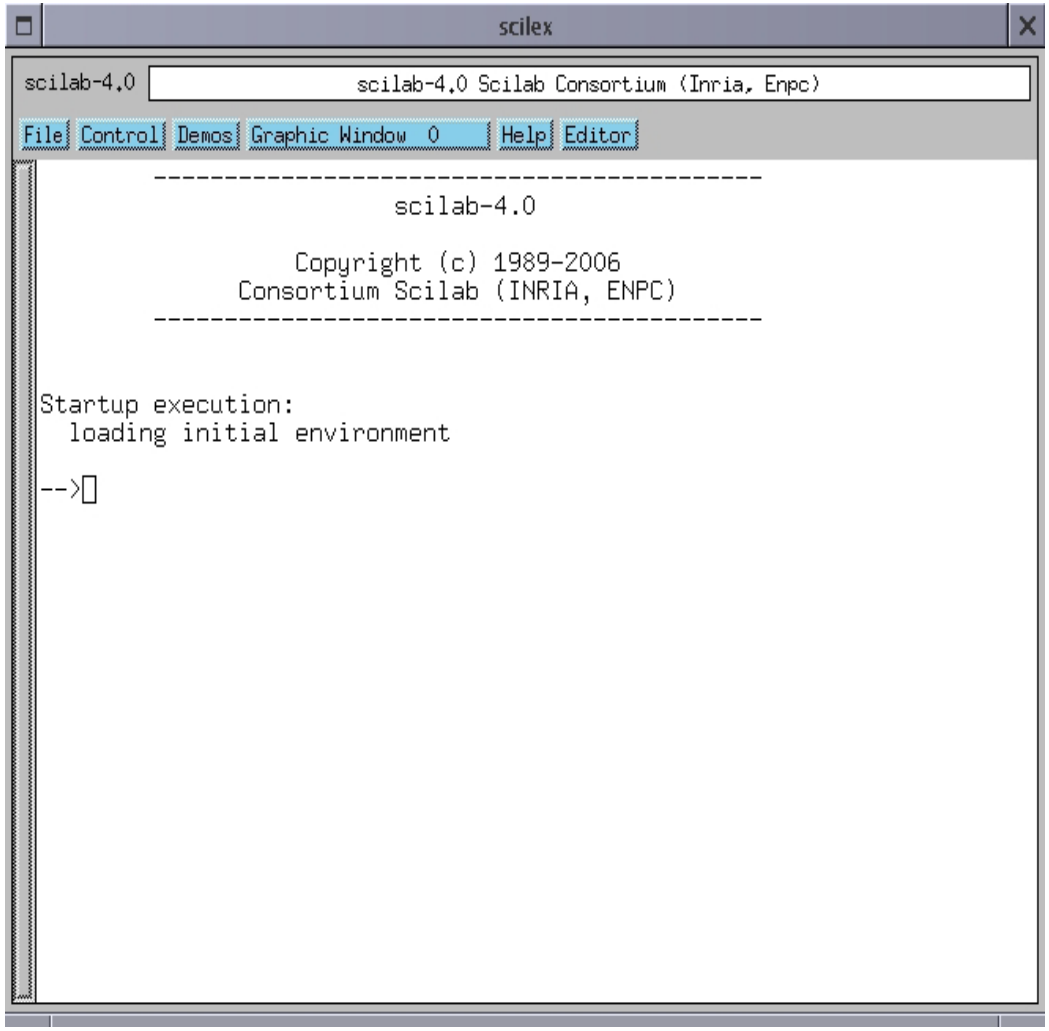
SCILAB

Introduction à Scilab

- Logiciel gratuit développé par l'INRIA
  - Windows,
  - Linux,
  - Mac.
- Dédié au calcul numérique
- Rénseignements: http ://www.scilab.org
- Langage de programmation basé essentiellement sur le calcul matriciel, avec des fonctionnalités mathématiques et graphiques étendues.

La Fenêtre Scilab



Les Fichiers Scilab

- Scilab utilise des fichiers texte.
- L’extension n’a à priori aucune importance.

Les Librairies

- Ce sont des fichiers du type file.sci. Ces fichiers sont dédiés à la définition des différentes fonctions.
- Pour être utilisés, ces fichiers doivent être compilés au préalable. Pour cela, on utilise la commande getf :
- getf(“fichier.sci”) ;
- On peut exécuter cette commande dans la fenêtre Scilab ou dans un script.

Les Scripts

- Ce sont des fichiers du type file.sce. Ils vont être exécutés par Scilab grâce à la commande exec :
- exec(“fichier.sce”) ;
- Ces fichiers contiennent une suite d’instructions.
- Ces instructions pourraient être exécutées directement dans la fenêtre Scilab .

L’Aide

- Hormis le menu help, il existe plusieurs moyens de trouver des informations :
  - La commande help() retourne des informations générales sur l’aide en ligne.
  - La commande help nom_fct effectue une recherche dans l’aide sur la commande nom_fct.
  - La commande apropos chaine effectue une recherche dans l’aide avec le mot-clé chaine.

Exemple

- help atan2
- ATAN2 Four quadrant inverse tangent.
  ATAN2(Y,X) is the four quadrant
  arctangent of the real parts of
  the elements of X and Y.
  -pi <= ATAN2(Y,X) <= pi.
  See also ATAN.

L’Espace de Travail

- Pour obtenir une liste des variables dans l'espace de travail:
  - who Affichage des variables dans l'espace de travail.
  - whos Affichage détaillé des variables dans l'espace de travail.

