$

```
t=linspace(0,2*pi,100);
x=sin(t).^3; y=cos(t)-cos(t).^4;
plot(x,y)

axis equal
xlabel('x'); ylabel('y'); title('sinus')
```

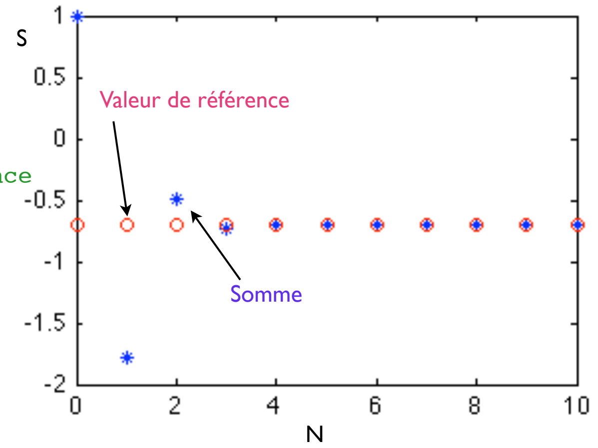


2) Ecrivez un code Matlab qui calcule la somme suivante (expansion de Taylor du cosinus en 0) pour une valeur donnée a=3π/4:

$$S_N(a) = \sum_{k=0}^{N} (-1)^k \frac{a^{2k}}{(2k)!}$$

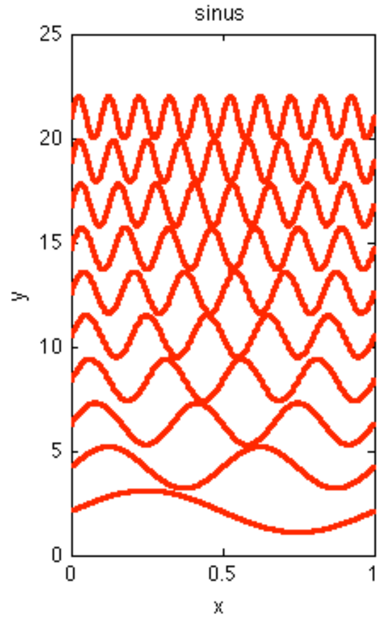
Tracez un graphique qui montre que cette somme tend vers -√2/2 lorsque N devient grand (la valeur de cos(3π/4)).

```
s=0; % on initialise la somme
N=10; % le nombre d'itérations
a=3*pi/4;
for k=0:N % la boucle
s=s+(-1)^k*a^(2*k)/(factorial(2*k))
% on trace la somme et la valeur de référence
plot(k,s,'*',k,-sqrt(2)/2,'ro'); hold on
end
```



3) Ecrivez un code matlab qui reproduit qualitativement la figure ci dessous (une fonction sinus dont on fait varier la période et la valeur moyenne):

```
x=linspace(0,1,1000); % les valeurs des x
for n=1:10 % une boucle de 1 à 10
y=2.1*n+sin(n*x*(2*pi)); % le sinus
plot(x,y,'r','linewidth',2); % on trace le graphique
hold on; % pour superposer les graphiques
end
xlabel('x'); ylabel('y'); title('sinus')
```



Ex2 Toricelli

L'image toricelli.tif représente la vidange d'une bouteille de Badoit qui a été percée d'un petit trou de diamètre d. C'est l'expérience de Toricelli. Avec l'équation de Bernoulli, on montre que la vitesse du jet à un temps donné est égale à $\sqrt{2gh}$, avec h la hauteur entre le trou et la surface de l'eau, et g=9.81m/s2. En prenant en compte la conservation du débit, on montre que la hauteur h évolue dans le temps comme ceci

$$h(t) = \left(\sqrt{h_0} - \frac{d^2}{D^2}\sqrt{\frac{g}{2}}t\right)^2$$

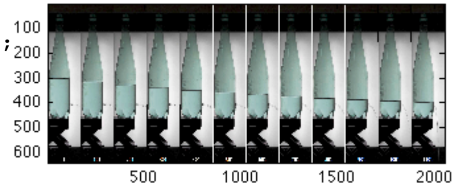
Ou h0 est la hauteur initiale, d le diamètre du petit trou, et D le diamètre de la bouteille. Nous allons étudier ce phénomène en comparant l'expérience à cette théorie.

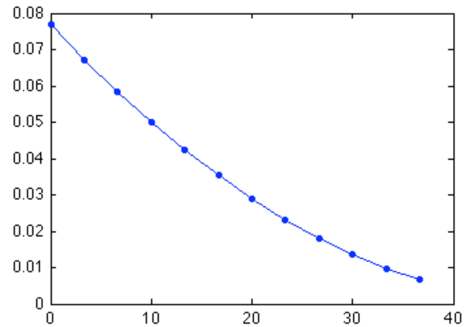
1) Lisez l'image toricelli.tif et affichez là dans une fenêtre graphique.
2) Calculez la taille d'un pixel de l'image en prenant pour étalon de longueur le diamètre de la bouteille D=8 centimètres.
3) Le vecteur temps: Le film est pris à 30 prises de vue par seconde, et le numéro de chaque prise de vue est affiché sur l'image. Construisez le tableau tvec qui contient les valeurs successives des temps de notre expérience.
4) Avec la fonction ginput, mesurez les valeurs successives de h(t) en mètres: la hauteur entre le centre du petit trou et la surface de l'eau. Vous utiliserez pour la transformation de pixel en mètre la taille d'un pixel que vous avez calculée à la question 2.
5) Tracez le graphique de h (en mètres) en fonction de t (en secondes). Annotez votre graphique: labels et titre. Voici notre courbe expérimentale.

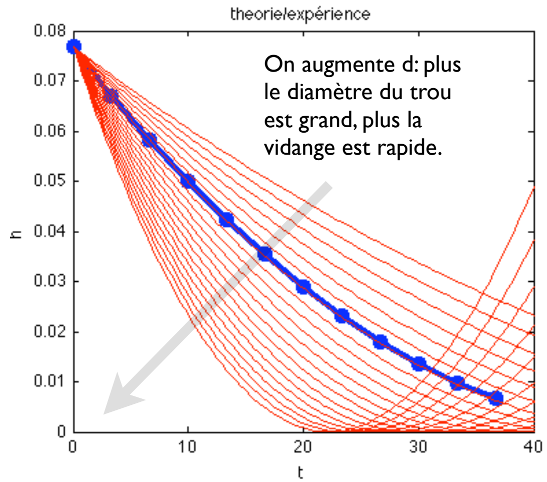
Nous allons maintenant comparer ce graphique expérimental avec la formule théorique.

6) Superposez la courbe expérimentale avec la formule théorique pour d le diamètre du petit trou égal à 3 millimètres. Cette valeur du diamètre est elle trop grande ou trop petite? Justifiez votre réponse en décrivant le graphique.
7) Superposez la courbe expérimentale avec la formule théorique pour 20 valeurs successives de d entre 3 et 6mm. En déduire une valeur estimée du diamètre d.
8) Maintenant une dernière manière pour estimer d: pour chaque valeur successive de d entre 3 et 6mm, calculez l'erreur E au sens des moindres carrés entre la courbe expérimentale et la courbe théorique. Cette erreur est la somme des carrés de l'erreur entre la valeur expérimentale et la valeur théorique. Tracez la courbe de E en fonction de d et déduisez en le diamètre du trou de notre bouteille de Badoit.

```
% on lit et affiche l'image
a=imread('films_matlab/toricelli.tif');
subplot(1,2,1)
image(a);

% la taille des pixels en mètres
taillepix=0.08/(801-692);

% les points mesurés sur l'image
d=1000*[ 0.0626    0.3024
    0.2245     0.3157
    0.3864     0.3275
    0.5592     0.3388
    0.7409     0.3493
    0.9100     0.3588
    1.0882     0.3676
    1.2519     0.3755
    1.4228     0.3827
    1.5974     0.3885
    1.7683     0.3938
    1.9356     0.3980];

% transformation de référentiel
h=d(:,2);
h=-taillepix*(h-407);


subplot(1,2,2);
% le vecteur du temps
t=((1:100:1101)-1)/30;
% on trace les données expérimentales
plot(t,h,'b.-');
```

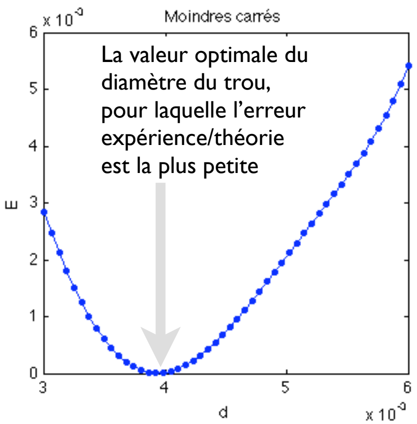







```
% boucle pour faire varier le diamètre
for d=linspace(0.003,0.006,20)
h0=h(1);
tt=linspace(0,40,100);
htheo=(sqrt(h0)-(d^2/0.08^2)*sqrt(9.81/2)*tt).^2;

plot(tt,htheo,'r')
end
xlabel('t'); ylabel('h'); title('theorie/expérience')

break
```

```
% calcul de l'erreur moindres carrés
dvec=linspace(0.003,0.006,50); % les valeurs de d
Evec=zeros(50,1); % un tableau pour mémoriser l'erreur

% la boucle
for ind=1:length(dvec)
h0=h(1); % la valeur initiale de h
d=dvec(ind); % la valeur actuelle du diamètre

% la formule théorique
htheo=(sqrt(h0)-(d^2/0.08^2)*sqrt(9.81/2)*t).^2;

% calcul de l'erreur aux moindres carrés
Evec(ind)=sum((htheo-h').^2);
end

% on trace le graphique
plot(dvec,Evec,'b.-');
xlabel('d'); ylabel('E'); title('Moindres carrés')
```



Matlab : applications en mécanique.
LA207

Université Pierre et Marie Curie.
Licence de mécanique.
Examen final, juin 2012.
Sujet de l'après-midi.

Ex1 Compétences générales

1) Tracez la courbe paramétrée en coordonnées polaires («papillon de T. Fay», cf mathcurve.com) pour le paramètre θ variant de 0 à 2π :

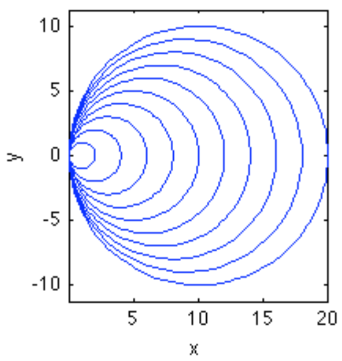
$$r = e^{\cos(\theta)} - 2\cos(4\theta)$$

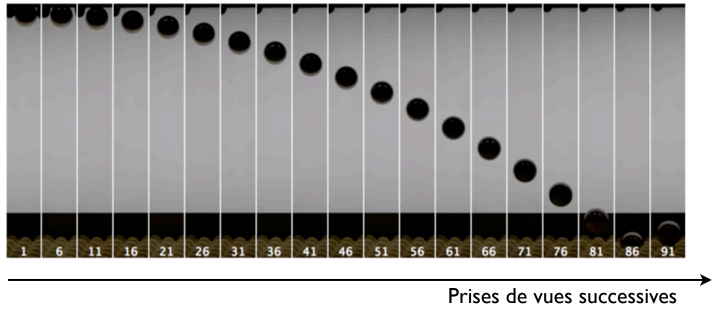
2) Ecrivez un code Matlab qui calcule la somme suivante (expansion de Taylor de l'exponentielle en 0) pour une valeur donnée a=2 :

$$S_N(a) = \sum_{k=0}^{N} \frac{a^k}{k!}$$

Tracez un graphique qui montre que cette somme tend vers exp(2) lorsque N devient grand.

3) Ecrivez un code matlab qui reproduit qualitativement la figure ci dessous (un cercle dont on fait varier le rayon et le centre):





Ex2 Chute libre

L'image chutelibre.tif représente la chute d'une boule sans vitesse initiale. Cet objet est soumis à la seule force de son poids, puisque aux vitesses qu'elle atteint et pour sa masse, on peux négliger la contribution des forces de frottement aérodynamiques. La boule chute donc à accélération constante, ce qui fait que la distance x au point de lâcher évolue comme :

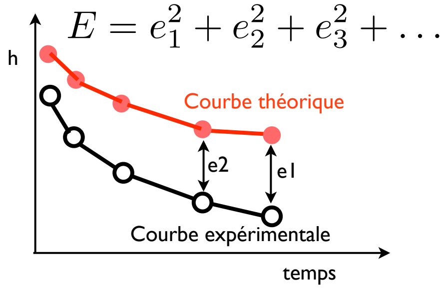
$$x = gt^2/2$$

Ou g est l'accélération de la gravité dont nous allons estimer la valeur grâce à cette expérience, et t est le temps compté depuis le lâcher de la boule (l'image initiale).

1) Lisez l'image chutelibre.tif et affichez là dans une fenêtre graphique.
2) Calculez la taille d'un pixel de l'image en prenant pour étalon de longueur la hauteur de fenêtre lumineuse dans le fond de l'image H=35.5cm.
3) Le vecteur temps: Le film est pris à 300 prises de vue par seconde, et le numéro de chaque prise de vue est affiché sur l'image. Construisez le tableau tvec qui contient les valeurs successives des temps de notre expérience.
4) Avec la fonction ginput, mesurez les valeurs successives de x(t) en mètres: la hauteur entre le point le plus bas de la boule à t=0, et le point le plus bas de la boule aux temps t successifs. Vous utiliserez pour la transformation de pixel en mètre la taille d'un pixel que vous avez calculée à la question 2.
5) Tracez le graphique de x (en mètres) en fonction de t (en secondes). Annotez votre graphique : labels et titre. Voici notre courbe expérimentale.

Nous allons maintenant comparer ce graphique expérimental avec la formule théorique.

