

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:

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

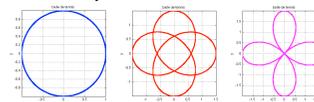
ou $t \in [0, 2\pi]$, et a, b , et c sont trois paramètres que nous pouvons faire varier pour changer l'allure de notre courbe

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

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

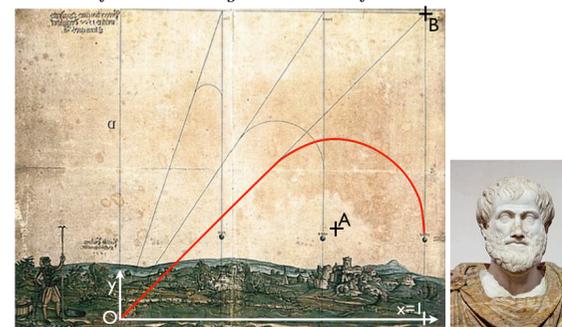
Ces courbes-ci ne correspondent en fait plus à la couture de la balle de tennis, mais montrent comment une forme peut changer lorsqu'on fait varier des paramètres numériques.

- 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.

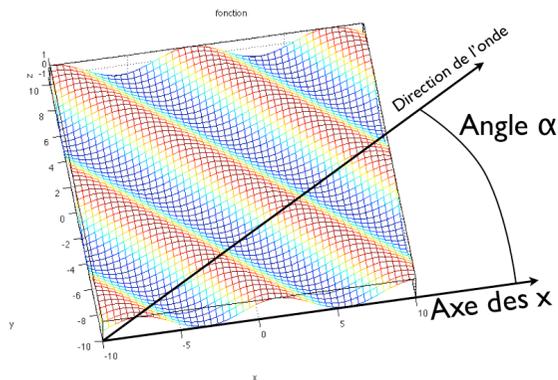
- Lecture de l'image:** Lisez l'image `aristote.jpg` sur le disque avec la fonction `imread` et affichez la avec la fonction `image`. Cette image doit être insérée dans le compte-rendu. Utilisez les commandes `axis equal` pour que le rapport d'aspect de l'image soit réaliste, et `axis tight` pour éliminer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

- 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.



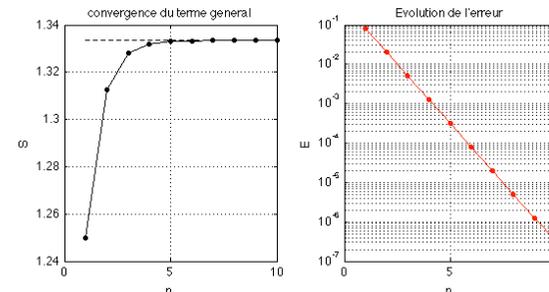
- Grille:** Construire une grille cartésienne avec la fonction `meshgrid` pour $x \in [-L, L], y \in [-L, L]$, avec $L = 10$.
- Affichage:** Tracez avec la fonction `mesh` la fonction

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

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

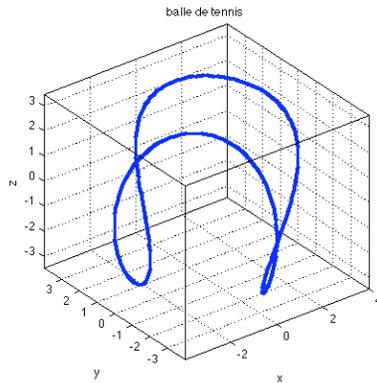
Il s'agit d'une des premières séries infinies dont on a prouvé la convergence. C'est le géomètre grec Archimède qui a démontré la convergence. Nous écrivons

$$S_n = \sum_{0}^n u_n, \quad u_n = \frac{1}{4^n}$$

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

- Boucle:** Codez une boucle `for` qui calcule la valeur de la somme partielle pour une valeur donnée de n . On prendra $n = 10$.
- Mémorisation:** Construisez un tableau nommé `smem` initialement rempli de zéros, et mémorisez dans ce tableau les valeurs de la somme partielle lorsque n augmente.
- Graphique:** Dans un sous-graphique, tracez en rouge, ligne continue avec des marqueurs `o` l'évolution de la somme partielle.
- 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.
- 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`.