- Liste des constantes spéciales:
  - %nan NotANumber
  - %inf l’infini
  - %t le booléen true
  - %f le booléen false
  - %eps le zéro machine
  - %i le nombre imaginaire
  - %e le nombre exponentiel

LES TYPES DE DONNÉES

Nombres et Opérations Arithmétiques

+	Add
-	Soustr
*	Multipl
/	Divis
\	Division à
^	Puiss

Nombres Complexes

real	partie réelle
imag	partie imaginaire
conj	conjugué
abs	module
phasemag	argument (en degrés)

Polynômes

poly(v,"x")	polynôme dont les racines sont les éléments de v
poly(v,"x","c")	polynôme dont les coefficients sont les éléments de v
inv_coeff(v)	idem
coeff(P)	coefficients du polynôme P
roots(P)	racines du polynôme P
factors	facteurs irréductibles réels d'un polynôme
horner	évaluation d'un polynôme

Nombres et Fractions

derivat	procède à la dérivation du polynôme ou fraction.
lcm, gcm	plus petit et plus grand multiple commun.
pdiv, ldiv	division euclidienne et division par ordre croissant.
simp	simplification de fraction.
factors	factorisation en éléments simples.

Chaîne de Caractère

evstr	évaluer une expression
deff	définir une fonction
execstr	exécuter une instruction
length	longueur
part	extraire
+	concaténer
string	transformer en chaîne

Vecteurs

- Pour définir un vecteur
  - En donnant la liste des éléments:
  - u = [0.3 23 35] ;
  - En utilisant une fonction qui génère un vecteur:
- u = debut:pas:fin ;
  - En donnant la suite qui forme le vecteur:
- u = linspace(debut,fin,nbval) ;
- u = logspace(debut,fin,nbval) ;
  - (éléments répartis uniformément dans un intervalle)

Matrices

- Pour définir une matrice élément par élément:
  - Les crochets servent à encadrer la matrice ;
  - un espace ou une virgule sépare deux éléments d'une même ligne;
  - un point-virgule sépare les lignes entre elles.
- A=[1 3 4; 3 5 0.4; 4 7 0]
- B=eye(3)

Opérations Matricielles

symbole	définition
[]	définition matricielle et concaténation
;	séparateur de colonne
()	extraction/insertion d'un élément
'	transposition
+	addition
-	soustraction
*	produit matriciel
\	division à gauche
/	division à droite
^	puissance
.*	produit élément par élément
.\	division à gauche élément par élément
./	division à droite élément par élément
.^	puissance élément par élément

Fonctions

- Il est préférable de définir des fonctions dans des fichiers de librairies plutôt que dans les scripts.
- function [y1,...,yn]=nom fct(x1,...,xp)
  ...
  ...
  endfunction
- Pour définir une fonction en ligne de commande:
- deff(“[y]=nom fct(x1,x2)”,”y=cos(x1+x2)”);

Communication avec l'Usager

- Pour afficher un message ou une valeur à l'écran:
  disp('Ceci est un test')
- Afficher "Ceci est un test" sur l'écran
- Pour entrer une valeur:
  x = input('Valeur de x = ')
- Afficher sur l'écran "Valeur de x = "
- Attendre qu'un nombre soit tapé au clavier

La Boucle FOR

- Ces boucles sont définies de la manière suivante:
- for i = debut:pas:fin
- ...

...

end

- Elles peuvent être imbriquées

```
for i=1:100
     wt = 24*i*0.01
     x(i)=12.5*cos(wt+pi/6)
end
```

```
for i=1:5
     for j=1:20
          amp=i*1.2
          wt=j*0.05
          v(i,j)=amp*sin(wt)
     end
end
```

La Boucle FOR

- Il existe une autre manière d'écrire une boucle for dont l'indice de boucle parcourt les éléments d'un vecteur v :
-

```
for i=v
...
...
end
```

La Boucle WHILE

- Les boucles while s’écrivent de la façon suivante :
- while expression
  ...
  ...
  end

L’Instruction Conditionnelle IF

- Pour effectuer un test, on peut utiliser la combinaison classique if-then-else :
-
```
if condition
then ...
else ...
end
```

L’Instruction Conditionnelle SELECT

- On peut également avoir à utiliser :
-
```
select var
    case val1 then ...
    case val2 then ...
    case val3 then ...
    end
```

Opérateurs Logiques

==	égal à
~= ou <>	différent de
<	inférieur à
<=	inférieur ou égal à
>	supérieur à
>=	supérieur ou égal à
&	et
|	ou
~	non

ENTRÉES/SORTIES

Ouvrir un Fichier

- La commande la plus courante est :
- [fid,error] =
  mopen('fichier.xxx',mode);
- La variable mode peut être
  - "r" (lecture),
  - "w" (écriture),
  - "b" (binaire),
  - une combinaison des 3.

Lire dans un Fichier

- La commande la plus courante est :
- x = mfscanf(n,fid,format) ;
- n correspond aux nombre d'occurrences de données de type format à extraire. Si n = −1, la commande récupère le maximum de données possibles jusqu'à la fin du fichier.
- La variable format est une chaine de caractères constituée de caractères du type:
  - %s (caractère),
  - %d (entier),
  - %f (flottant),
  - ...

Ecrire dans un fichier

- La commande la plus courante est :
- x = mfprintf(fid,format,a1,...,an);

Fermer un Fichier

- mclose(fid) ;

GRAPHIQUES

Tracé en 2D

- plot2d (x,[y1 y2 ... yn], style=style, strf=chaine, leg=legende, rect=cadre) ;

  - les yi sont des vecteurs colonnes,
  - n désigne le nombre de courbes à tracer.
  - style est un vecteurs d'entiers relatifs de taille n.
    Si cet entier est strictement positif, alors il caractérise une ligne continue colorée, sinon une ligne de symboles noirs et blancs.
  - …

Tracé en 2D

- plot2d (x,[y1 y2 ... yn], style=style, strf=chaine, leg=legende, rect=cadre) ;

  ….