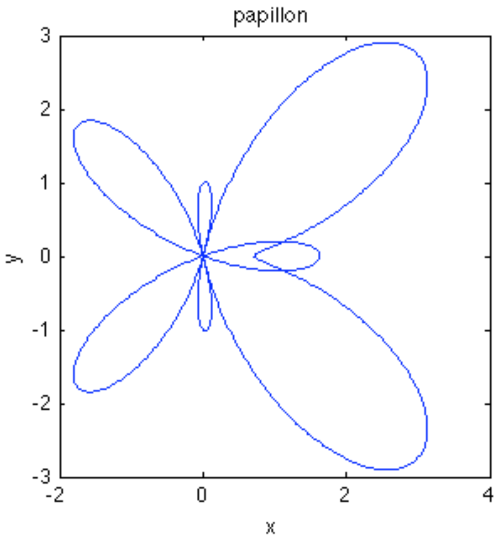
6) Superposez la courbe expérimentale avec la formule théorique pour g=5. Cette valeur de g est elle trop grande ou trop petite? Justifiez votre réponse en décrivant le graphique.
7) Superposez la courbe expérimentale avec la formule théorique pour 20 valeurs successives de g entre 5 et 15. En déduire une valeur estimée de g.
8) Maintenant une dernière manière pour estimer g : pour chaque valeur successive de g entre 5 et 15, calculez l'erreur E au sens des moindres carrés entre la courbe expérimentale et la courbe théorique. Cette erreur est la somme des carrés de l'erreur entre la valeur expérimentale et la valeur théorique. Tracez la courbe de E en fonction de g et déduisez-en l'accélération de la gravité à la surface de notre planète la terre.



Ex I Compétences générales

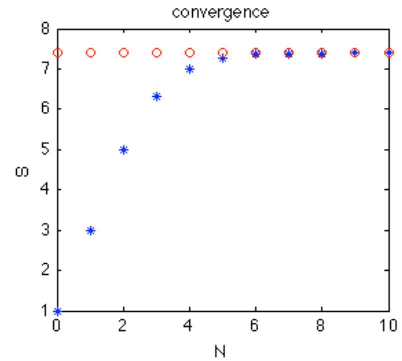
1) Tracez la courbe paramétrée en coordonnées polaires («papillon de T. Fay», cf mathcurve.com) pour le paramètre θ variant de 0 à 2π:

```
% le tableau des valeurs de theta
th=linspace(0, 2*pi,1000);

% evolution du rayon
r=exp(cos(th))-2*cos(4*th);

% transformation en coordonnées cartésoennes
x=r.*cos(th);
y=r.*sin(th);

% on trace
plot(x,y,'b')
xlabel('x');ylabel('y')
title('papillon');
```



2) Ecrivez un code Matlab qui calcule la somme suivante (expansion de Taylor de l'exponentielle en 0) pour une valeur donnée a=2:

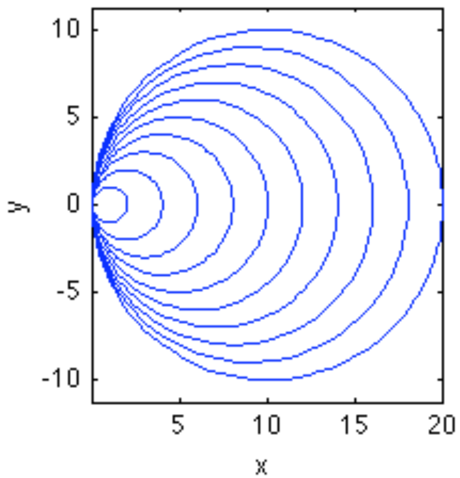
```
s=0; % on initialise la somme
N=10; % le nombre d'itérations
a=2;
for k=0:N % la boucle
s=s+a^k/(factorial(k));
% on trace la somme et la valeur de référence
plot(k,s,'*',k,exp(a),'ro'); hold on
end
xlabel('N');ylabel('S'); title('convergence');
```



Tracez un graphique qui montre que cette somme tend vers exp(2) lorsque N devient grand.

3) Ecrivez un code matlab qui reproduit qualitativement la figure ci dessous (un cercle dont on fait varier le rayon et le centre):

```
% le tableau des theta
th=linspace(0,2*pi,100);
% les valeurs de x et y pour un cercle de centre 0
x=cos(th);
y=sin(th);

% une boucle pour faire varier le rayon et le centre
for a=1:10;
    % on trace un cercle
    plot(x*a+a,y*a,'b','linewidth',1);
    hold on
end
% on annote le graphique
axis equal
xlabel('x'); ylabel('y')
```



Ex2 Chute libre

L'image chutelibre.tif représente la chute d'une boule sans vitesse initiale. Cet objet est soumis à la seule force de son poids, puisque aux vitesses qu'elle atteint et pour sa masse, on peux négliger la contribution des forces de frottement aérodynamiques. La boule chute donc à accélération constante, ce qui fait que la distance x au point de lâcher évolue comme:

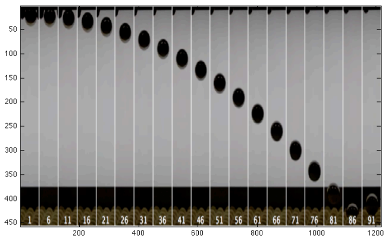
Ou g est l'accélération de la gravité dont nous allons estimer la valeur grâce à cette expérience, et t est le temps compté depuis le lâcher de la boule (l'image initiale).

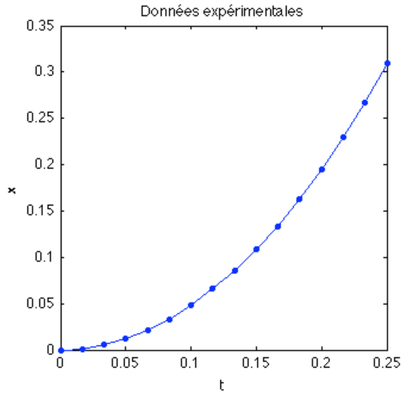
1) Lisez l'image chutelibre.tif et affichez là dans une fenêtre graphique.
2) Calculez la taille d'un pixel de l'image en prenant pour étalon de longueur la hauteur de fenêtre lumineuse dans le fond de l'image H=35.5cm.
3) Le vecteur temps: Le film est pris à 300 prises de vue par seconde, et le numéro de chaque prise de vue est affiché sur l'image. Construisez le tableau tvec qui contient les valeurs successives des temps de notre expérience.
4) Avec la fonction ginput, mesurez les valeurs successives de x(t) en mètres: la hauteur entre le point le plus bas de la boule à t=0, et le point le plus bas de la boule aux temps t successifs. Vous utiliserez pour la transformation de pixel en mètre la taille d'un pixel que vous avez calculée à la question 2.
5) Tracez le graphique de x (en mètres) en fonction de t (en secondes). Annotez votre graphique: labels et titre. Voici notre courbe expérimentale.

Nous allons maintenant comparer ce graphique expérimental avec la formule théorique.

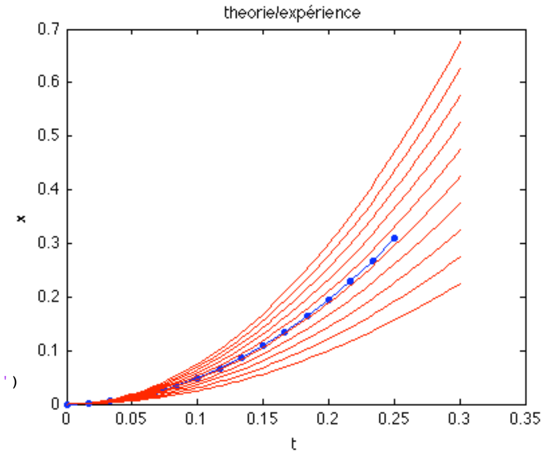
6) Superposez la courbe expérimentale avec la formule théorique pour g=5. Cette valeur de g est elle trop grande ou trop petite? Justifiez votre réponse en décrivant le graphique.
7) Superposez la courbe expérimentale avec la formule théorique pour 20 valeurs successives de g entre 5 et 15. En déduire une valeur estimée de g.
8) Maintenant une dernière manière pour estimer g: pour chaque valeur successive de g entre 5 et 15, calculez l'erreur E au sens des moindres carrés entre la courbe expérimentale et la courbe théorique. Cette erreur est la somme des carrés de l'erreur entre la valeur expérimentale et la valeur théorique. Tracez la courbe de E en fonction de g et déduisez en l'accélération de la gravité à la surface de notre planète la terre.

```
% chute libre

% on lit et affiche l'image
a=imread('films_matlab/chutelibre.tif');
subplot(1,2,1)
image(a);

% la taille des pixels en mètres
taillepix=0.355/(376-8);

% les points mesurés sur l'image
d=[ 36.8688   44.2553
  101.0489   45.2788
  164.3733   50.3964
  227.6977   57.0493
  291.0222   66.7727
  354.3466   78.5431
  417.6710   94.4076
  481.8512  112.3191
  546.0313  132.7895
  610.2115  156.3303
  674.3916  182.9418
  737.7160  213.6473
  801.0405  246.3998
  866.0764  281.7111
  930.2565  321.6282
  994.4367  365.6394];

% transformation de référentiel
x=d(:,2);
x=taillepix*(x-x(1));


subplot(1,2,2);
% le vecteur du temps
t=((1:5:76)-1)/300;
% on trace les données expérimentales
plot(t,x,'b.-');
xlabel('t'); ylabel('x'); title('Données expérimentales')
```

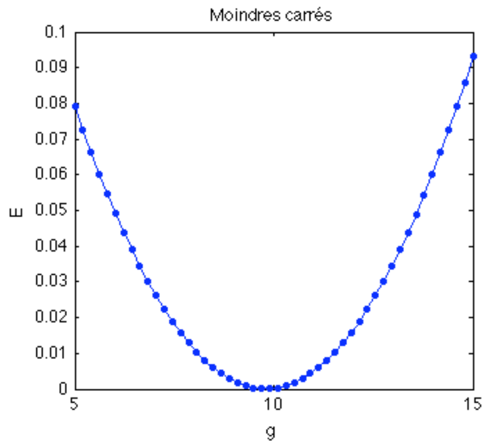





```
hold on

% boucle pour faire varier g
for g=linspace(5,15,10)
tt=linspace(0,0.3,100);
xtheo=g*tt.^2/2;

plot(tt,xtheo,'r')
end
xlabel('t'); ylabel('x'); title('theorie/expérience')
```




```
% calcul de l'erreur moindres carrés
N=50; % le nombre de valeurs de g à tester
gvec=linspace(5,15,N); % les valeurs de d
Evec=zeros(N,1); % un tableau pour mémoriser l'erreur

% la boucle
for ind=1:length(gvec)
g=gvec(ind); % la valeur actuelle du diamètre

% la formule théorique
xtheo=g*t.^2/2;
% calcul de l'erreur aux moindres carrés
Evec(ind)=sum((xtheo-x').^2);
end

% on trace le graphique
plot(gvec,Evec,'b.-');
xlabel('g'); ylabel('E'); title('Moindres carrés')
```



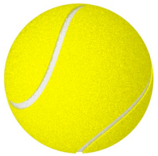
Matlab : applications en mécanique. LA207

Université Pierre et Marie Curie.
Licence de mécanique.
Rattrapage, juin 2012.

Ex1 Compétences générales

1) Pour tracer des lignes en 3D, on utilise la fonction plot3(x,y,z), ou x, y et z sont les tableaux des valeurs successives des coordonnées des points de la courbe. Tracez la courbe paramétrée de la couture de la balle de tennis (cf mathcurve.com) pour le paramètre t variant de 0 à 2π, et a=3, b=1, c=$2\sqrt{3}$:

$$\begin{cases} x = a\cos t + b\cos 3t \\ y = a\sin t - b\sin 3t \\ z = c\sin 2t \end{cases}$$

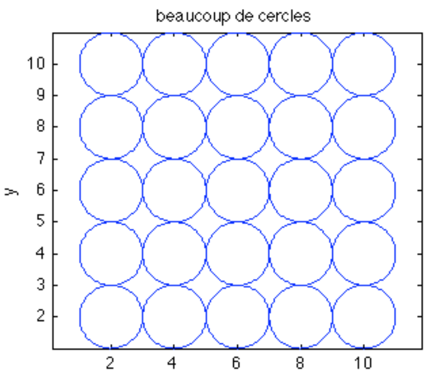


2) Ecrivez un code Matlab qui calcule le produit suivant P_N pour N=10 en utilisant une boucle «for»:

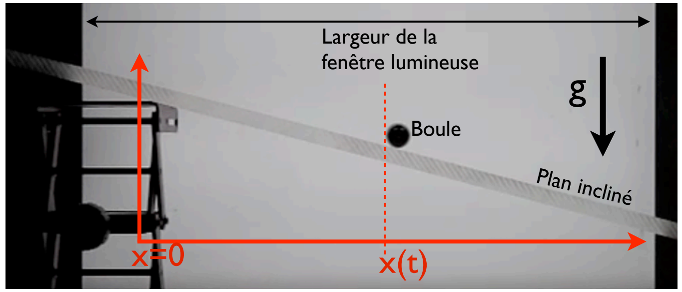
$$P_N = \prod_{k=1}^{N} \frac{4k^2}{4k^2-1} = \frac{4\times 1^2}{4\times 1^2 -1} \times \frac{4\times 2^2}{4\times 2^2 -1} \times \frac{4\times 3^2}{4\times 3^2 -1} \times \cdots \times \frac{4\times N^2}{4\times N^2 -1}$$

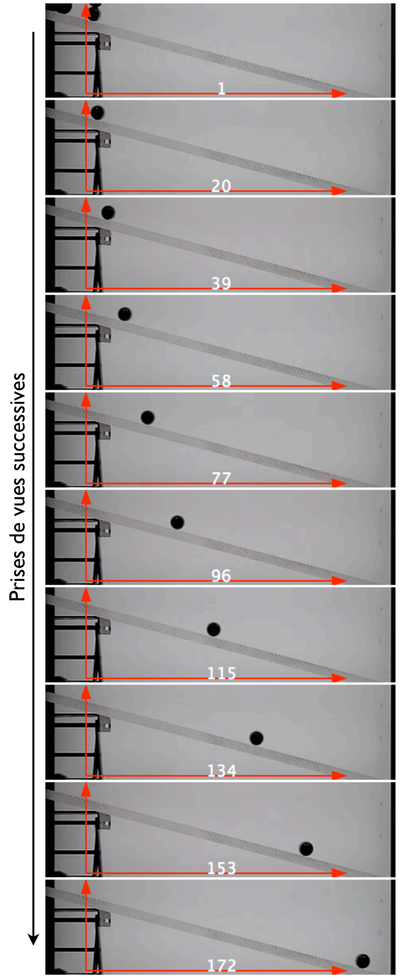
Tracez un graphique qui montre que ce produit tend vers π/2 lorsque N devient grand.