I.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')
```

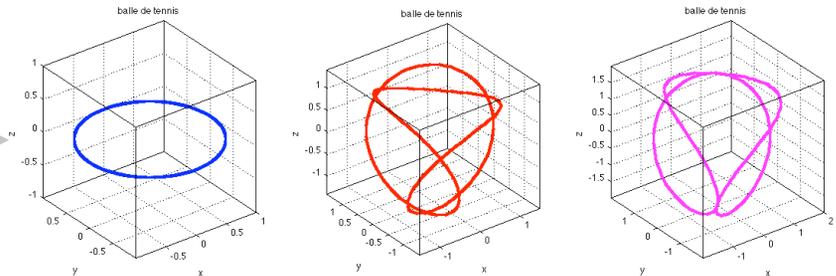
```
%%%%%%%%%%%%%% courbe 3
```

```
a=1; b=1; c=2*sqrt(a*b);
```

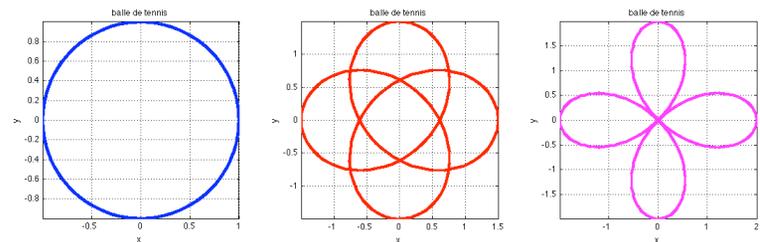
```
x=a*cos(t)+b*cos(3*t);
y=a*sin(t)-b*sin(3*t);
z=c*sin(2*t);
```

```
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')
```

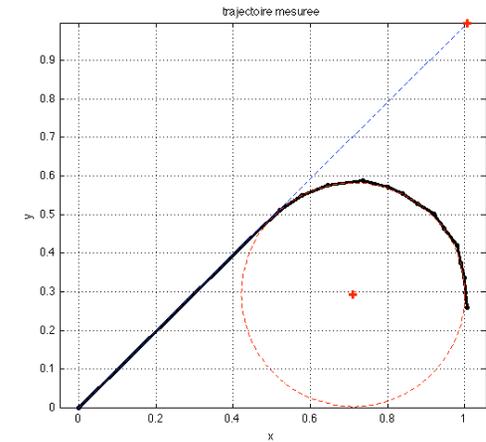
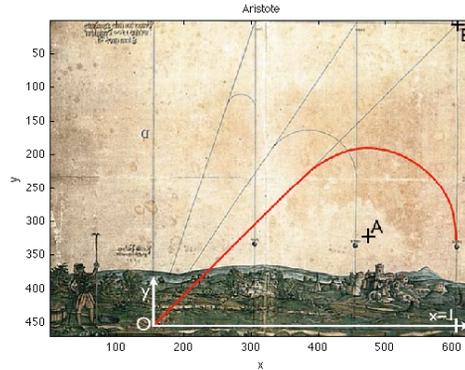
Vue par défaut



Pour la vue de dessus, plutôt que de changer le script avec la fonction `view(90,90)`, on peut changer la vue avec l'outil 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];
```

Script

```
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');
```

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.

```
% 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
```

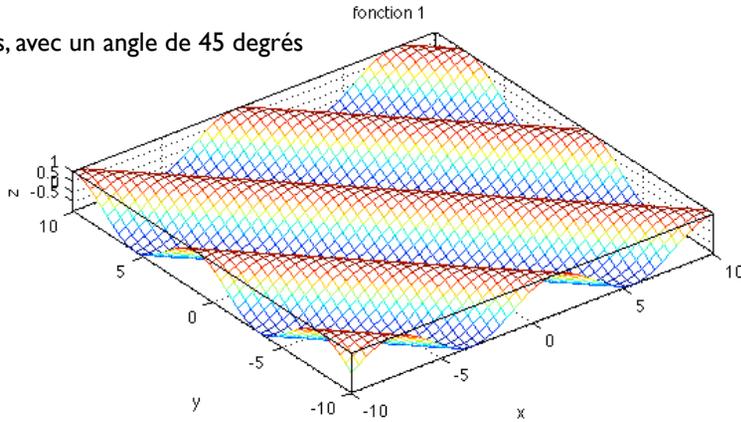
```
% on trace la droite OB
plot([0,bx],[0,by], 'b--')
```

```
% 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--')
```

