Révision Matlab, ce qu’il faut savoir !

>> a=2+5j

a =

2.0000 + 5.0000i

>> abs(a)

ans =

5.3852

>> angle(a) //réponse en radian

ans =

1.1903

>> angle(a)\*180/pi //réponse en degré

ans =

68.1986

5 cjs (25) = 5 cos (25)\* j sin (25) seulement on need radian !

>> a=25\*pi/180

a =

0.4363

>> 5\*(cos(a)+(j\*(sin(a))))

ans =

4.5315 + 2.1131i

a=[1,2,3] 1 ligne, 3 colonnes

b=[1 ;2 ;3] 3 lignes, 1 colonnes

Concaténation de vecteur :

a= [ b, c]

a=transpose(a) permet de changer les lignes en colonnes et inversement.

Lancer un programme C

!nomprogramme & (mettre le programme dans la racine du dossier travail matlab)

M=[1 2;3 4];

Grace à « : » on peut voir une ligne entière ou une colonne entière

>> M(1, :)

ans =

1 2

>> M(2,:)

ans =

3 4

>> M(:,1)

ans =

1

3

>> M(:,2)

ans =

2

4

M= ones(1)

M =

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

>> M(:,2)=2 (tout les lignes de la deuxième colonne deviennent 2 )

M =

1 2 1 1

1 2 1 1

1 2 1 1

1 2 1 1

>> M(3,:)=3 (toute les colonnes de la 3eme ligne deviennent 3 )

M =

1 2 1 1

1 2 1 1

3 3 3 3

1 2 1 1

Pour elever chaque terme au carré, utiliser :

M= M.^2

V= 10:20

V =

10 11 12 13 14 15 16 17 18 19 20

V=V.^2 // on utilise donc pas de boucle !!

V =

100 121 144 169 196 225 256 289 324 361 400

clear all;

close all;

clc;

t=0:0.1:2;

y=0.5\*sin(t);

subplot(2,1,1)

plot(t,y)

subplot(2,1,2)

stem(t,y) permet de dessiner le graph sans relier les points !

Trouver les 10 premieres periodes :

clear all;

close all;

clc;

A=0.5;

fs=22000;

f=440;

t=0:1/fs:4;

y= A\*sin(2\*pi\*f\*t);

%sound(y, fs)

dixperiode =((1/f)\*10\*fs);

y2=y(1:dixperiode)

plot(y2)

Explication : 1/f = une période, il nous en faut 10 ! par seconde, il y a 22000 échantillons, donc nous devons faire une multiplication par fs pour connaitre le nombre d’échantillons pendant ces 10 périodes !

Specgram(signal) affiche evolution du spectre dans le temps

duree=2;

t=0:1/Fs:duree;

y=A\*sin(2\*pi\*F\*t);

% pour 0 a FS

% f=0:1/duree:Fs;

% u=abs(fft(y));

% plot(f,u);

f= -Fs/2 : 1/duree : Fs/2;

u=abs(fftshift(fft(y)));

plot(f,u);

[son,Fs]=wavread('C:\Documents and Settings\kev\Bureau\ephec\3eme\traitement de signal labo\Révision\tarzan.wav');

fc=1000;

wn= (fc/(Fs/2)) fréquence de coupure divisé par fréquence échantillon/2

[b,a]=butter(2,wn); 2 correspond a l’ordre du filtre

S=filter(b,a,son);

im= imread('C:\Documents and Settings\kev\Bureau\ephec\3eme\traitement de signal labo\Révision\pout.tif');

im2=(double(im)/255);

mesh(im2) affiche l’image en « 3D » selon le niveau de gris

im2=im(:,:,1); /composante rouge

im3=im(:,:,2); /composante vert

im4=im(:,:,3); /composante bleu

subplot(2,2,1);

imshow(im);

subplot(2,2,2);

imshow(im2);

subplot(2,2,3);

imshow(im3);

subplot(2,2,4);

imshow(im4);

R(:,:,1)= ones(400);

R(:,:,2)=zeros(400);

R(:,:,3)=zeros(400);

subplot(2,4,1);

imshow(R);

title('rouge');

pour la couleur d’une image, ouvrir l’image avec un imread, puis faire un improfile : regarder les composantes de l’image (attention de ne pas melanger les couleurs). On va diviser ces composantes car l’image que l’on crée est composé de valeur de 0 a 1 et non de 0 a 255.

fleur(:,:,1)=(55/255) \*ones(400);

fleur(:,:,2)=(70/255) \*ones(400);

fleur(:,:,3)=(125/255) \*ones(400);

subplot(2,4,8);

imshow(fleur);

title('couleurs fleurs');

voir rgb2gray et luminance dans cours

correct = im2double(img);

im2double permet de convertir l’image en double et ne pas devoir diviser par 255 apres !

clear all;

close all;

clc;

img= imread('LENNA.BMP');

for ii= 10 : 10 : 100 img(ii,ii) = 0; end;

sav=img;

img=im2double(img);

for ii=1:256

for jj=1:256

if img(ii,jj)==0

img(ii,jj) = ((img(ii-1,jj)+img(ii-1,jj+1)+img(ii-1,jj-1)+img(ii+1,jj)+img(ii+1,jj+1)+img(ii+1,jj-1)+img(ii,jj+1)+img(ii,jj-1))/8);

end;

end;

end;

methode par fonction :

sortie trouble et moche !

h=ones(3)/9;

img=imfilter(img,h);

h=ones(5)/25;

img=imfilter(img,h); // encore plus trouble, mais points presque disparu

img=im2bw(img, graythresh(img)); % graythresh pour le niveau logique de conversion vers binaire (im2bw)

[L,NUM] = BWLABELN(img) ; %L = matrice avec 0 et 1111 22222 NUM = nombre d'element trouvé

stat=regionprops(L);

stat.Area…

img = imread('circuit1.tif');

level = graythresh(img);

bw = im2bw(img,level);

[L, num]= bwlabeln(bw); % pas assez precis !

S=regionprops(L);

test=ismember(L,find( [S.Area] < 4)); % on voit tout les points plus petit que 4 en area !

imshow(test);

final= bwareaopen(bw, 4); % remplace tout les points plus petit que 4

imshow(final)