3) Ecrivez un code matlab qui reproduit qualitativement la figure ci dessous (un cercle que l'on translate vers le haut et vers le bas):



Ex2 roulement sur plan incliné





L'image roulepetit.tif représente le roulement d'une boule sur un plan incliné sans vitesse initiale. Cet objet est soumis à son poids et à la résistance du support, puisque aux vitesses qu'elle atteint et pour sa masse, on peux négliger la contribution des forces de frottement. La boule roule donc à accélération constante, ce qui fait que la distance horizontale parcourue x évolue comme :

$$x = Kt^2, \quad K = \frac{10}{28} g\cos(\alpha)\sin(\alpha)$$

Ou g=9.81 est l'accélération de la gravité et α est l'angle entre le plan incliné et l'horizontale dont nous allons estimer la valeur grâce à cette expérience, et t est le temps compté depuis le lâcher de la boule (l'image initiale).

1) Lisez l'image roulepetit.tif et affichez là dans une fenêtre graphique.
2) Calculez la taille d'un pixel de l'image en mètres, en prenant pour étalon de longueur la largeur de la fenêtre lumineuse dans le fond de l'image H=0.355m.
3) Le vecteur temps: Le film est pris à 300 prises de vue par seconde, et le numéro de chaque prise de vue est affiché sur l'image. Construisez le tableau tvec qui contient les valeurs successives des temps de notre expérience (ce tableaux a dix éléments: le nombre de prises de vue).
4) Avec la fonction ginput, mesurez les valeurs successives de x(t) en mètres: la distance horizontale entre le point le plus à gauche de la boule à t=0, et le point le plus à gauche de la boule aux temps t successifs. Vous utiliserez pour la transformation de pixel en mètre la taille d'un pixel que vous avez calculée à la question 2.
5) Tracez le graphique de x (en mètres) en fonction de t (en secondes). Annotez votre graphique : labels et titre. Voici notre courbe expérimentale.

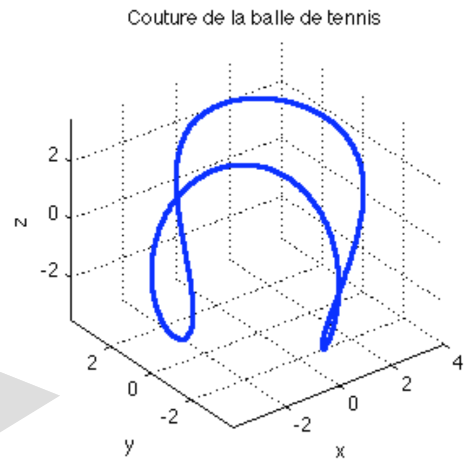
Nous allons maintenant comparer ce graphique expérimental avec la formule théorique.

6) Superposez la courbe expérimentale avec la formule théorique pour α=0.1 radians. Cette valeur de α est elle trop grande ou trop petite? Justifiez votre réponse en décrivant le graphique.
7) Superposez la courbe expérimentale avec la formule théorique pour 10 valeurs successives de α entre 0.2 et 0.4. En déduire une valeur estimée de α.
8) Mesurez l'angle α sur l'image et comparez cette valeur à la valeur estimée précédemment.

Ex 1 Compétences générales

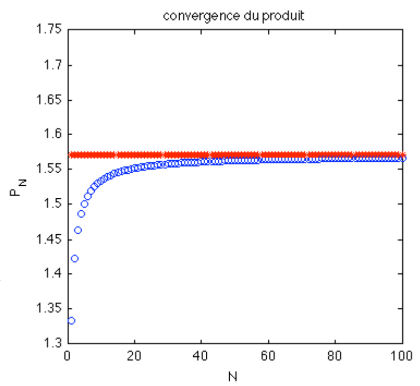
1) Pour tracer des lignes en 3D, on utilise la fonction plot3(x,y,z), ou x, y et z sont les tableaux des valeurs successives des coordonnées des points de la courbe. Tracez la courbe paramétrée de la couture de la balle de tennis (cf mathcurve.com) pour le paramètre t variant de 0 à 2π, et a=3, b=1, c=2√3 :

```
% couture de la balle de tennis

% valeurs du paramètre t
th=linspace(0,2*pi,1000);

% les trois paramètres a,b et c
a=3; b=1; c=2*sqrt(3);

% les tableaux de coordonnées
x=a*cos(th)+b*cos(3*th);
y=a*sin(th)-b*sin(3*th);
z=c*sin(2*th);

% on trace le graphique
plot3(x,y,z,'linewidth',2);
grid on

% annotations du graphique
axis equal
xlabel('x'); ylabel('y'); zlabel('z')
title('Couture de la balle de tennis');
```



2) Ecrivez un code Matlab qui calcule le produit suivant P_N pour N=10 en utilisant une boucle «for»:

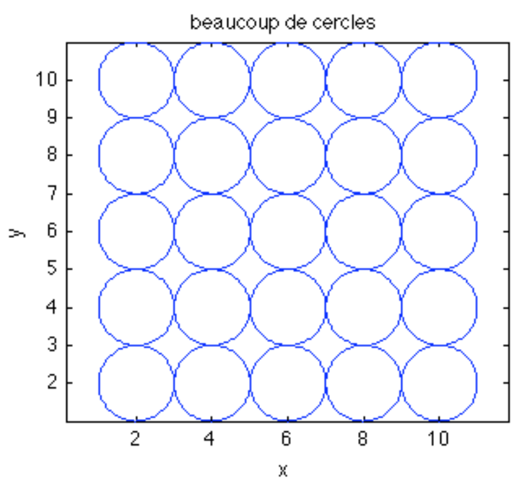
```
% un produit infini
set(gcf,'color','w')


p=1 %initialisation du produit
for k=1:100;
    % on ajoute le terme courant au produit
    p=p*(4*k^2)/(4*k^2-1);

    % on trace la valeur actuelle de P
    plot(k,p,'bo',k,pi/2,'r*');

    % pour superposer tous les points
    hold on
end
% annotations du graphique
xlabel('N'); ylabel('P_N');
title('convergence du produit');
```



Tracez un graphique qui montre que ce produit tend vers π/2 lorsque N devient grand.

3) Ecrivez un code matlab qui reproduit qualitativement la figure ci dessous (un cercle que l'on translate vers le haut et vers le bas):

```
% un graphique intéressant

% la tableau des theta
th=linspace(0,2*pi,100);

% transformation en coordonnées
% cartésiennes pour un cercle
x=cos(th); y=sin(th);

% une double boucle
for i=1:5;
    for j=1:5;
        % on trace un cercle qui est translaté
        plot(x+2*i,y+2*j);
        hold on % pour superposer les graphiques
    end
end

% annotations du graphique
axis equal
xlabel('x'); ylabel('y');
title('beaucoup de cercles');
```



Ex2 roulement sur plan incliné

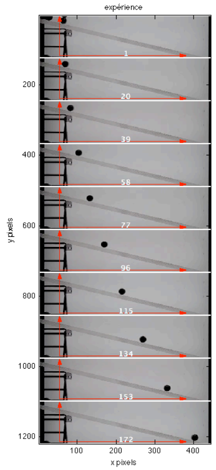
L'image roulepetit.tif représente le roulement d'une boule sur un plan incliné sans vitesse initiale. Cet objet est soumis à son poids et à la résistance du support, puisque aux vitesses qu'elle atteint et pour sa masse, on peux négliger la contribution des forces de frottement. La boule roule donc à accélération constante, ce qui fait que la distance horizontale parcourue x évolue comme :

Ou g=9.81 est l'accélération de la gravité et α est l''angle entre le plan incliné et l'horizontale dont nous allons estimer la valeur grâce à cette expérience, et t est le temps compté depuis le lâcher de la boule (l'image initiale).

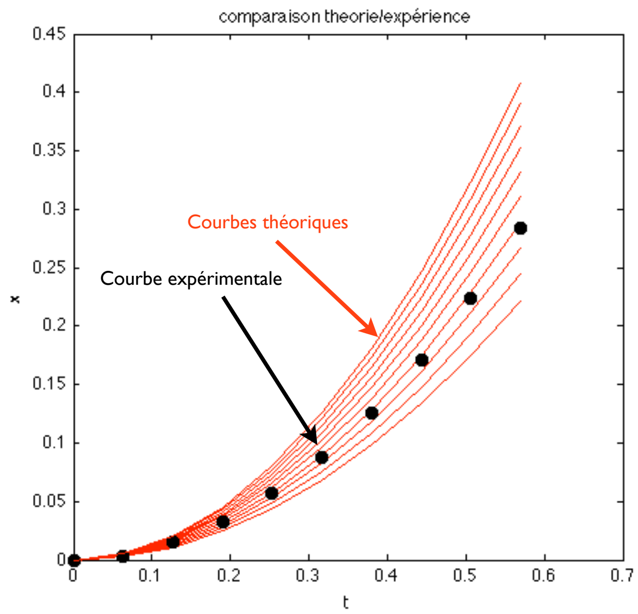
1) Lisez l'image roulepetit.tif et affichez là dans une fenêtre graphique.
2) Calculez la taille d'un pixel de l'image en mètres, en prenant pour étalon de longueur la largeur de la fenêtre lumineuse dans le fond de l'image H=0.355m.
3) Le vecteur temps: Le film est pris à 300 prises de vue par seconde, et le numéro de chaque prise de vue est affiché sur l'image. Construisez le tableau tvec qui contient les valeurs successives des temps de notre expérience (ce tableaux a dix éléments: le nombre de prises de vue).
4) Avec la fonction ginput, mesurez les valeurs successives de x(t) en mètres: la distance horizontale entre le point le plus à gauche de la boule à t=0, et le point le plus à gauche de la boule aux temps t successifs. Vous utiliserez pour la transformation de pixel en mètre la taille d'un pixel que vous avez calculée à la question 2.
5) Tracez le graphique de x (en mètres) en fonction de t (en secondes). Annotez votre graphique : labels et titre. Voici notre courbe expérimentale.

Nous allons maintenant comparer ce graphique expérimental avec la formule théorique.

6) Superposez la courbe expérimentale avec la formule théorique pour α=0.1 radians. Cette valeur de α est elle trop grande ou trop petite? Justifiez votre réponse en décrivant le graphique.
7) Superposez la courbe expérimentale avec la formule théorique pour 10 valeurs successives de α entre 0.2 et 0.4. En déduire une valeur estimée de α.
8) Mesurez l'angle α sur l'image et comparez cette valeur à la valeur estimée précédemment.




```
% On charge et on affiche l'image
a=imread('films_matlab/roulepetit.tif');
subplot(1,2,1)
image(a);

% annotations
xlabel('x pixels');ylabel('y pixels'); title('expérience')

% les points de mesure avec ginput
d=1000*[  0.0560    0.0154
    0.0603    0.1424
    0.0738    0.2663
    0.0947    0.3933
    0.1235    0.5250
    0.1609    0.6568
    0.2064    0.7901
    0.2603    0.9249
    0.3223    1.0613
    0.3935    1.2056];

% on ne prend que le x
x=d(:,1);

% changement de référentiel
taillepix=0.355/(438-15);
x=(x-x(1))*taillepix;

subplot(1,2,2);

% le vecteur temps
tvec=((1:19:172)-1)/300;

% paramètres
g=9.81;
%alpha=(14.7/360)*2*pi;

for alpha=linspace(0.2,0.4,10)

% la loi théorique
xtheo=cos(alpha)*tvec.^2*g*sin(alpha)*(10/28);

% on trace le graphique
plot(tvec,x,'k.',tvec,xtheo,'r','markersize',20)
hold on
end

% annotations
xlabel('t');ylabel('x'); title('comparaison theorie/expérience')
```

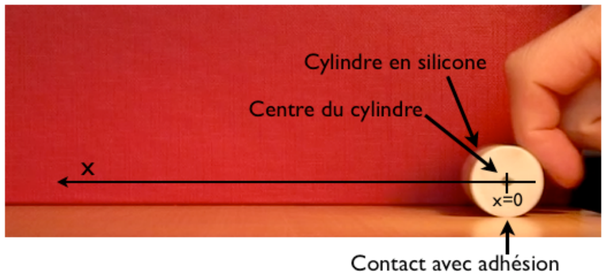


Examen LA207
Sujet du matin
Matlab: applications en mécanique
Mercredi 25 mai 2011

Université Pierre et Marie Curie.
`www.lmm.jussieu.fr/~hoepffner/enseignement`

Tous les graphiques doivent être annotés: titres, labels des axes et légendes. Les scripts doivent être insérés dans le compte-rendu auprès des graphiques associés. Chaque bloc de commande des scripts doit être commenté. La notation prendra en compte la qualité de la présentation. Durée: 1h 45.

2.17.14 Cylindre roulant avec adhésion



Nous considérons un cylindre en silicone qui roule sur un plan. Il y a une légère adhérence entre le plan et le cylindre, qui va freiner le roulement du cylindre. Nous voulons étudier comment se comporte cette adhérence. Pour ceci, nous avons réalisé un film à partir duquel nous voudrions tracer les graphiques de la position, vitesse et accélération du cylindre.