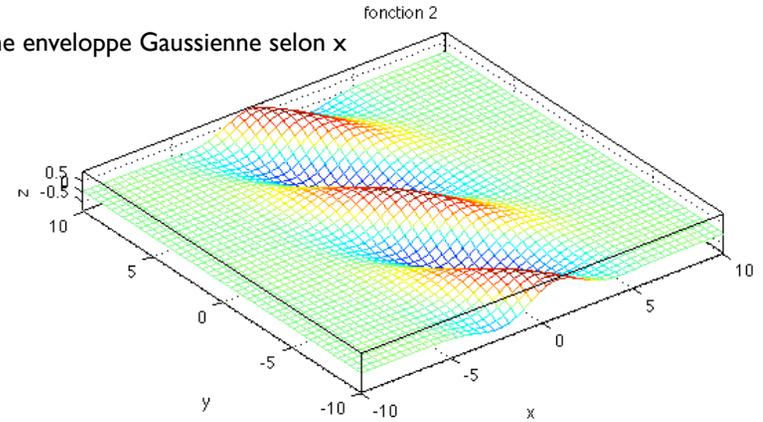
Script+

I.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
```

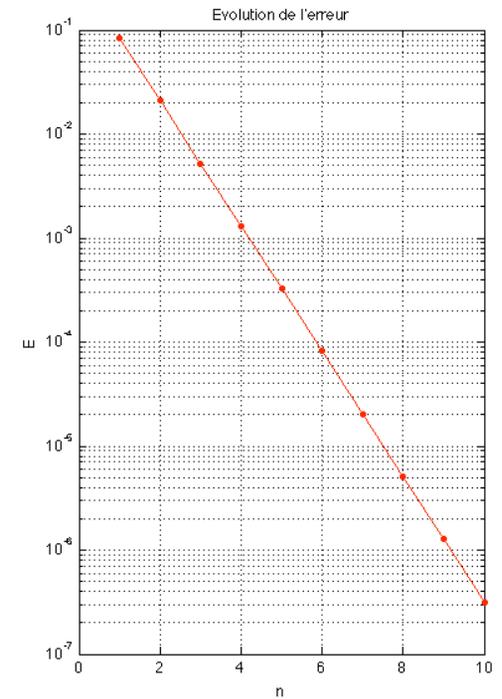
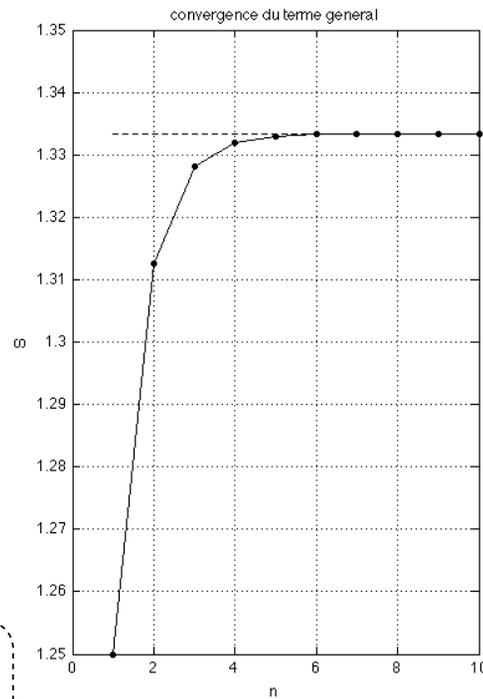
Il faut bien faire attention à utiliser les opérations élément par élément lorsque l'on multiplie la fonction en cosinus avec l'enveloppe en gaussienne.

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
```

I.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