  - chaine est un argument du type ’xyz’
    - x=0 : pas de légende
    - x=1 : affiche la légende
    - y=0 : bornes courantes
    - y=1 : bornes définies dans cadre
    - y=2 : bornes ajustées
    - z=0 : pas d’entourage graphique
    - z=1 : axes gradués
    - z=2 : cadre autour du graphique
  - legende est une chaine de caractère du style ’leg1@leg2@... ’, qui contient la liste des légendes pour chacune des courbes.
  - cadre détermine les bornes des échelles graphiques. C’est un argument de la forme [xmin ymin xmax ymax].

Commandes plus spécifiques

- La commande fplot2d permet de tracer une courbe définie par une fonction. Ses arguments sont les mêmes que plot2d à ceci pr`es que la matrice des ordonnées est remplacée par le nom de la fonction.
- – Les commandes xpoly et xpolys permettent de tracer des lignes brisées et par extension des polygones. La seconde est notamment utilisée pour afficher des maillage.
- – D'autres fonctions : graypolarplot, grayplot, contour2d, . . .

Histogrammes

- La fonction plot2d3 permet de représenter des diagrammes en bâtons.
- La fonction histplot représente des histogrammes. Son premier paramètre peut être un entier (nombre de classes), auquel cas l'histogramme est régulier, ou un vecteur donnant les bornes des classes.

Tracés en 3D

- plot3d(x,y,z, theta=theta, alpha=alpha, leg=legende, flag=[mode, type, box], ebox=cadre) ;
  - x et y sont des vecteurs de tailles nx et ny. z est une matrice de taille (nx,ny).
  - theta et alpha sont des param`etres angulaires permettant de définir l'angle de vue. Les valeurs par défaut sont 45 et 35.
  - legende est une chaine de type 'xlabel@ylabel@zlabel'.
  - mode définit la couleur
    - mode=-n : la couleur n est choisie, mais la grille n'est pas tracée
    - mode=0 : seule la grille est tracée
    - mode=n : la couleur n est choisie et la grille est tracée
  - ….

Tracés en 3D

- plot3d(x,y,z, theta=theta, alpha=alpha, leg=legende, flag=[mode, type, box], ebox=cadre) ;
  - ….
  - type définit les param`etres d'échelle
    - type=0 : bornes courantes
    - type=1 : bornes définies avec cadre
    - type=2 : bornes ajustées
  - box définit l'entourage graphique
    - box=0 : pas d'entourage graphique
    - box=1 : pas d'entourage graphique
    - box=2 : seuls les axes derri`ere la surface sont affichés
    - box=3 : une boite et les légende sont affichées
    - box=4 : une boite, les légendes et les axes sont affichés
  - cadre détermine les bornes du graphique.
    C'est un argument de la forme [xmin xmax ymin ymax zmin zmax].

TRAITEMENT D'IMAGES

SIP

- Librairie Image de Scilab:
  Scilab Image Processing.
  http://siptoolbox.sourceforge.net

- SIP utilise ImageMagick pour transformer un fichier image en une matrice utilisable en Scilab.
- De nombreux formats sont reconnus parmi lesquels:
  JPEG, TIFF, BMP, PNG,...

Espace Mémoire

- L’espace mémoire occupé par les images étant très important, il est important d’augmenter la taille de la pile utilisée par Scilab en utilisant
- --> stacksize(3e7)
- (l’espace mémoire choisi dépend de votre pc).

Ouvrir une Image

- Pour lire une image en niveaux de gris ou en couleur
- --> mat=imread('ararauna.jpg');
- (si vous oubliez le point virgule, la matrice de l'image sera entièrement affichée à l'écran).
- La matrice appelée "mat" contient la matrice de l'image "ararauna.jpg".
- Tous les outil disponibles pour le traitement d'images en Scilab sont applicables.

Afficher une Image

- --> xbasc();imshow(mat);
- (xbasc() supprime le contenu de la fenêtre graphique)

Sauvegarder une Image

- --> imwrite(mat,‘path/test.jpg')

- Pour connaître l’emplacement courant, vous pouvez utiliser la commande pwd.
- De même pour connaître les documents présents dans le dossier courant, la commande ls fonctionne

Normaliser une Image

- Pour assombrir une image:
- --> dmat=mat*0.6;
- --> xset('window',2);xbasc();imshow(dmat);
- --> max(dmat) returns 0.6
- A présent pour récupérer une image comprise entre 0 et 1 (1 et 256) on peut utiliser l’instruction de normalisation:
- --> nmat=normal(dmat);

Coordonnées des Pixels

- La numérotation des coordonnées de l’image commencent en haut à gauche
- --> image=imread(‘path/ararauna.jpg');
- --> [r c]=size(image);

Exemple

- --> image(10,:,1)=1;
  (dessine une ligne rouge à la 10ième ligne)
- --> image(:,10,2)=1;
  (dessine une ligne rouge à la 10ième colonne)

Moyenne de 2 Images

- --> mat1=imread('image1.jpg');
- --> mat2=imread('image2.jpg');
- --> mat_mean=(mat1+mat2)/2;
- --> xbasc();
- --> imshow(mat_mean);

Multiplication de 2 Images

- La multiplication de deux images N'EST PAS EGALE à la multiplication des deux matrices.
- Il s'agit ici de multiplier le coefficient (i,j) de l'image 1 par le coefficient (i,j) de l'image 2 pour deux images de même taille.
- --> mat=imread('image.jpg');
- --> pmat=mat.*mat;
- A comparer avec:
- --> pmat=mat*mat;

Addition ou Soustraction MOD 256

- En utilisant la fonction imread, la matrice obtenue est codée en doubles compris entre 0 and 1.
- Si vous avez une image 8bit et voulez l'additionner ou la soustraire avec une autre image modulo 256 il faut écrire:
- --> mat1=imread('image1.jpg');
- --> mat2=imread('image2.jpg');
- --> mat_modulo=uint8(mat1*255)-uint8(mat2*255);
- --> mat_modulo=double(mat_modulo)/255;
- (pour éviter les problèmes par la suite, repassez l'image dans son format original)

Image en Niveaux de Gris

- Une image en niveaux de gris est un tableau 2D des valeurs des pixels. Pour lire cette image, vous pouvez utiliserla fonction gray_imread
- --> mat=gray_imread('onca.gif');
- --> xbasc();imshow(mat);
- --> size(mat)
- Cette image qu'enne soit en 16 bits (1 à 65535) ou en 8 bits (1 à 255), les valeurs de la matrice appartiennent à [0,1]. Si vous préférez travailler dans [0, 255], vous pouvez multiplier cette matrice par ce facteur.