1. **Données:** Les données sont stockées sur le disque dans le fichier `cylindre.mat`. Chargez-le dans matlab avec la commande `load cylindre.mat`. une fois cette commande exécutée vous avez disponible la matrice `m` qui a trois dimensions: `m(:,:,5)` par exemple correspond au tableau des tons de gris pour la cinquième image. Donnez dans votre compte rendu le nombre `nx` de pixels selon l'horizontale et `ny` selon la verticale, ainsi que le nombre `n` d'images qui composent

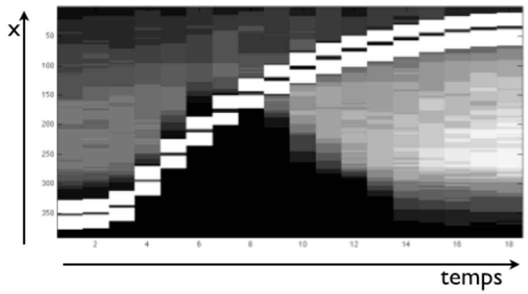
ce film. Tracez la première image dans une figure graphique avec la fonction `imagesc`.

2. **Animation** Réalisez l'animation du film avec une boucle `for` et la fonction `drawnow`. Il y a peu d'images, donc pour que l'animation ne soit pas trop courte, utilisez la fonction `pause`.

3. **Taille d'un pixel:** Le cylindre sur le film à un diamètre de 0.025 mètres. Déduisez-en la taille en mètres d'un pixel, et indiquez cette valeur dans le compte-rendu.

4. **Le vecteur temps:** Créez le vecteur temps `tvec`, sachant que le film est enregistré à 20 images par seconde. La première image correspondra au temps 0. Donnez la valeur en secondes de l'intervalle de temps Δt entre les images.

5. **Diagramme spatio-temporel** Nous allons maintenant réaliser un diagramme spatio-temporel à partir de ce film, qui va nous faciliter la mesure de l'avancée du cylindre dans le temps. Sur une image quelconque du film, repérez avec l'outil d'étiquetage la position verticale `loc` à laquelle se situe le point noir qui est au centre du cylindre. La matrice `d=squeeze(m(loc,:,:))` est ainsi un diagramme spatio-temporel: image qui trace l'évolution de la position du cylindre dans le temps. Tracez cette image:



6. **Mesure:** Avec la fonction `ginput`, mesurez sur le diagramme spatio-temporel l'évolution dans le temps du centre du cylindre x. Faites le changement de référentiel pour avoir des données en mètres et pour que $x = 0$ corresponde à la première image. Tracez une figure qui représente l'évolution de x dans le temps.

7. **Vitesse et acceleration:** Calculez et tracez la vitesse $\dot{x}$ et l'accélération $\ddot{x}$[9]. On rappelle la formule pour la vitesse au temps i:

$$\dot{x}_i = (x_{i+1} - x_{i-1})/(2\Delta t).$$

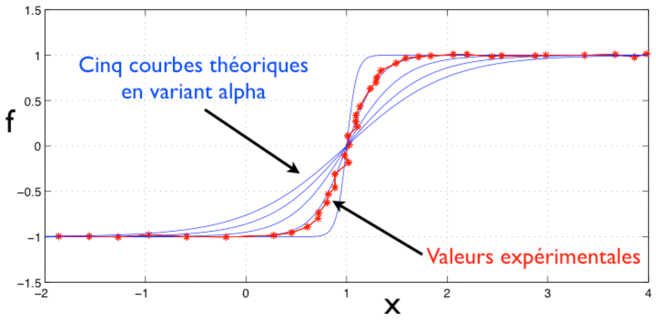
8. **Mesure du coefficient de frottement:** Il y a initialement une grande accélération positive à cause de la poussée du doigt, puis l'accélération devient négative (le cylindre ralentit) et suit une loi approximativement linéaire, ce qui montre que la loi d'évolution est $\ddot{x} + \alpha\dot{x} = 0$. Tracez l'évolution dans le temps de $\ddot{x}/\dot{x}$ et déduisez-en une valeur approximative du coefficient de frottement α.

2.17.15 Données expérimentales et théorie

Nous considérons un phénomène physique décrit par la formule

$$f(x) = \tanh\left(\frac{x-1}{\alpha}\right)$$

mais nous ne connaissons pas la valeur du paramètre α, nous allons l'estimer à partir de données expérimentales bruitées.



1. **Données:** Les données sont stockées sur le disque dans le fichier `tanhdata.txt`. Chargez ces données avec la commande `d=load('tanhdata.txt')`. La première colonne de `d` correspond aux valeurs de x : $(x_1, x_2, \dots)$ la seconde colonnes correspond aux valeurs expérimentales de f : $(f_1, f_2, \dots)$. Tracez ces données dans un graphique.

[9] L'acceleration, c'est la "vitesse de la vitesse".

2. **Courbes théoriques:** Sur ce même graphique, tracez la formule théorique pour 5 valeurs du paramètre α dans l'intervalle $[0.1, 1]$ en utilisant une boucle `for`. Déduisez-en une première estimation de la valeur de α.

3. **Transformation:** Une seconde méthode: la fonction $g = \operatorname{atanh}(f)$[10] est une droite d'équation $(x-1)/\alpha$ (bruitée). Tracez g en fonction de x pour les données expérimentales. Mesurez graphiquement la pente de $g(x)$ dans le voisinage de $x = 1$ et déduisez-en une seconde estimation de la valeur de α.

4. **Pente:** Superposez au graphique précédent une droite qui correspond à la pente mesurée, ligne pointillée, couleur noire.

5. **Moindres carrés:** Pour $\alpha = 1$ calculez l'erreur entre théorie et données expérimentales avec la méthode des moindres carrés[11]:

Erreur pour le premier point

$$E = (\underbrace{e_1}_{f_1 - f(x_1)})^2 + (\underbrace{e_2}_{f_2 - f(x_2)})^2 + \dots$$

Valeur expérimentale

Valeur théorique

Donnez la valeur de cette erreur dans votre compte-rendu.

[10] fonction arctangente hyperbolique

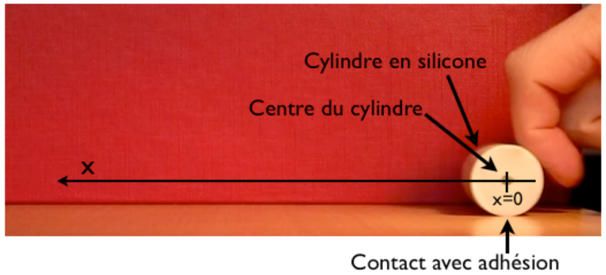
[11] La somme pour tous les points de mesure du carré de la différence entre la valeur mesurée et la valeur théorique

Examen LA207
Sujet de l'après-midi
Matlab: applications en mécanique
Mercredi 25 mai 2011

Université Pierre et Marie Curie.
www.lmm.jussieu.fr/~hoepffner/enseignement

Tous les graphiques doivent être annotés: titres, labels des axes et légendes. Les scripts doivent être insérés dans le compte-rendu auprès des graphiques associés. Chaque bloc de commande des scripts doit être commenté. La notation prendra en compte la qualité de la présentation. Durée: 1h 45.

2.17.16 Cylindre roulant avec adhésion



Nous considérons un cylindre en silicone qui roule sur un plan. Il y a une légère adhérence entre le plan et le cylindre, qui va freiner le roulement du cylindre. Nous voulons étudier comment se comporte cette adhérence. Pour ceci, nous avons réalisé un film à partir duquel nous voudrions tracer les graphiques de la position, vitesse et accélération du cylindre.

1. **Données:** Les données sont stockées sur le disque dans le fichier `cylindre.mat`. Chargez-le dans matlab avec la commande `load cylindre.mat`. une fois cette commande exécutée vous avez disponible la matrice `m` qui a trois dimensions: `m(:,:,5)` par exemple correspond au tableau des tons de gris pour la cinquième image. Donnez dans votre compte rendu le nombre `nx` de pixels selon l'horizontale et `ny` selon la verticale, ainsi que le nombre `n` d'images qui composent

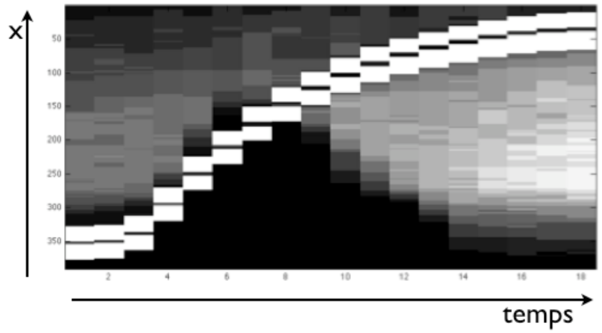
ce film. Tracez la première image dans une figure graphique avec la fonction `imagesc`.

2. **Animation** Réalisez l'animation du film avec une boucle `for` et la fonction `drawnow`. Il y a peu d'images, donc pour que l'animation ne soit pas trop courte, utilisez la fonction `pause`.

3. **Taille d'un pixel:** Le cylindre sur le film à un diamètre de 0.025 mètres. Déduisez-en la taille en mètres d'un pixel, et indiquez cette valeur dans le compte-rendu.

4. **Le vecteur temps:** Créez le vecteur temps `tvec`, sachant que le film est enregistré à 20 images par seconde. La première image correspondra au temps 0. Donnez la valeur en secondes de l'intervalle de temps Δt entre les images.

5. **Diagramme spatio-temporel** Nous allons maintenant réaliser un diagramme spatio-temporel à partir de ce film, qui va nous faciliter la mesure de l'avancée du cylindre dans le temps. Sur une image quelconque du film, repérez avec l'outil d'étiquetage la position verticale `loc` à laquelle se situe le point noir qui est au centre du cylindre. La matrice `d=squeeze(m(loc,:,:))` est ainsi un diagramme spatio-temporel: image qui trace l'évolution de la position du cylindre dans le temps. Tracez cette image:



6. **Mesure:** Avec la fonction `ginput`, mesurez sur le diagramme spatio-temporel l'évolution dans le temps du centre du cylindre x. Faites le changement de référentiel pour avoir des données en mètres et pour

que $x = 0$ corresponde à la première image. Tracez une figure qui représente l'évolution de x dans le temps.

7. **Vitesse et acceleration:** Calculez et tracez la vitesse $\dot{x}$ et l'accélération $\ddot{x}$[12]. On rappelle la formule pour la vitesse au temps i:

$$\dot{x}_i = (x_{i+1} - x_{i-1})/(2\Delta t).$$

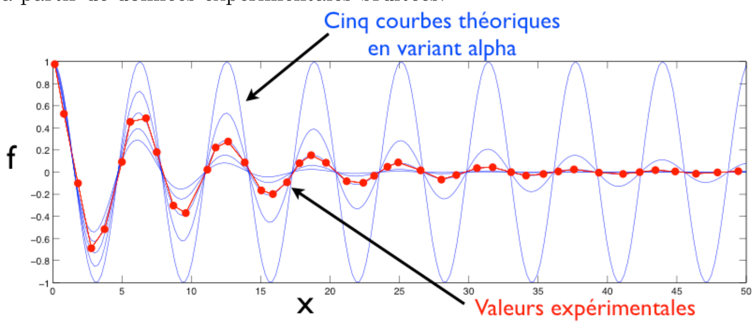
8. **Mesure du coefficient de frottement:** Il y a initialement une grande accélération positive à cause de la poussée du doigt, puis l'accélération devient négative (le cylindre ralentit) et suit une loi approximativement linéaire, ce qui montre que la loi d'évolution est $\ddot{x} + \alpha\dot{x} = 0$. Tracez l'évolution dans le temps de $\ddot{x}/\dot{x}$ et déduisez-en une valeur approximative du coefficient de frottement α.

2.17.17 Données expérimentales et théorie

Nous considérons un phénomène physique décrit par la formule

$$f(x) = \exp(-\alpha x)\cos(x)$$

mais nous ne connaissons pas la valeur du paramètre α, nous allons l'estimer à partir de données expérimentales bruitées.



1. **Données:** Les données sont stockées sur le disque dans le fichier `cosdata.txt`. Chargez ces données avec la commande `d=load('cosdata.txt')`. La première colonne de `d` correspond aux valeurs expérimentales de x : $(x_1, x_2, \dots)$ la seconde colonnes correspond aux valeurs expérimentales de f : $(f_1, f_2, \dots)$. Tracez ces données dans un graphique.

[12] L'acceleration, c'est la "vitesse de la vitesse".

2. **Courbes théoriques:** Sur ce même graphique, tracez la formule théorique pour 5 valeurs du paramètre α dans l'intervalle $[0, 0.2]$ en utilisant une boucle `for`. Déduisez-en une première estimation de la valeur de α.

3. **transformation:** Une seconde méthode: la fonction

$$g = \log(f/\cos(x))$$

est une droite d'équation $-\alpha x$ (bruitée). Tracez g en fonction de x pour les données expérimentales. Mesurez graphiquement la pente de $g(x)$ et déduisez-en une seconde estimation de la valeur de α.

4. **Pente:** Superposez au graphique précédent une droite qui correspond à la pente mesurée, ligne pointillée, couleur noire.

5. **Moindres carrées:** Pour $\alpha = 0.2$ calculez l'erreur entre théorie et données expérimentales avec la méthode des moindres carrés[13]:

Erreur pour le premier point

$$E = (\underbrace{e_1}_{f_1 - f(x_1)})^2 + (\underbrace{e_2}_{f_2 - f(x_2)})^2 + \dots$$

Valeur expérimentale

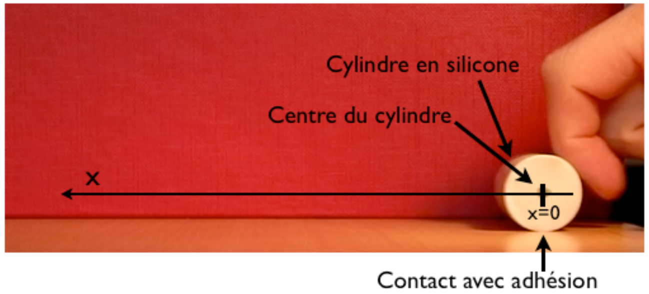
Valeur théorique

Donnez la valeur de cette erreur dans votre compte-rendu.

