Deuxième partie : Les Toolboxes de Matlab

Les toolboxes sont réellement des caisses à outils comportant une collection de fonctions relatives à plusieurs domaines scientifiques et techniques.

Les toolboxes existant à partir de la version 5.3 sont :

Statistics Toolbox, Signal Processing Toolbox, Image Processing Toolbox, Fuzzy Logic Toolbox, Neural Networks Toolbox, Spline Toolbox, Wavelet Toolbox, Mapping Toolbox,

Control System Toolbox, Optimization Toolbox, Robust Control Toolbox, System Identification Toolbox, Higher-Order Spectral Analysis Toolbox, DSP Blockset, Frequency Domain, Mu Analysis and Synthesis Toolbox, Power System Blockset, Data Acquisition Toolbox,

Database Toolbox, Financial Toolbox, Communications Toolbox, MATLAB Web Server,

Symbolic Math Toolbox, Partial Differential Equations (PDE) Toolbox,

Attention : pour être accessibles, ces toolboxes ainsi que leur documentation "pdf" doivent être sélectionnés lors de l'installation de Matlab.

Nous allons essayer, dans ce qui suit, de survoler les toolboxes les plus couramment utilisées pour le calcul scientifique et la modélisation notamment dans le domaine de la géographie.

VII-1- Toolbox Statistiques

Les fonctions du toolbox statistique peuvent être classées comme suit :

- Les fonctions (lois) de densité de probabilité et fdp cumulatives : beta, binomial, Chi2, exponential, F, gamma, normal, Poisson, Rayleigh, T, Uniform, Weibull,…

- Les fonctions inverses de toutes ces fonctions sont aussi disponible : betainv, binoinv, chi2inv, gaminv, norminv, poissinv, raylinv, weibinv,…

- Les générateurs de nombres aléatoires distribués selon ces lois : betarnd, binornd, chi2rnd, frnd, gamrnd, lognrnd, normrnd, poissrnd, unifrnd, weibrnd,…

- Statistiques de toutes ces lois (moyenne, variance, …) : betastat, binostat, chi2stat, gamstat, poisstat, weibstat, …

- Statistiques descriptives : mean, median, geomean, harmmean, std, var, skewness, kurtosis, Kurtosis, iqr (interquartile range), corrcoef , cov,…

- Les tests d'hypothese : ranksum, signrank, ztest, ttest, ttest2, anova1, anova2, …

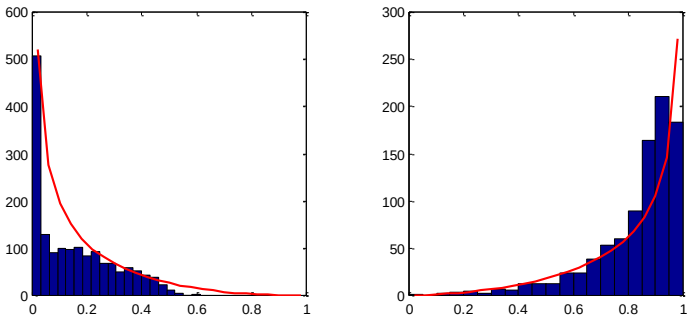
- Modèles linéaires : polyfit, polyval, regress, …

- Classification : pdist, linkage, dendrogram, cluster, clusterdata, …

- Analyse en Composantes Principales : pcacov, pcares, princomp, …

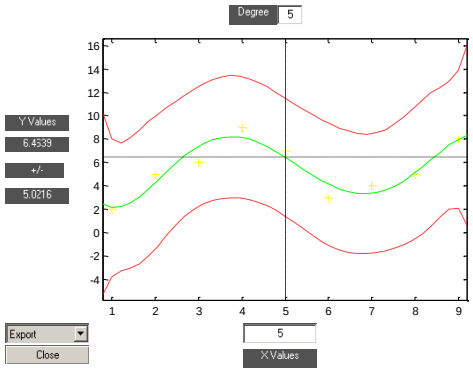
Exemple 1 : Modélisation d'une loi de Probabilité

```
load fkta02.mat
load fkta08.mat
whos
%help stats
%help betafit
x=0.02:0.04:0.98;
BETAFIT(kta02(1:350))
y=betapdf(x,0.5,3.05);
BETAFIT(kta08(1:350))
z=betapdf(x, 2.63,0.5);
subplot(121);hist(kta02,20),hold on,plot(x,(length(kta02)/20)*y,'r')
subplot(122);hist(kta08,20),hold on,plot(x,(length(kta08)/20)*z,'r')
```



