

Matlab: applications en mécanique

LA207. Université Pierre et Marie Curie

www.lmm.jussieu.fr/~hoepffner/enseignement

Cours3: logiques et graphiques 3D

Ici un exemple d'utilisation des opérations logiques pour compter le nombre d'éléments qui satisfont à une inégalité. On l'utilise pour vérifier la propriété de convergence des probabilités lorsque le nombre d'éléments aléatoire est très grand.

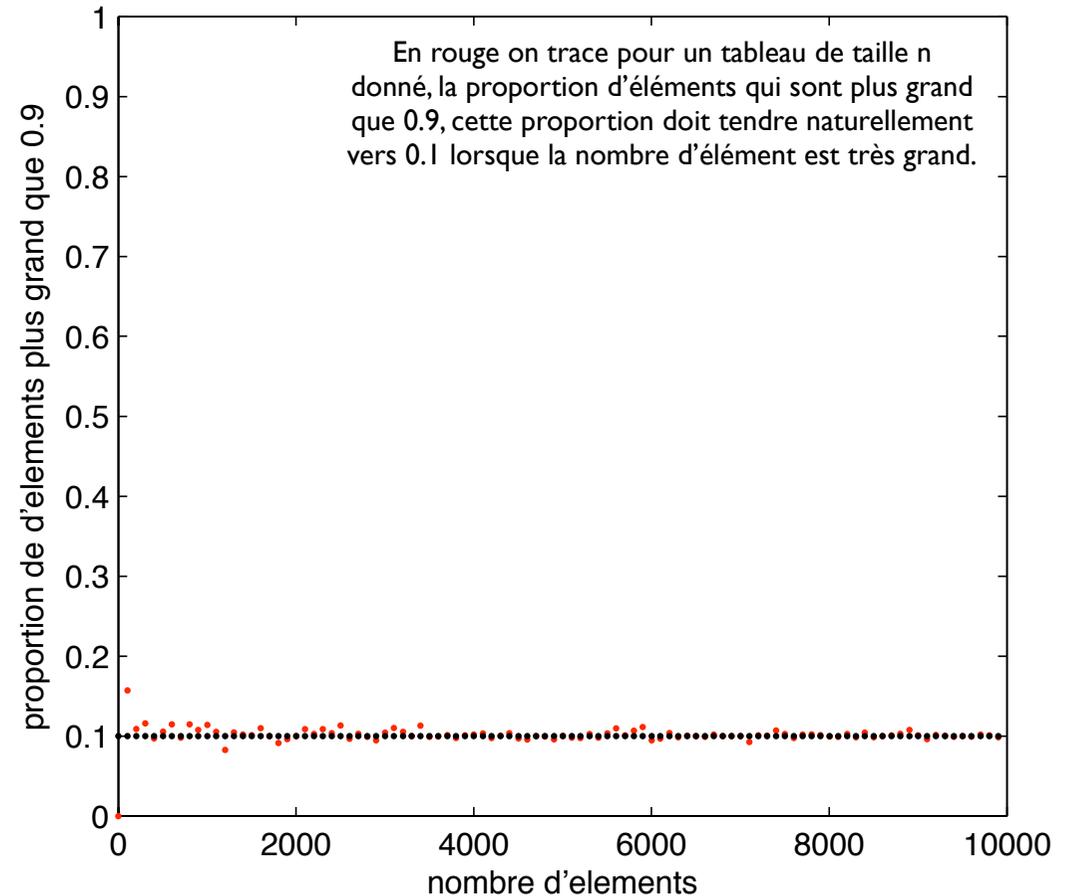
```
% etude statistique avec les operations logiques

% on fait varier la taille du tableau aléatoire
for n=2:100:10000;
% on construit le tableau aleatoire
a=rand(n,1);

% selection des elements plus grand que 0.9
s=a>0.9;

% on compte ces éléments
c=sum(s)/n;

% on trace
plot(n,c,'r.',n,0.1,'k. ');
hold on
xlabel('nombre d'elements');
ylabel('proportion de d'elements plus grand que 0.9');
ylim([0,1]);
end
```