[13] La somme pour tous les points de mesure du carré de la différence entre la valeur mesurée et la valeur théorique

Matlab : applications en mécanique
LA207, 2010-2011
Examen 25 mai 2011:

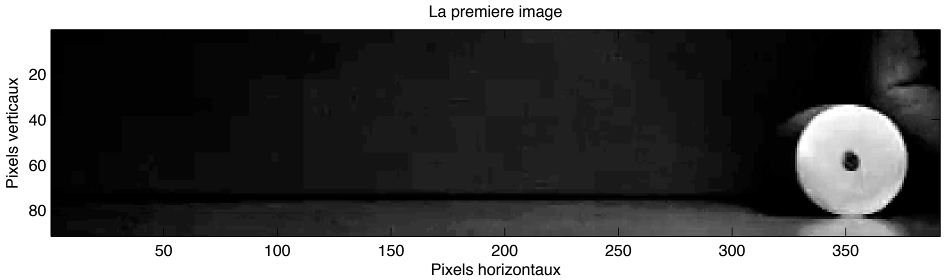
1) Cylindre roulant



On charge les données. On mesure la taille de notre matrice à trois dimensions avec la fonction «size»

```
% on charge les donnes et on mesure la taille
    load exdata/cylindre.mat
    nx=size(m,1)
    ny=size(m,2)
    n=size(m,3)

    % on affiche la premiere image
    imagesc(m(:,:,1));
    axis equal tight
    title('La premiere image')
    xlabel('Pixels horizontaux');
    ylabel('Pixels verticaux')
    colormap gray
```

On affiche la première image



Animation

Pour l'animation, on utilise «drawnow» et «pause».

```
% animation
  for ind=1:n
      % on affiche une image
      imagesc(m(:,:,ind));
      axis equal tight
      xlabel('Pixels horizontaux');
      ylabel('Pixels verticaux')
      colormap gray
      pause(0.1)
      drawnow;
  end
```

```
% taille d un pixel
taillepix= 0.025/(60-10)

% vecteur temporel
dt=1/20 % intervalle entre les images
tvec=(0:17)*dt;
```

Taille de pixel et vecteur temps

Un pixel mesure 0.0005 millimètres. L'intervale de temps Dt est 0.05 secondes

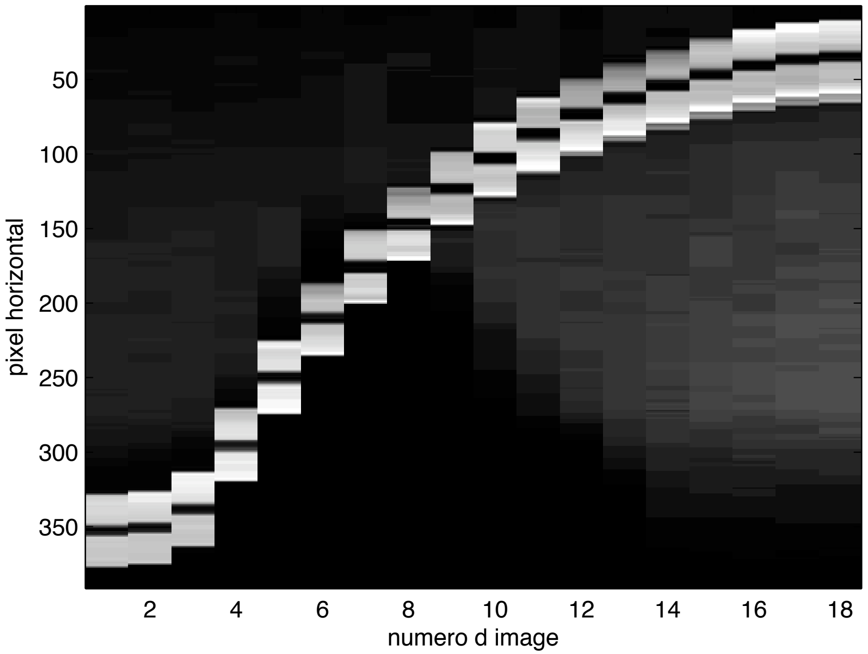


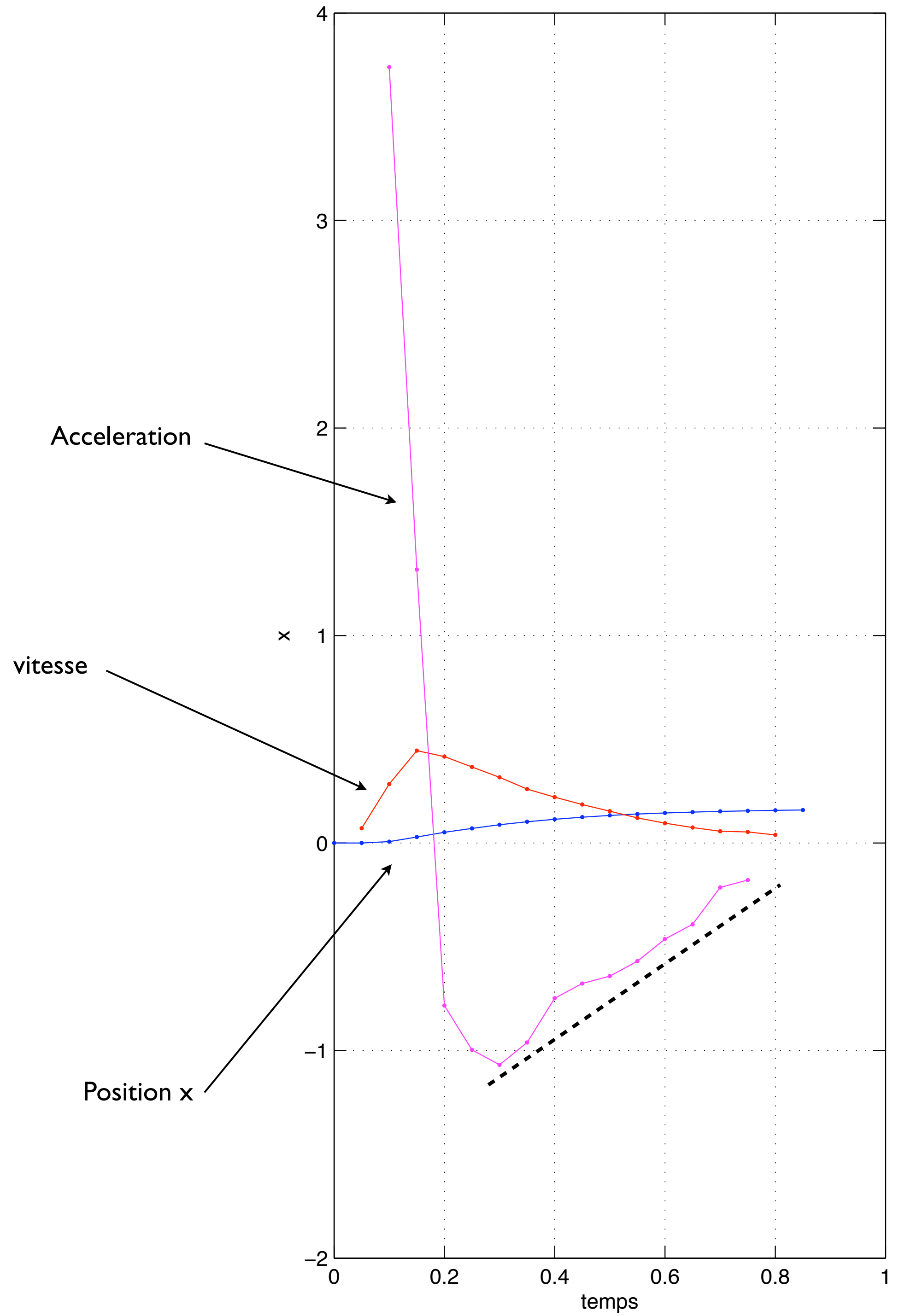
Diagramme spatio-temporel

Voici le diagramme spatio-temporel.

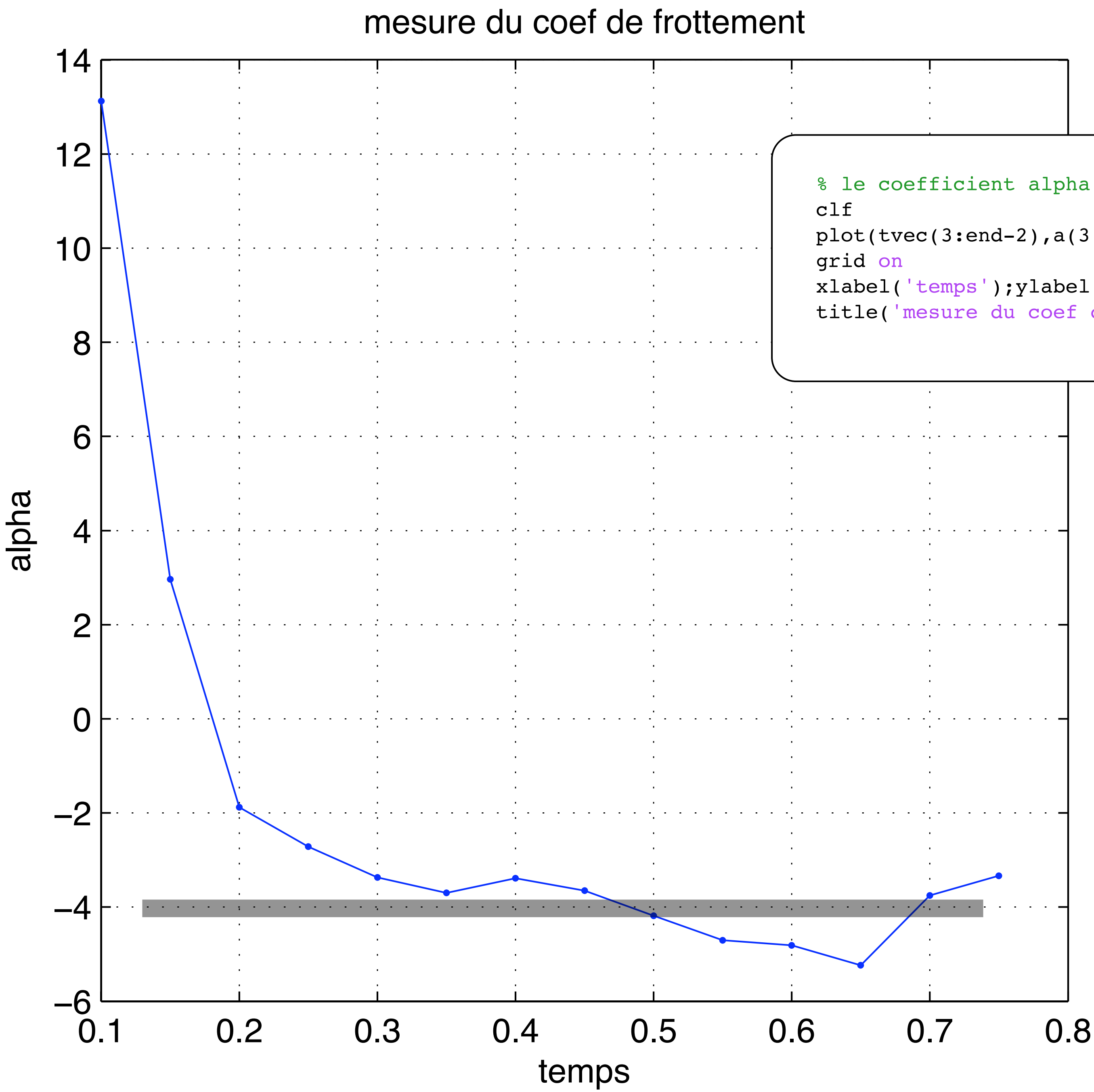
```
% juste une ligne de pixels
  % pour le diagramme spatio temporel
  dd=squeeze(m(58,:,:));
  imagesc(dd);
  xlabel('numero d image');
  ylabel('pixel horizontal')
  colormap gray
```

Position vitesse et accelleration

```
% mesures avec ginput de la position
% du centre du cylindre

d=[
0.9430   352.6849
1.8544   351.9727
2.9937   338.4408
4.0316   294.9964
5.0443   249.4153
5.9557   211.6685
7.0190   176.0583
8.0316   148.2823
8.9430   124.0674
10.0316   104.1257
11.0190    87.0328
11.9810    73.5009
12.9937    62.8179
14.0316    54.2714
15.0190    47.8616
16.0316    42.8761
17.0190    37.1785
17.9810    35.0419];

        % traitement des mesures: changement de référentiel
        y=d(:,2);
        y=-(y-y(1))*taillepix;
        plot(tvec,y,'b.-'); xlabel('temps');ylabel('x');
        grid on
        hold on

        % vitesse
        v=zeros(n,1);
        for ind=2:n-1
            v(ind)=(y(ind+1)-y(ind-1))/(2*dt);
        end
        hold on;
        plot(tvec(2:end-1),v(2:end-1),'r.-')

        % acceleration
        a=zeros(n,1);
        for ind=2:n-1
            a(ind)=(v(ind+1)-v(ind-1))/(2*dt);
        end
        hold on;
        plot(tvec(3:end-2),a(3:end-2),'m.-')
        grid on
        xlim([0,1]);
```