EX2 : Lissage polynomial et régression :

```
Polyfit :  help polyfit
X=[1 2 3 4 5 6 7 8 9];
Y=[2 5 6 9 7 3 4 5 8];
polytool(X,Y,3);
a=polyfit(X,Y,5)
```

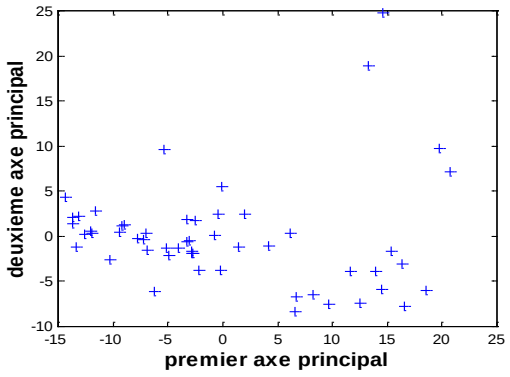


Regression :

```
help regress
b = regress(Y',[X' X'.^2])
```

Eemple3 : ACP aux données météorologiques

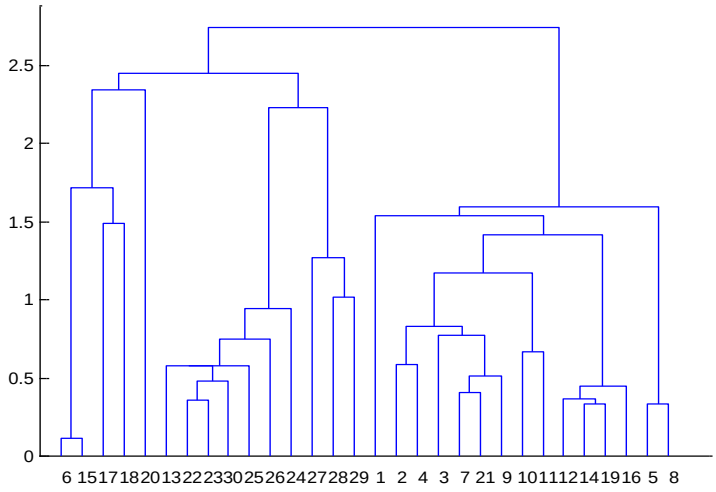
```
help princomp
[PC, SCORE, LATENT, TSQUARE] = PRINCOMP(T);
plot(SCORE)
plot(SCORE)
S1= SCORE(:,1); S2= SCORE(:,2);
figure;plot(S1,S2,'+')
```



- Data management and organization
- Data filtering and visualization
- Descriptive statistics
- Hypothesis testing and ANOVA
- Regression analysis

Exemple 4 : Classification de stations météorologiques (dendrogramme)

Y = PDIST([S1 S1],METRIC); L=LINKAGE(Y, 'single') ; DENDROGRAM(Z);



VII-2- Toolbox traitement de signal

L'un des premiers principes du traitement du signal est la représentation du signal dans une base vectorielle de fonctions (signaux simples) dans le but d'une meilleur extraction et interprétation des propriétés de ce signal.

$$f(t)=\sum_i F_i g_i(t)$$

$f(t)$ est le signal à analyser et $gi(t)$ sont les fonctions de représentation auxquelles on exige de former une base vectorielle orthonormées.

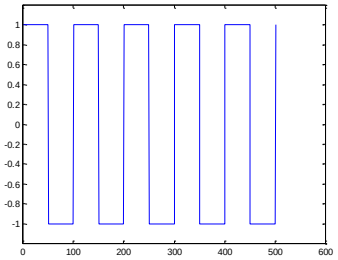
Les fonctions sinus et cosinus sont les fonctions idéales pour cette représentation. En effet, Joseph Fourier a démontré que tout signal pouvait être représenté sous la forme d'une somme (souvent infinie) de signaux sinusoïdaux de fréquences et de phases différentes.
Si le signal est périodique, on parle de série de Fourier. Pour un signal quelconque (cas général), c'est plutôt la transformée de Fourier qui est utilisée.
Le signal et sa transformée de Fourier sont liés par les expressions suivantes.