Ici un autre exemple, on utilise encore une fois une inégalité pour sélectionner tous les pixels d'une image qui sont plus sombres ou plus clairs qu'une limite donnée. A partir de cette sélection, on va binariser l'image: plutôt qu'un gradient de tons de gris, on va avoir deux couleurs. C'est une opération très utile pour le traitement d'images scientifiques.

```
% binarisation de l'image "sable.png"

% on lit l'image et on l'affiche
subplot(2,1,1);
a=imread('sable.png');
image(a); axis equal tight;
title('image originale')

subplot(2,1,2);
lim=120;

% on ne prend qu'une seule couleur
% pour avoir une image en noir et blanc
b=a(:,:,1);

% on binarise
b(b>lim)=200;
b(b<lim)=1;

% on affiche l'image
imagesc(b);
axis equal tight
colormap gray
xlabel('x');
ylabel('y');
title('image binarisee')
drawnow
```

image originale

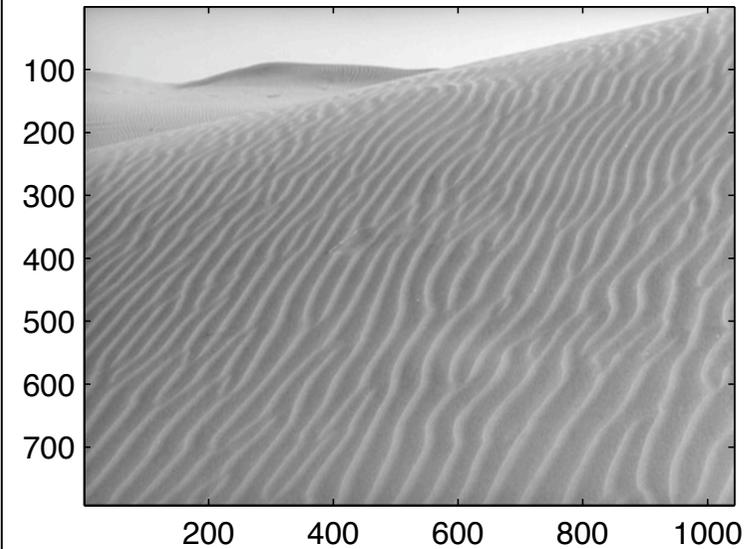
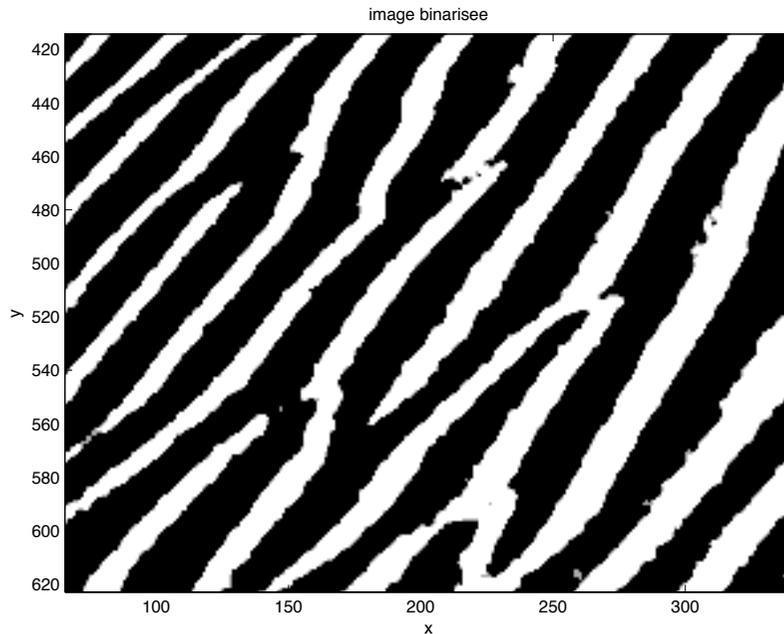
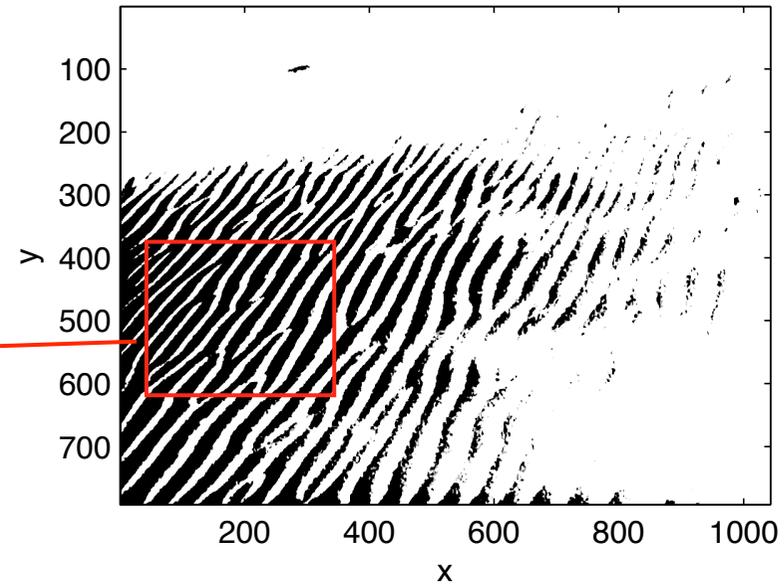
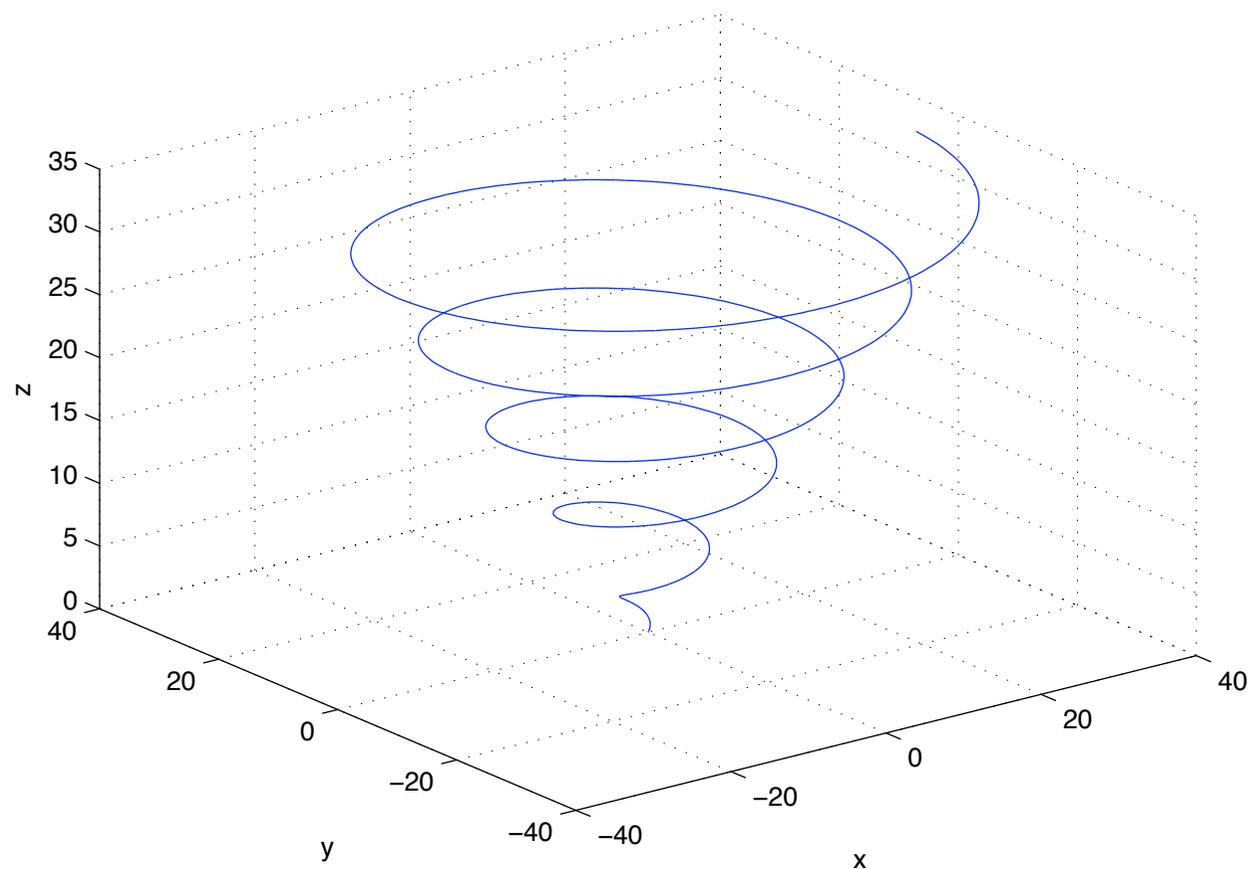