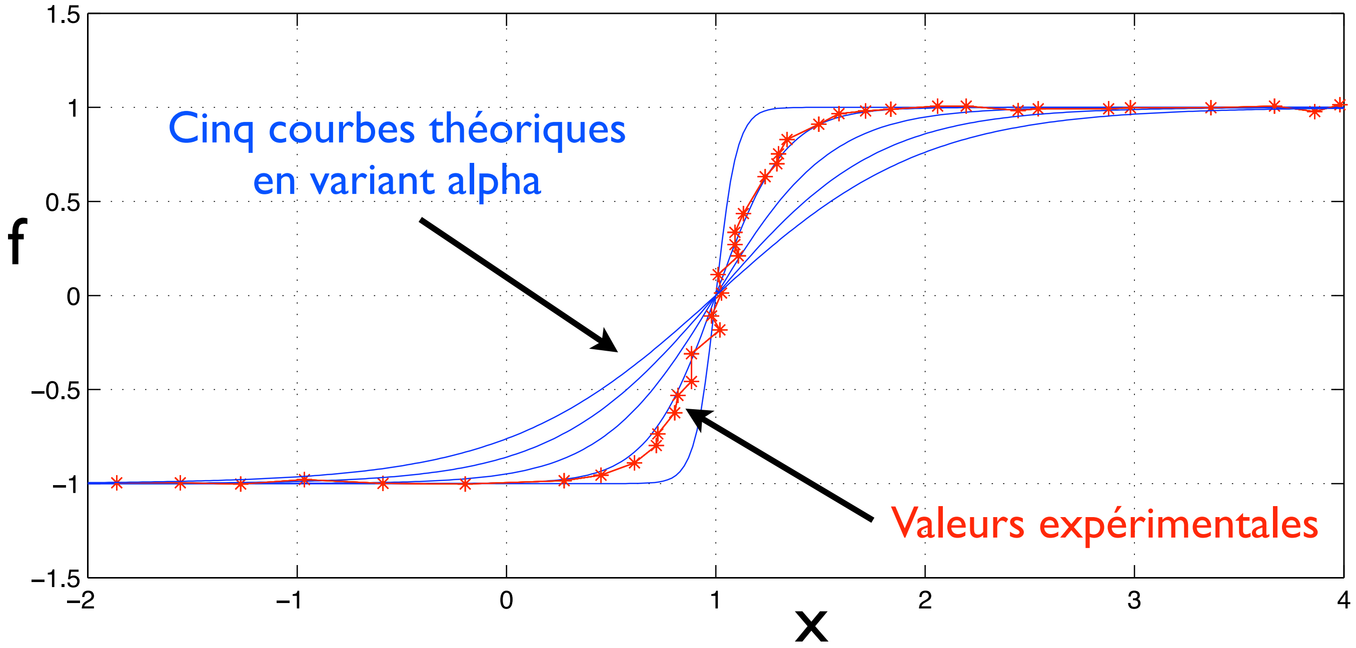
Coefficient de frottement




```
% le coefficient alpha
clf
plot(tvec(3:end-2),a(3:end-2)./v(3:end-2),'b.-')
grid on
xlabel('temps');ylabel('alpha');
title('mesure du coef de frottement')
```

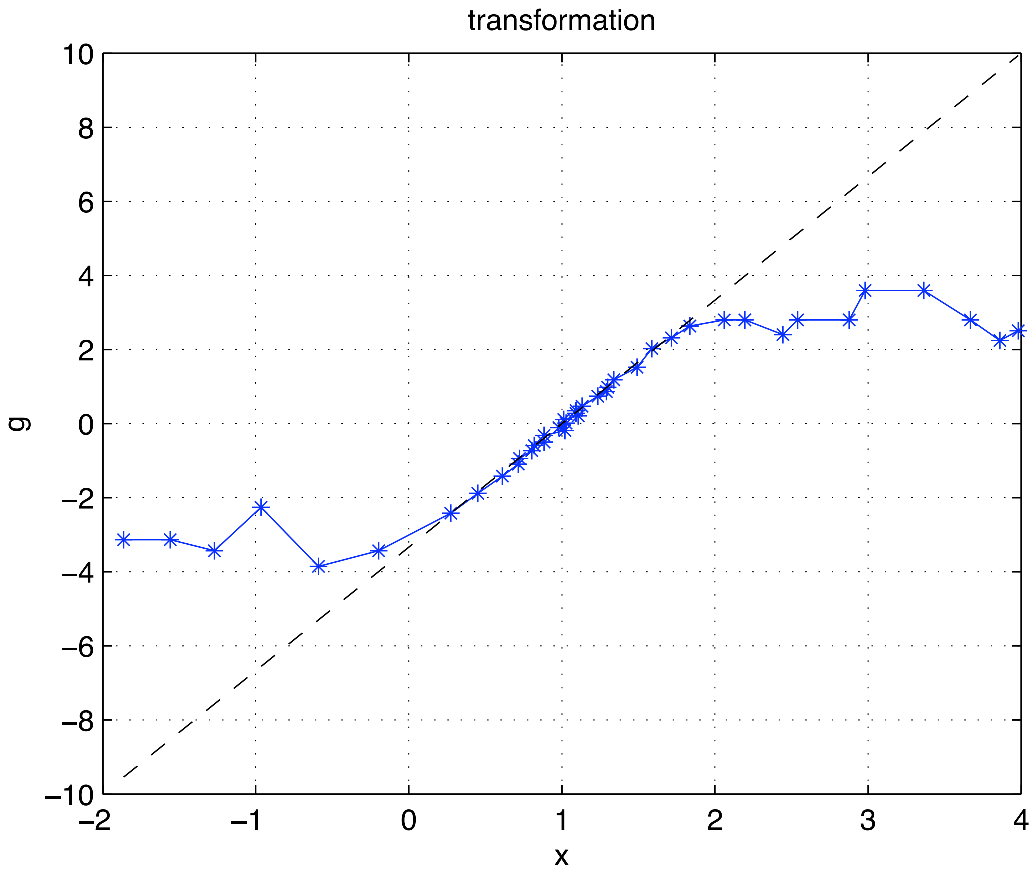
On observe que la courbe de l'acceleration divisée par la vitesse est à peu près constante et vaut -4, donc on a une estimation du coefficient de frottement alpha=4.

2) Données expérimentales et théorie: tanh



On superpose aux données expérimentales cinq courbes théoriques en faisant varier la valeur du coefficient inconnu alpha. Cela nous permet de visualiser comment alpha fait varier l'allure de la courbe théorique et aussi pour obtenir une première estimation de sa valeur numérique.

```
% une tangente hyperbolique
xvec=linspace(-2,4,300);
d=load('tanhdata.txt');

xx=d(:,1);
yy=d(:,2);

% on trace cinq valeurs de alpha
lvec=linspace(0.1,1,5);
for ind=1:length(lvec)
    ftheo=tanh((xvec-1)/lvec(ind));
    plot(xx,yy,'r*-',xvec,ftheo,'b'); hold on
end
grid on
xlabel('x'); ylabel('y'); title('theorie et mesures')
```



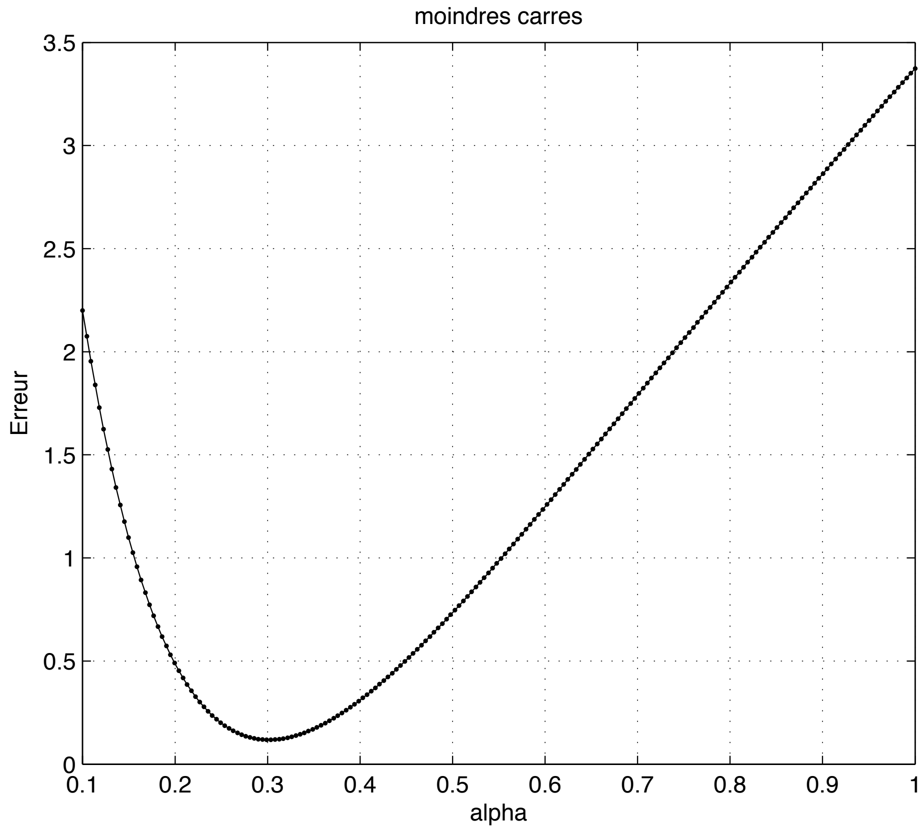
Transformation des données pour obtenir une droite

On trace les données mesurées en opérant à une transformation de sorte à ce que la courbe obtenue soit une droite dont on peut mesurer graphiquement la pente. Cette pente nous donne une estimation de la valeur numérique du paramètre alpha.

Pour bien vérifier cette valeur, nous avons tracé une droite avec la pente associée, en pointillés.

```
% transformation
clf;
plot(xx,atanh(ff),'b*-')
grid on
xlabel('x'); ylabel('g'); title('transformation')

hold on
plot(xx,(xx-1)/0.3,'k--')
```

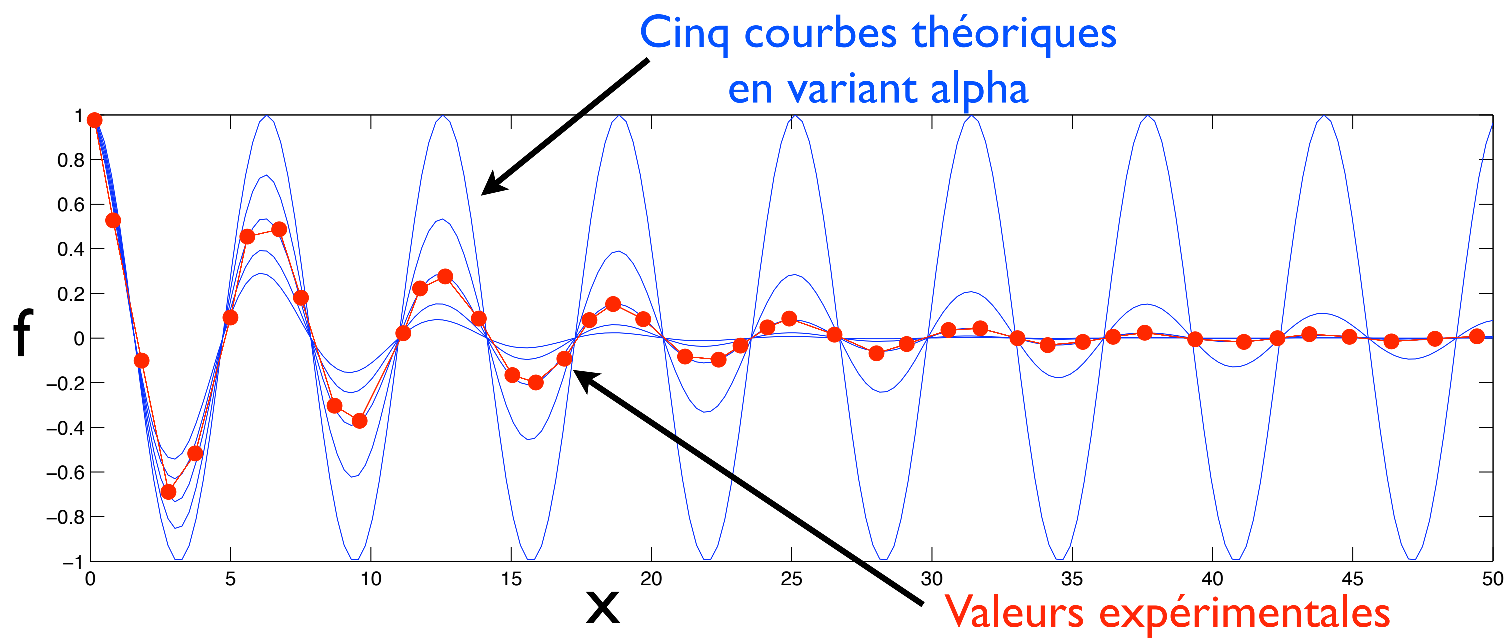
Moindres carrés



Nous avons calculé la valeur de l'erreur des moindres carrés pour un grand nombres de valeurs de alpha. Le miminum de cete courbe correspond à la valeur optimale du coefficient.

```
% calcul de l erreur moindres carres
clf
lvec=linspace(0.1,1,200);
evec=0*lvec;
for ind=1:length(lvec);
    ftheo=tanh((xx-1)/lvec(ind));
    evec(ind)=sum((ff-ftheo).^2);
end
clf;
plot(lvec,evec,'k.-');
grid on
xlabel('alpha'); ylabel('Erreur'); title('moindres carres')
```

2) Données expérimentales et théorie: cos



On superpose aux données expérimentales cinq courbes théoriques en faisant varier la valeur du coefficient inconnu alpha. Cela nous permet de visualiser comment alpha fait varier l'allure de la courbe théorique et aussi pour obtenir une première estimation de sa valeur numérique.