Images en Couleurs

- Pour les images en couleur, on a 3 niveaux, un pour chaque couleur (R,V,B)
- --> mat=imread('ararauna.jpg');
- --> xbasc();imshow(mat);
- --> size(mat)
- mat(:,:,1) est la composante Rouge
- mat(:,:,2) est la composante Verte
- mat(:,:,3) est la composante Bleue

Conversion en Niveaux de Gris

- --> mat=imread('ararauna.jpg');
- --> gmat=im2gray(mat);
- --> xset('window',1);xbasc();imshow(gmat);
- Ou directement:
- --> gmat=gray_imread('ararauna.jpg');
- Note:
  typeof(mat) est ’hypermat’ (color)
  typeof(gmat) est constant(grayslevels).

Seuillage

- Pour seuiller l'image (la matrice aura 2 valeurs: 0 ou 1)
- --> bmat=im2bw(gmat,0.7);
- --> imshow(bmat);

Original image	Threshold 70%	Threshold 50%
		

Convolution (Rappel)

- Supposons que l'on veuille convoluer le voisinage de pixel suivant avec un filtre moyenneur:
1	5	1
2	4	3
8	2	1

inage

1	1	1
1	1	1
1	1	1

re
- (1x1)+(1x5)+(1x1)+(1x2)+(1x4)+(1x3)+(1x8)+(1x2)+(1x1)= 27.
  on divise par 9 (somme des coefficients du filtre)
  la valeur filtrée du pixel central est 3.
- Cette opération est répétée pour chaque pixel de l'image.

Convolution (Rappel)

- On dispose de la matrice ”mat” des pixels de l’image. On effectue le produit de convolution comme suit:
- --> kern=[1 1 1;1 1 1;1 1 1]
- --> mat_conv=(1/9)*imconv(mat,kern);
- --> xbasc(); imshow(mat_conv);

Filtres Pré-Définis

- Cependant, inutile de réinventer la roue à chaque fois, il existe un certain nombre de filtres pré-définis.
- --> mkfilter('mean')
- --> mat_conv=imconv(mat,mkfilter('mean'));
- --> xbasc(); imshow(mat_conv);

Quelques Filtres

- ”mean”

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- ”low-pass”

1/12	1/12	1/12
1/12	4/12	1/12
1/12	1/12	1/12

- ”laplace1”

0	-1	0
-1	4	-1
0	-1	0

- ”laplace2”

-1	-1	-1
-1	8	-1
-1	-1	-1

- ”laplace3”

1	-2	1
-2	4	-2
1	-2	1

Exemple de Filtrage

Original image	Mean filtering	Laplace1 filtering
		

Filtre Médian

- Le filtre médian consiste à trier les valeurs des pixels et à utiliser la valeur centrale. Dans cet exemple, la valeur utilisée est en 5ième position, il s'agit de la valeur 2.

1	5	1
2	4	3
8	2	1

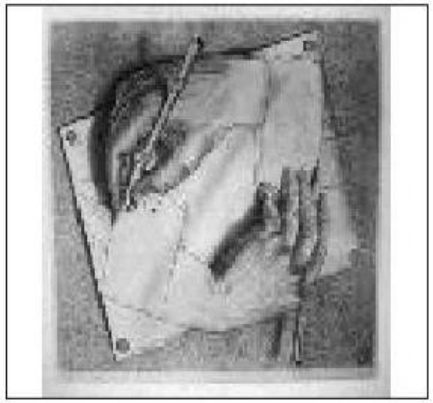
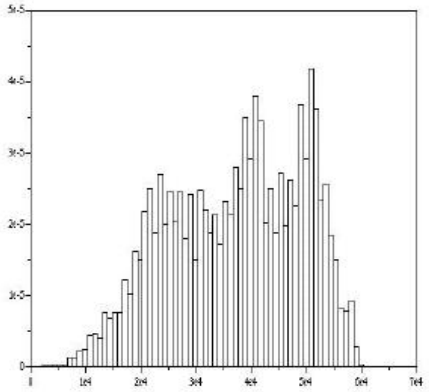
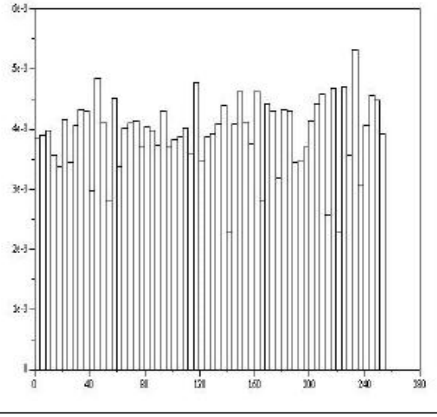
1 1 1 2 2 3 4 5 8

- Il s'agit d'un filtre passe-bas (il supprime les hautes fréquences, supprime le bruit en effectuant un lissage de l'image)
- --> mmat=mogrify(mat,'-median 1');.