image binarisee



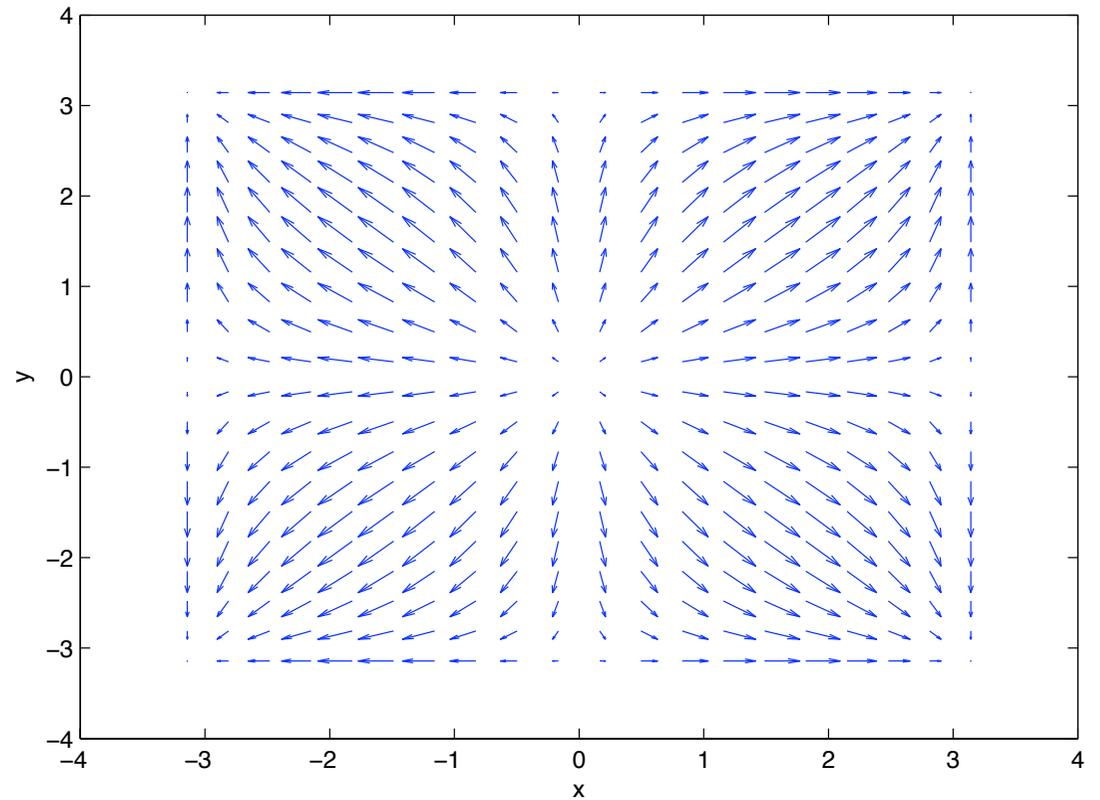
Un exemple de tracé de lignes en 3 dimensions avec la fonction plot3. Ici une spirale Archimédienne sous la forme d'une vis avec la coordonnées z en plus.

```
% la fonction plot3  
  
theta=linspace(0,5*2*pi,1000);  
r=1.1*theta;  
  
% une animation  
for t=linspace(0,6*pi,400);  
  
% les coordonnées x et y  
x=r.*cos(theta+t);  
y=r.*sin(theta+t);  
z=r; % l'altitude  
  
% on trace  
plot3(x,y,z,'b-');  
xlabel('x');  
ylabel('y');  
zlabel('z');  
grid on  
drawnow  
  
end
```



Tracé d'un champ de vecteurs avec la fonction quiver. ici comme pour les exemples suivants avec les fonctions mesh et surf, on utilise la fonction meshgrid pour générer les tableaux de coordonnées.

```
% on trace un champ de vecteur avec quiver
n=20;
% definition des coordonnées
x=linspace(-pi,pi,n);
y=linspace(-pi,pi,n);
% on construit les deux tableaux de coordonnées avec meshgrid
[X,Y]=meshgrid(x,y);
% une boucle pour une animation
for t=linspace(0,2*pi,200);
u=sin(X+t); v=sin(Y);
% on trace le champ de vecteurs
quiver(X,Y,u,v);
xlabel('x');
ylabel('y');
drawnow;
end
```



Tracé d'une fonction de deux variables avec la fonction surf. On utilise shading interp pour avoir une distribution interpolée des couleurs.

```
% une surface avec surf
n=50;
x=linspace(-pi,pi,n);
y=linspace(-pi,pi,n);
[X,Y]=meshgrid(x,y);

% animation dans le temps
for t=linspace(0,6*pi,100);

% la formule pour f
f=exp(-1*(X.^2+Y.^2)).*sin(2*X+t);

% on trace avec surf
surf(X,Y,f);
shading interp

xlabel('x');
ylabel('y');
zlabel('z');

% on impose les limites z et couleur
zlim([-1,1]);
caxis([-1,1]);
drawnow
end
```