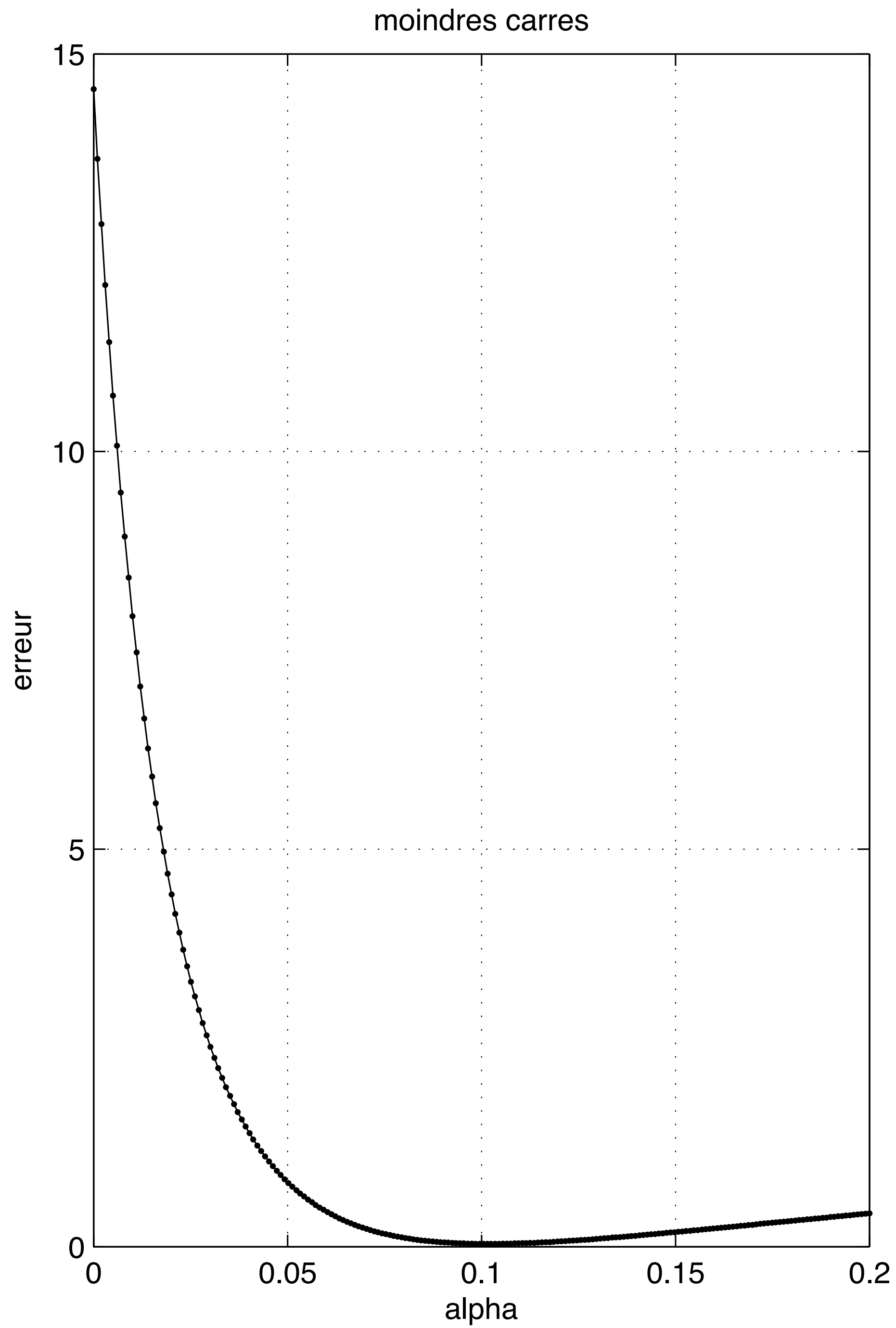
```
% on trace pour cinq valeurs de alpha
lvec=linspace(0,0.2,5);
for ind=1:length(lvec); l=lvec(ind);
    ftheo=exp(-l*tvec).*cos(tvec);
    plot(tvec,ftheo,'-',xx,ff,'r*-')
    hold on
end
```

Transformation des données pour obtenir une droite

On trace les données mesurées en opérant à une transformation de sorte à ce que la courbe obtenue soit une droite dont on peut mesurer graphiquement la pente. Cette pente nous donne une estimation de la valeur numérique du paramètre alpha.

Pour bien vérifier cette valeur, nous avons tracé une droite avec la pente associée, en pointillés.

```
% transformation
clf
plot(xx,log(ff./cos(xx)),'b*-');
grid on
hold on
plot(-0.1*xx,'k--');
xlabel('x'); ylabel('f');
title('transformation')
```



Moindres carrés

```
% moindres carres
clf;
lvec=linspace(0,0.2,200);
evec=0*lvec;
for ind=1:length(lvec);
    l=lvec(ind);
    ftheo=exp(-l*xx).*cos(xx);
    evec(ind)=sum((ff-ftheo).^2);
end
plot(lvec,evec,'k.-')
grid on
xlabel('alpha'); ylabel('erreur');
title('moindres carres')
```

Nous avons calculé la valeur de l'erreur des moindres carrés pour un grand nombres de valeurs de alpha. Le miminum de cete courbe correspond à la valeur optimale du coefficient.

Examen LA207
Matlab: applications en mécanique
Lundi 14 juin 2010

Le compte-rendu est le produit de votre travail: c'est le compte-rendu qui est noté. Les scripts et les graphiques doivent être insérés dans le compte-rendu. Commentez les scripts, annotez les graphiques, soignez votre présentation.

1 Session de rattrapage

1.1 La couture de la balle de tennis

Compétences: *Tracer des courbes paramétrées en 3D avec la fonction* `plot3`. *Représentation selon différents angles de vue.*

L'équation d'une courbe en 3D, dans un repère cartésien (x, y, z) fermée le long d'une sphère, qui a la forme de la couture d'une balle de tennis:

$$\begin{cases} x = a\cos(t) + b\cos(3t), \\ y = a\sin(t) - b\sin(3t), \\ z = c\sin(2t) \end{cases}$$

ou $t \in [0, 2\pi]$, et a, b, et c sont trois paramètres que nous pouvons faire varier pour changer l'allure de notre courbe

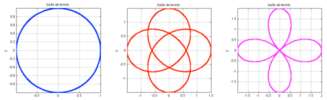
1. **Couture:** Tracez dans un graphique la courbe avec `plot3`, avec les paramètres $a = 3, b = 1, c = 2\sqrt{ab}$. Il s'agit des paramètres pour lesquelles la courbe correspond à la couture de la balle de tennis.

2. **Variation des paramètres:** Tracez dans trois sous-graphiques la courbe en 3D avec `plot3`, avec trois couleurs différentes, et avec les paramètres:

$$\begin{cases} \Gamma_1: \quad a = 0, \quad b = 1, \quad c = 2\sqrt{ab} \\ \Gamma_2: \quad a = 0.5, \quad b = 1, \quad c = 2\sqrt{ab} \\ \Gamma_3: \quad a = 1, \quad b = 1, \quad c = 2\sqrt{ab} \end{cases}$$

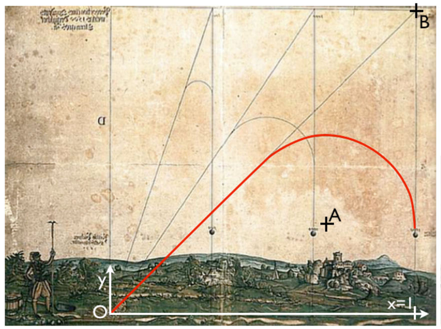
Ces courbes-ci ne correspondent en fait plus à la couture de la balle de tennis, mais montrent comment une forme peut changer lorsqu'on fait varier des paramètres numériques.

3. **Changement du point de vue** Maintenant tracez de nouveaux les trois courbes de la question précédente dans trois sous-graphiques et avec trois couleurs différentes, mais en imposant la vue de dessus avec la fonction `view(90,90)`.



1.2 Balistique de Aristote

Compétences: *Mesurer des points sur une image, et tracer la courbe obtenue après avoir fait un changement de référentiel.*



L'image ci-dessus est une illustration de la théorie balistique selon Aristote. La trajectoire d'un mobile pesant dans le champ de pesanteur serait la composition d'une ligne droite puis d'un arc de cercle. Nous savons maintenant que cette théorie est éronée, et que la trajectoire est proche d'une parabole. Dans cet exercice nous allons prendre des points de mesure sur la trajectoire.

1. **Lecture de l'image:** Lisez l'image `aristote.jpg` sur le disque avec la fonction `imread` et affichez la avec la fonction `image`. Cette image doit être insérée dans le compte-rendu. Utilisez les commandes `axis equal` pour que le rapport d'aspect de l'image soit réaliste, et `axis tight` pour éliminer les espaces blancs autour de l'image.

2. **Prise des points de mesure** A l'aide de l'outils d'étiquetage du menu de fenêtre graphique de Matlab, mesurer une vingtaine de points de la trajectoire tracée en rouge sur l'image. Vous mettrez ces points de mesure dans deux tableaux `x` et `y`.

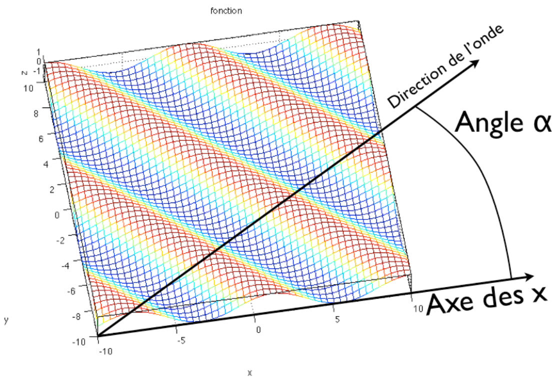
3. **Changement de référentiel:** Manipulez les deux tableaux `x` et `y` pour effectuer le changement de référentiel: centre du référentiel au point O sur

l'image et valeur de $x = 1$ au point représenté sur l'image.

4. **Affichage des points mesurés:** Dans deux sous-graphiques: tracez à gauche l'image originale et à droite tracez les points de mesure dans le bon référentiel.

1.3 Surface mouvante

Compétences: *tracer une fonction de deux variables avec la fonction* `mesh`, *réaliser une animation du mouvement de cette surface, avec la fonction* `drawnow`, *utiliser les opérations de tableaux élément par élément.*



1. **Grille:** Construire une grille cartésienne avec la fonction `meshgrid` pour $x \in [-L, L], y \in [-L, L]$, avec $L = 10$.

2. **Affichage:** Tracez avec la fonction `mesh` la fonction

$$f(x, y) = \cos(s), \quad \text{avec } s = \cos(\alpha)x + \sin(\alpha)y,$$

avec le paramètre $\alpha = \pi/8$; il s'agit d'une oscillation sinusoïdale dans la direction qui fait un angle α avec l'axe des x.

3. **Enveloppe:** Maintenant, on modifie un peu la fonction, tracez

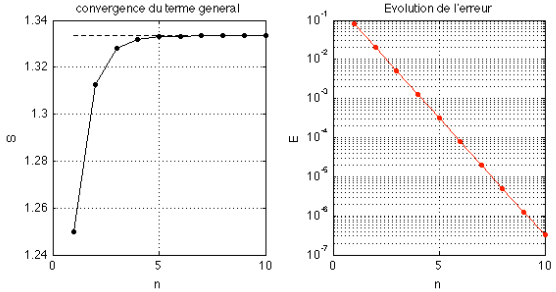
$$g(x, y) = \exp(-x^2/10)\cos(s), \quad \text{avec } s = \cos(\alpha)x + \sin(\alpha)y,$$

encore avec $\alpha = \pi/8$. On a rajouté une enveloppe sous la forme d'une Gaussienne dans la direction x.

4. **Animation:** Ecrivez maintenant les lignes de code qui font l'animation de la fonction (choisissez f ou g selon votre préférence), lorsque l'angle α varie de 0 à 2π, c'est à dire pour un tour complet de la direction de l'onde. Vous imposerez les limites des axes avec les fonction `xlim, ylim, zlim` pour que le cadre de la figure reste constant. Utilisez également la fonction `drawnow` pour que le graphique soit affiché pendant la boucle.

1.4 Convergence d'une série

Compétences: *Etablir une boucle qui calcule l'évolution du terme d'une série, et mémorisation dans un tableau de l'évolution de la somme partielle*



On considère la série

$$\frac{4}{3} = \frac{1}{4^0} + \frac{1}{4^2} + \frac{1}{4^3} + \dots$$

Il s'agit d'une des premières séries infinies dont on a prouvé la convergence. C'est le géomètre grec Archimède qui a démontré la convergence. Nous écrivons

$$S_n = \sum_0^n u_n, \quad u_n = \frac{1}{4^n}$$

et l'on appelle S_n la somme partielle, et u_n le terme général.

1. **Boucle:** Codez une boucle `for` qui calcule la valeur de la somme partielle pour une valeur donnée de n. On prendra $n = 10$.

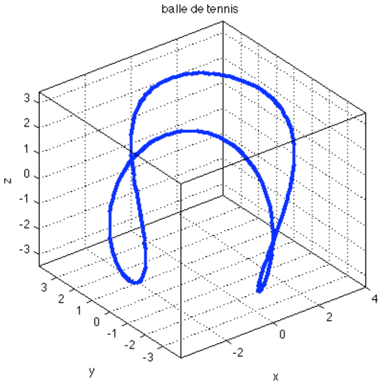
2. **Mémorisation:** Construisez un tableau nommé `smem` initialement rempli de zéros, et mémorisez dans ce tableau les valeurs de la somme partielle lorsque n augmente.

3. **Graphique:** Dans un sous-graphique, tracez en rouge, ligne continue avec des marqueurs `o` l'évolution de la somme partielle.

4. **Ligne horizontale:** Sur le graphique de la question précédente, ajouter avec la fonction `plot` une ligne horizontale pour $y = 4/3$, en ligne noire pointillée.