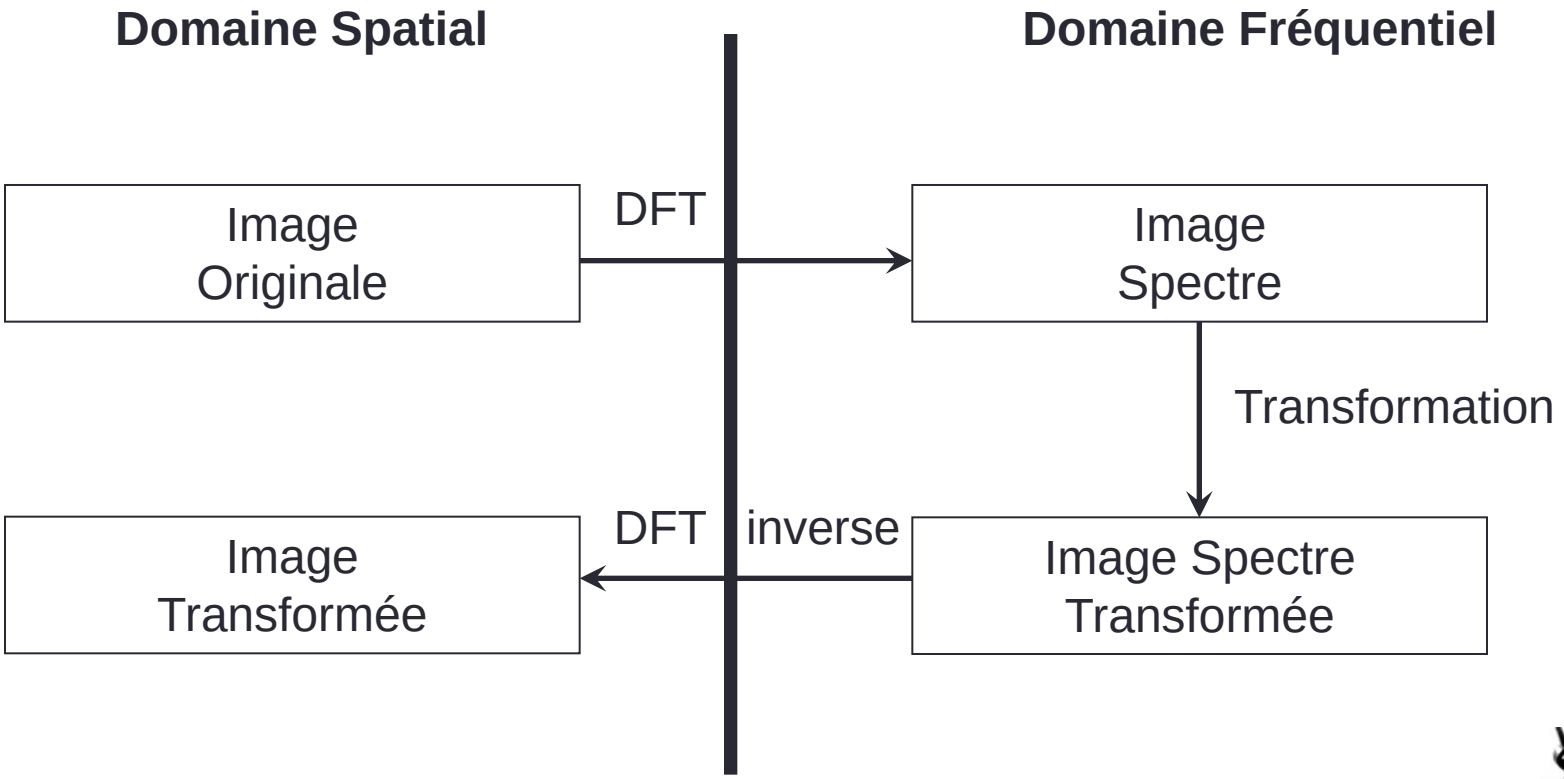
Histogramme

- Cette opération est souvent utilisée pour rehausser ou modifier le contraste de l'image:
- --> image=gray_imread('gra.jpg');
- --> xset('window',0);imshow(image);
- --> fimage=mogrify(image,['-equalize']);
- --> xset('window',1);imshow(fimage);
- Pour voir l'histogramme et les modifications effectuées:
- --> xset('window',2);histplot(64,image);
- --> xset('window',3);histplot(64,fimage);