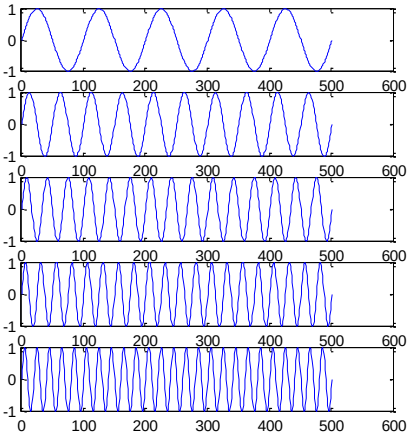
$$f(t)=\sum_i F(\omega).e^{j\omega t}.d\omega \qquad \text{et} \qquad F(\omega)=\sum_i f(t).e^{-j\omega t}.dt$$

Notons que cette somme est plus souvent représentée sous forme d'intégrale.

Exemple : TF de signal carré

```
t = 0:0.01:5;
y = square(2*pi*t);, plot(t,y)
z=fft(y);
y=abs(z(1))*cos(2*pi*t+angle(z(1))
   + abs(z(2))*cos(2*pi*t+angle(z(2))+ …
```



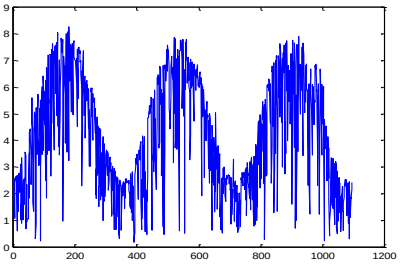


Le traitement de signal comporte outre les techniques basées sur la représentation de Fourier, d'autres transformations qui ne sont que, comme la TF, des représentations dans des espaces de fonctions orthonormées. Le TS comporte aussi des techniques de traitement statistique du signal.

Le toolbox "signal processing" de Matlab comporte une collection très riche de fonctions de traitement du signal. On peut citer les sous familles suivantes :
- Transformées : dct, fft, hilbert, idct, ifft, czt, …
- Analyse et implémentation de filtres : FIR, IIR, …
- Translation fréquentielle (modulation),
- Traitement statistique du signal : corrcoef, cov, pcov, spectrum, tfe, xcorr, xcov, …

- Modélisation paramétrique de série temporelle :
ar, arima, arburg, arcov, aryule, ident, …

Eemple1 : Transformée de Fourier 1D

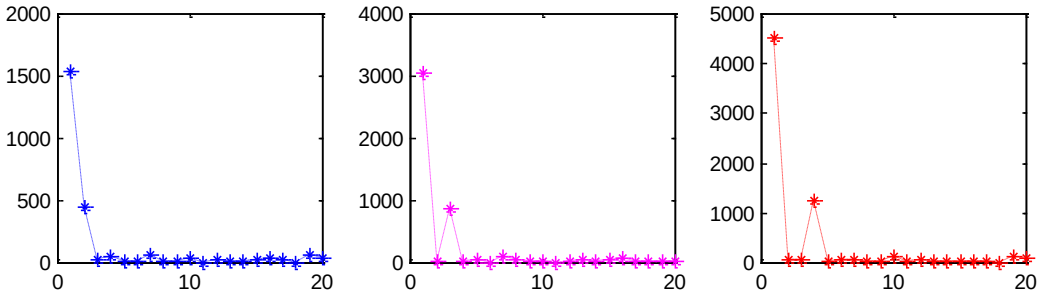


Analysons les séquences annuelles de mesures d'énergie de rayonnement solaire journalier.

```
ghan91;ghan92;ghan93;
GH=[gha91' gha92' gha93']/1000;
plot(GH);pause
plot(GH(1:2*365));pause,
plot(GH(1:365))
```