5. **Erreur:** Dans un second sous-graphique, tracez l'évolution de l'erreur $S_n - 4/3$. Cette erreur doit normalement tendre vers zéro, puisque la série tend vers $4/3$. tracez tout-d'abord l'évolution de l'erreur avec la fonction `plot`, puis ensuite avec un graphique semi-logarithmique avec la fonction `semilogy`.

1.1 Couture de la balle de tennis



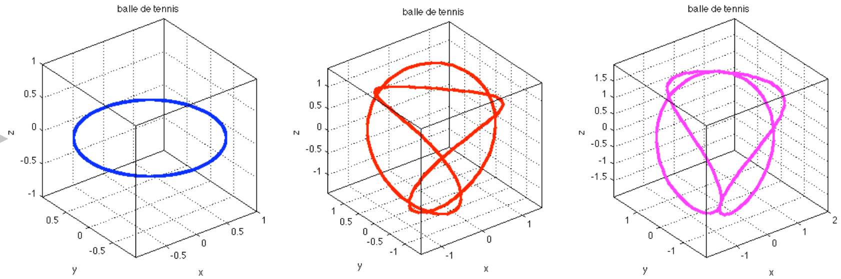
Script 1

```
% les paramètres
a=3; b=1;
c=2*sqrt(a*b);

% les équations
t=linspace(0,2*pi,200);
x=a*cos(t)+b*cos(3*t);
y=a*sin(t)-b*sin(3*t);
z=c*sin(2*t);

% on trace la figure
subplot(1,2,1)
plot3(x,y,z,'b','linewidth',3);
grid on; box on; axis equal
xlabel('x'); ylabel('y');zlabel('z');title('balle de tennis')
```

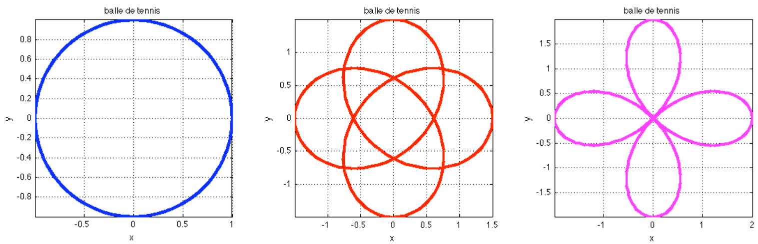
Script 2

```
% trois familles de paramètres

    %%%%%%%%%%%% courbe 1
    a=0; b=1; c=2*sqrt(a*b);

    x=a*cos(t)+b*cos(3*t);
    y=a*sin(t)-b*sin(3*t);
    z=c*sin(2*t);

    subplot(1,3,1)
    plot3(x,y,z,'b','linewidth',3);
    grid on; box on; axis equal
    xlabel('x'); ylabel('y');zlabel('z');title('balle de tennis')

    %%%%%%%%%%%% courbe 2
    a=0.5; b=1; c=2*sqrt(a*b);

    x=a*cos(t)+b*cos(3*t);
    y=a*sin(t)-b*sin(3*t);
    z=c*sin(2*t);

    subplot(1,3,2)
    plot3(x,y,z,'r','linewidth',3);
    grid on; box on; axis equal
    xlabel('x'); ylabel('y');zlabel('z');title('balle de tennis')

    %%%%%%%% courbe 3
    a=1; b=1; c=2*sqrt(a*b);

    x=a*cos(t)+b*cos(3*t);
    y=a*sin(t)-b*sin(3*t);
    z=c*sin(2*t);

    subplot(1,3,3)
    plot3(x,y,z,'m','linewidth',3);
    grid on; box on; axis equal
    xlabel('x'); ylabel('y');zlabel('z');title('balle de tennis')
```

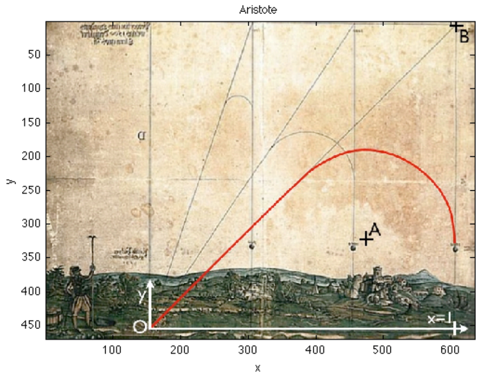
Vue par défaut

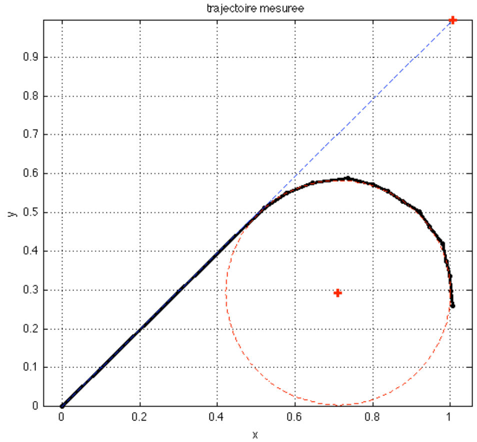


Pour la vue de dessus, plutôt que de changer le script avec la fonction view(90,90), on peut changer la vue avec l'outils de rotation 3D dans le menu de la fenêtre graphique, puis on sauve de nouveau la figure.



1.2 Balistique de Aristote






```
% trajectoire de aristote
  a=imread('aristote.jpg');
  subplot(1,2,1);
  image(a);
  axis equal
  axis tight
  xlabel('x'); ylabel('y'); title('Aristote');

  % mesure des points
x =[
  154.0979
  388.4632
  414.6075
  444.4867
  485.5707
  514.5162
  531.3233
  549.0641
  567.7386
  579.8770
  594.8166
  599.4853
  603.2202
  605.0876
  606.9551];
y =[
  454.9256
  224.2953
  207.4882
  195.3498
  189.7474
  197.2172
  205.6208
  216.8255
  228.9639
  246.7047
  266.3130
  286.8549
  304.5957
  321.4028
  338.2099];


pix=1/(605-155); % taille d'un pixel

% Changement de référentiel
x=x-x(1); y=y-y(1); % centre du referentiel
y=-y; % inversion des y
x=x*pix; y=y*pix; % mise a l'echelle

% on trace la trajectoire
plot(x,y,'k.-','linewidth',2)

axis equal; grid on
xlabel('x'); ylabel('y'); title('trajectoire mesuree');
```

Script

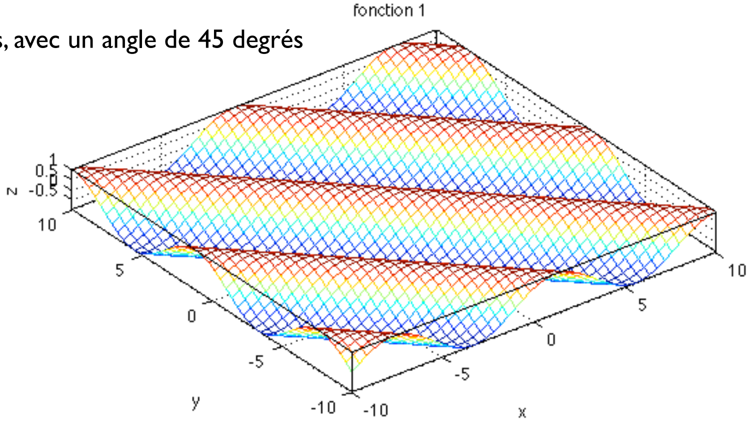
Ici quelques opérations en plus (pas demandées dans le sujet d'examen: on trace les points A et B, le segment de droite ainsi que le cercle qui correspondent à la théorie de l'impétus de Aristote.

Script+

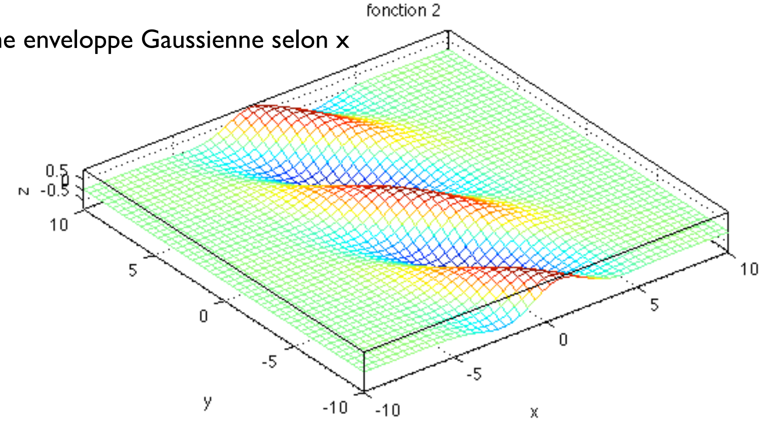
```
% les points A et B
ax=474; ay=323;
bx=607; by=7;

% manipulations:
ax=ax-x(1); ay=ay-y(1); bx=bx-x(1); by=by-y(1); % centre du ref
ax=ax*pix; ay=ay*pix; bx=bx*pix; by=by*pix; % echelle
ay=-ay; by=-by; % inverse les y

% on trace les points A et B
subplot(1,2,2);
plot(ax,ay,'r+',bx,by,'r+','linewidth',2)
hold on

% on trace la droite 0B
plot([0,bx],[0,by],'b--')

% on trace le cercle de centre A
t=linspace(0,2*pi,200);
R=1-0.7109; % le rayon du cercle
xx=ax+R*cos(t); yy=ay+R*sin(t);
plot(xx,yy,'r--')
```

1.3 Surface mouvante

Un cosinus, avec un angle de 45 degrés



Avec une enveloppe Gaussienne selon x



Script

```
n=50;
L=10;
x=linspace(-L,L,n);
y=linspace(-L,L,n);
[X,Y]=meshgrid(x,y);
a=pi/4;

% les fonctions en matrices
f1=cos(cos(a)*X+sin(a)*Y);
f2=exp(-(X.^2)/10).*cos(cos(a)*X+sin(a)*Y);


subplot(1,2,1);
mesh(X,Y,f1);
xlabel('x');ylabel('y');zlabel('z');title('fonction 1')
grid on; box on; axis equal

% on trace la seconde fonction
subplot(1,2,2);
mesh(X,Y,f2);
xlabel('x');ylabel('y');zlabel('z');title('fonction 2')
grid on; box on; axis equal
```

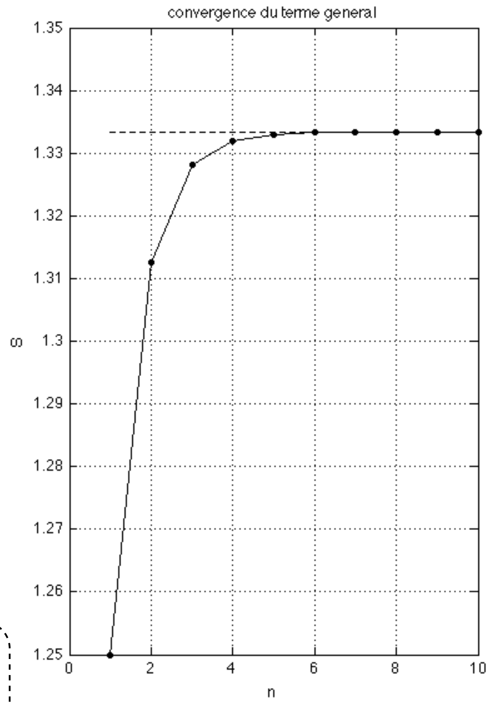
Il faut bien faire attention à utiliser les opérations élément par élément lorsque l'on multiplie la fonction en cosinus avec l'enveloppe en gaussienne.

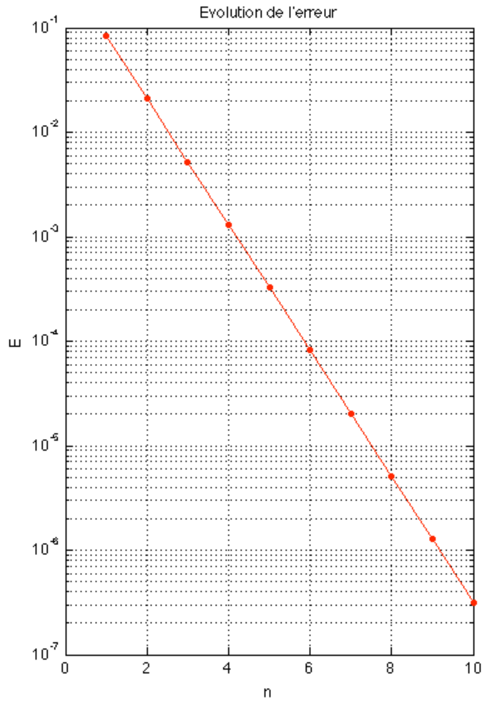
Pour l'animation:

Script

```
% le vecteur des angles successifs
   avec=linspace(0,2*pi,100);

   % boucle temporelle
   for ind=1:length(avec);
       a=avec(ind);

       % la fonction ? tracer
       f=cos(cos(a)*X+sin(a)*Y);

       % on trace le graphique
       mesh(X,Y,f);
       xlabel('x');ylabel('y');zlabel('z');title('fonction')
       grid on; box on; axis equal
       axis([-L L -L L -1 1])
       drawnow
   end
```

1.4 Convergence d'une série





Script

```
% paramètre n: nombre d'itérations
n=10;

% on construit un tableau pour mémoriser la somme partielle
S=zeros(n,1);

% la boucle, et on initialise la somme partielle
s=1;
for ind=1:n
    u=1/4^ind;
    s=u+s;
    S(ind)=s;
end

% on trace l'évolution de la somme partielle
subplot(1,2,1);
plot(1:n,S,'k.-',[1,n],[4/3,4/3],'k--')
grid on
xlabel('n');ylabel('S');title('convergence du terme general')

% on trace l'évolution de l'erreur
subplot(1,2,2);
semilogy(1:n,abs(S-4/3),'r.-')
grid on
xlabel('n');ylabel('E');title('Evolution de l''erreur');
```

Le terme général

La somme partielle

Mémorisation de la somme partielle