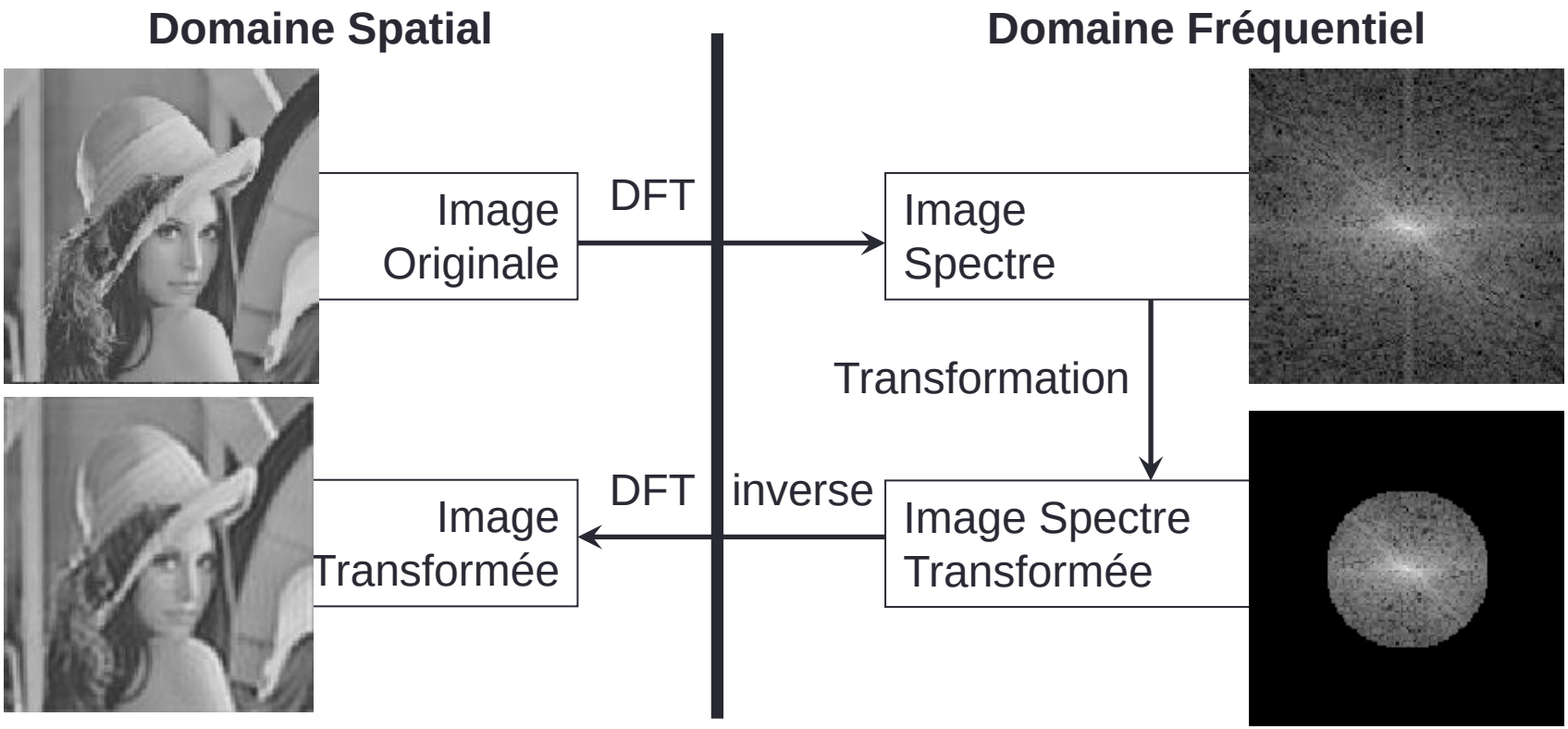
Exemple

Original image	Histogram
	
Filtered image	**Histogram**
	

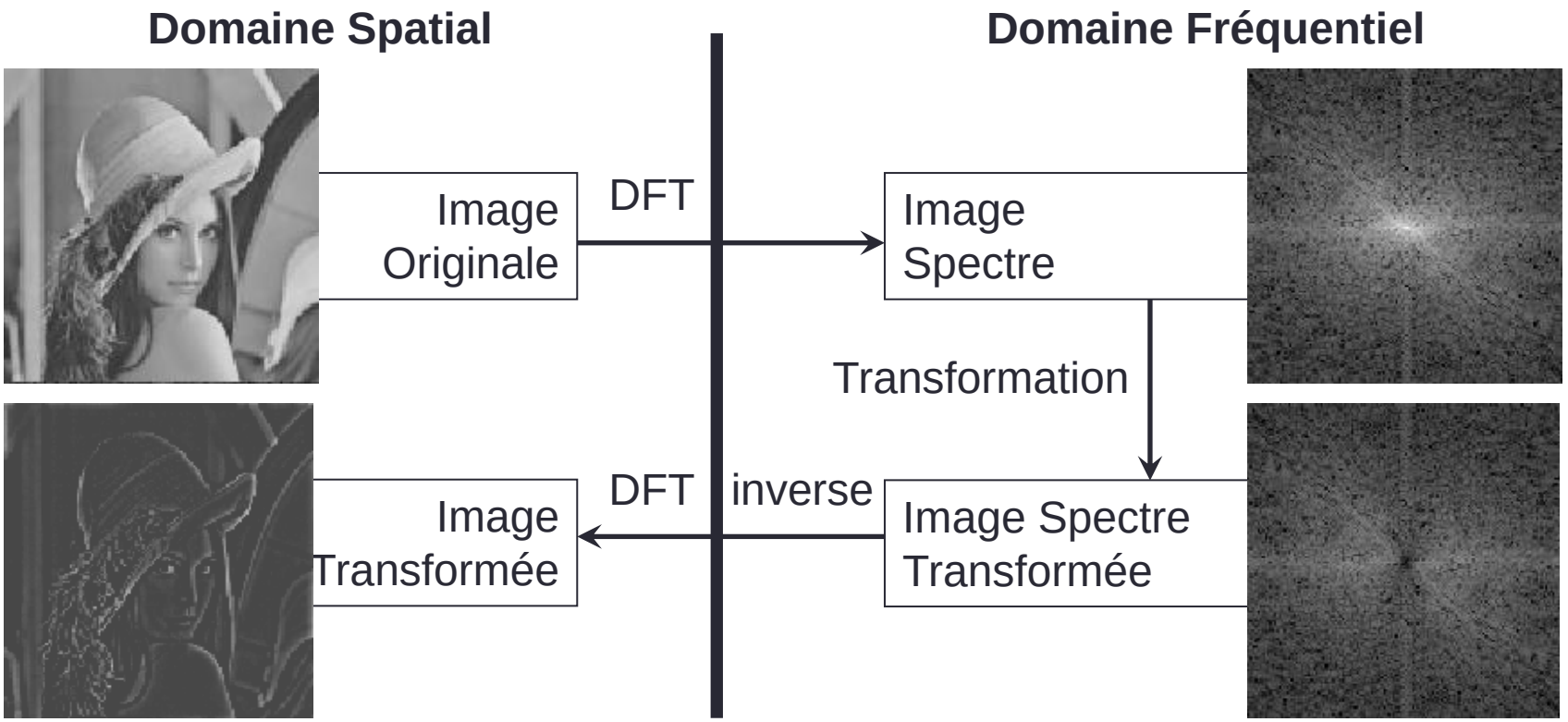
Opération dans l'espace de Fourier



Filtrage Passe bas

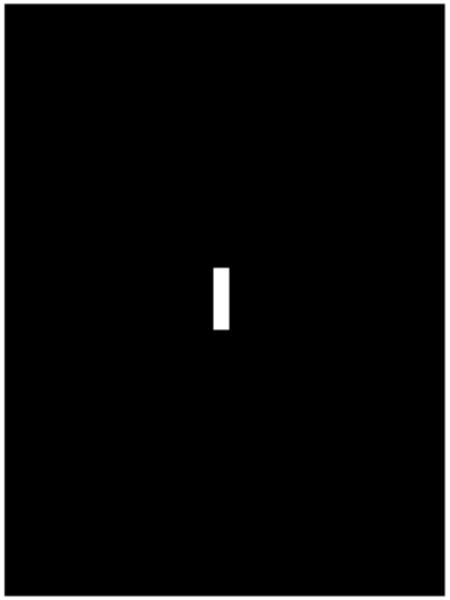


Filtrage Passe haut



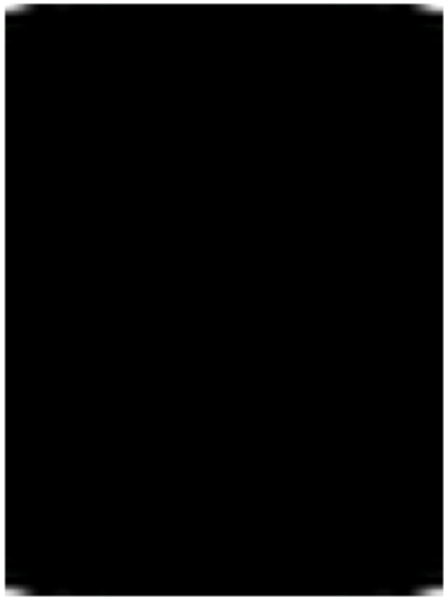
Transformée de Fourier

- --> image=zeros(400,300);
- --> image(180:220,145:155)=1;
- --> xset("window",0);xbasc();imshow(image);



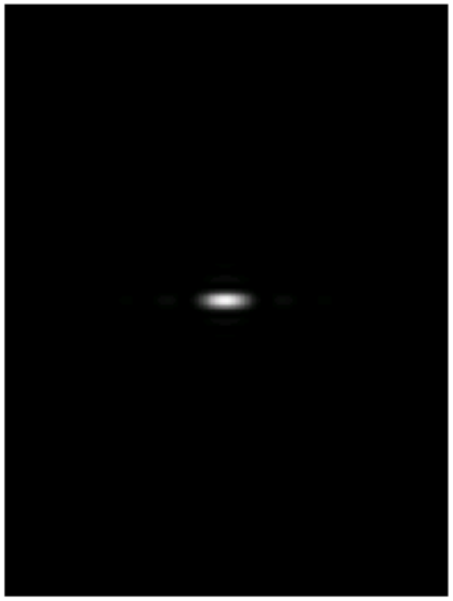
Transformée de Fourier

- --> IM=fft(image,-1);
- --> spectrum=real((IM).*conj(IM));
- --> xset("window",1);xbasc();imshow(spectrum,[]);



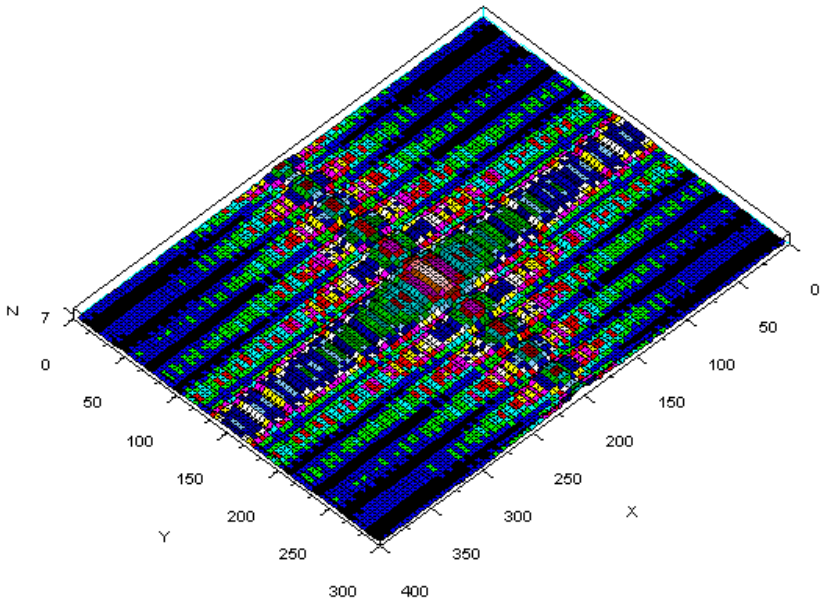
Transformée de Fourier

- --> spectrum2=sip_fftshift(spectrum);
- --> xset("window",2);xbasc();imshow(spectrum2,[]);
- Cette image montre un pic central très important.



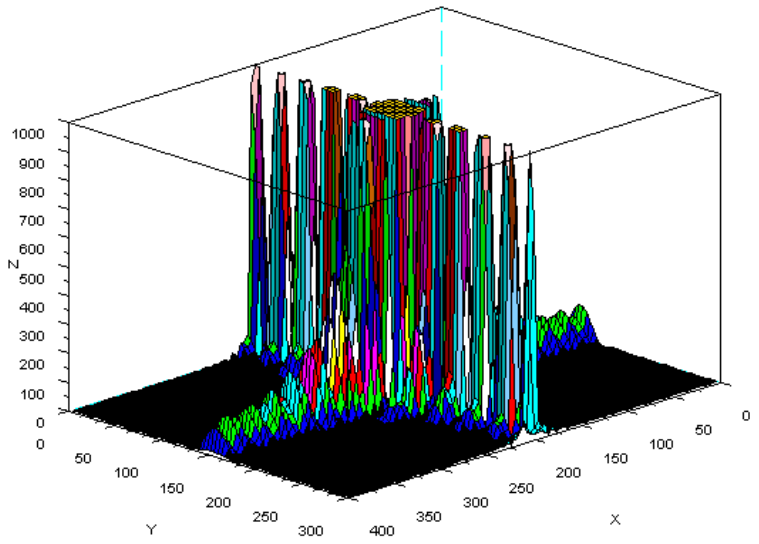
Transformée de Fourier

- Une meilleure visualisation peut être obtenue par 2 méthodes:
- --> spectrum3=log(spectrum2+1);
- --> xset("window",3);xbasc();
- -->plot3d1(1:4:400,1:4:300, spectrum3(1:4:400,1:4:300));



En 3D

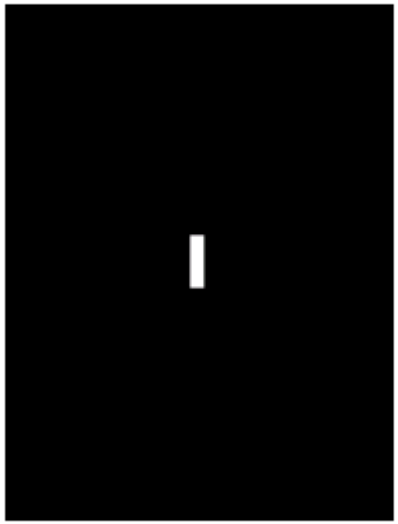
- --> spectrum3=spectrum2;threshold=1e3;
- --> spectrum3(find(spectrum3>threshold))=threshold;
- --> xset("window",3);xbasc();
- --> plot3d1(1:4:400,1:4:300, spectrum3(1:4:400,1:4:300));

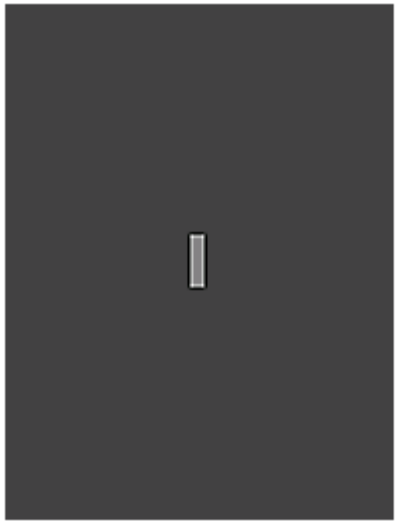


Transformée de Fourier

- Avec un filtre passe bas et un filtre de netteté

- -->image=imconv(image,mkfilter('mean'));
- -->image=imconv(image,mkfilter('sh2'));





Transformée de Fourier

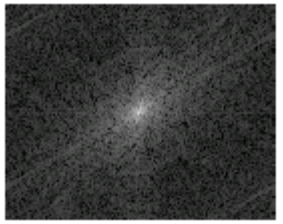
- L’image originale est restituée par
- --> im2=real(fft(IM,1));
- Avant de restituer l’image, on peut appliquer un filtre sur le spectre. Le plus simple est un filtre binaire.

Transformée de Fourier

- --> image=gray_imread(SIPDIR+'images/ararauna.jpg');
- --> [r,c]=size(image);
- --> xset("window",0);xbasc();imshow(image);

- --> IM=fft(image,-1);
- --> spectrum=real((IM).*conj(IM));
- --> xset("window",1);xbasc();imshow(spectrum,[]);

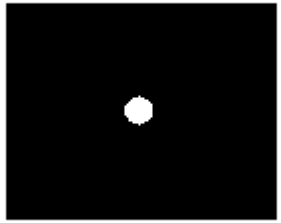
- --> sp2=fftshift(spectrum);to center the spectrum
- --> xset("window",2);xbasc();imshow(sp2,[]);



- --> sp3=log(sp2+1);
- --> xset("window",3);xbasc();imshow(sp3,[]);

Transformée de Fourier

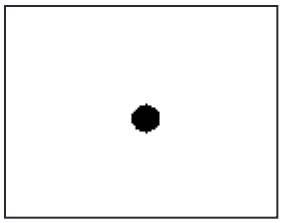
- --> //design the binary filter
- --> z=zeros(image);
- --> x0=round(r/2);radius=10;y0=round(c/2);
- --> for x=x0-radius:x0+radius
- --> y=round(sqrt(radius^2-(x-x0)^2));
- --> z(x,y0-y:y0+y)=1;
- --> end;
- --> xset("window",4);xbasc();imshow(z,[]);



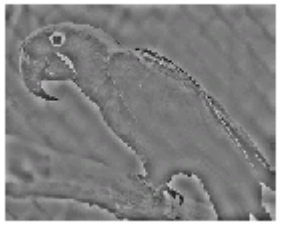
- --> IM2=IM.*fftshift(z);
- --> // Transformée de Fourier inverse
- --> im2=real(fft(IM2,1));
- --> xset("window",5);xbasc();imshow(im2,[]);

Transformée de Fourier

- Avec le filtre inverse
- z=z-1



- --> IM2=IM.*fftshift(z);
- --> // Transformée de Fourier inverse
- --> im2=real(fft(IM2,1));
- --> xset("window",5);xbasc();imshow(im2,[]);



Transformée de Fourier

- ATTENTION:
  ce n’est pas équivalent à
- -->IM3=fftshift(IM).*z
- -->im3=real(fft(IM3,1));

- Les hautes fréquences doivent être au centre de l’image pour permettre de calculer correctement la transformée de Fourier inverse.
- Vous pouvez également construire des filtres avec la fonction mkfftfilter.