L'application de la TF à des séquences annuelles, bi-annuelles et tri-annuelles donne :

```
FGH1=fft(GH(1:365));
FGH1= 1.54, -0.44-0.04i, 0.02-0.01i, 0.02+0.05i, -0.002-0.015i, 0.01+0.008i, …

FGH2=fft(GH(1:2*365));
FGH2= 3.05, -0.01-0.03i, -0.87-0.08i, -0.02+0.02i, 0.025-0.04i, 0.003-0.01i, …

FGH3=fft(GH);
FGH3= 4.52, -0.03-0.05i, -0.04-0.07i, -1.25-0.16i, 0.016+0.03i, -0.02+0.07i, …

subplot(131);plot(abs(FGH1(1:20)),'*b:');
subplot(132);plot(abs(FGH2(1:20)),'*m:');
subplot(133);plot(abs(FGH3(1:20)),'*r:');
```

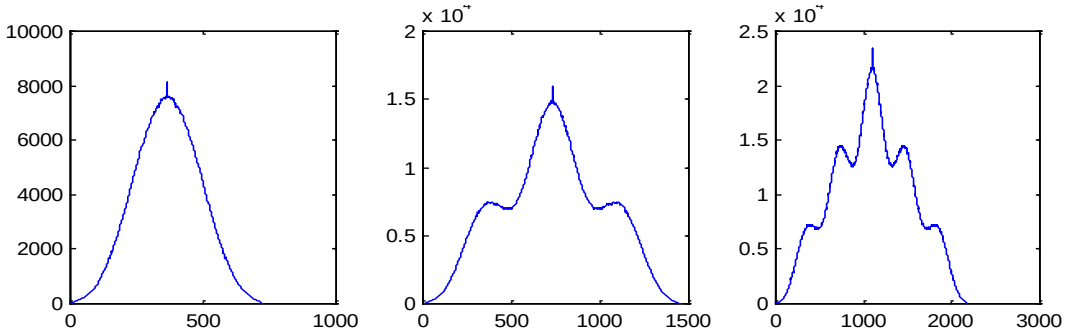


Les harmoniques ont des contribution importantes pour les variations du signal correspondantes.

EX2 : Fonction d'autocorrélation du signal

```
subplot(131);plot(xcorr(GH(1:365)))
subplot(132);plot(xcorr(GH(1:2*365)))
subplot(133);plot(xcorr(GH(1:3*365)))
```

L'autocorrélation mesure la ressemblance entre le signal et des versions décalées de ce même signal. On peut voir que lorsque le décalage est multiple de 365 (nombre de jours/an), l'autocorrélation augmente indiquant une ressemblance (saison par saison) entre les séquences annuelles.



VII-3- Toolbox Traitement d'images

Le toolbox "images" est une collection de fonction spécialisées dans le traitement des images appliqué à des domaines aussi variés que l'imagerie médicale, la télédétection et le contrôle non destructif. Ces outils sont groupés en sous-familles comme suit :

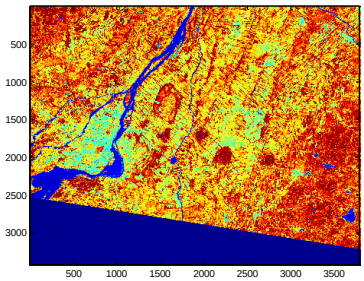
- Visualisation d'images : image, imshow, immovie, montage, colorbar, pixval
- Écriture et lecture d'image : imread et imwrite (pour les format image tels que tif, jpg, bmp,…)
- Operations géometriques : imcrop, imresize, imrotate, …
- Statistiques : mean2, std2, corr2, imhist,
- Analyse d'image : edge, qtdecomp, imcontour, …
- Réhaussement : histeq, imadjust, medfilt2, ordfilt2, …
- Transformées : fft2, ifft2,…
- Filtrage : fsamp2, ftrans2, fwind2, …
- Palette des couleurs : brighten, colormap, …
- Conversion de types : gray2ind, im2bw, im2double, im2uint8, im2uint16, ind2gray, ind2rgb, isbw, isgray, isind, isrgb, mat2gray, rgb2gray, rgb2ind

taper **`help`** ***topic*** pour plus de détails

Eemple1: diverses opérations

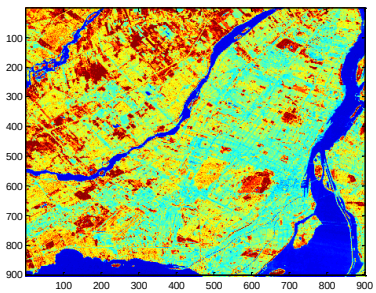
- lecture à partir d'un fichier binaire

```
fid=fopen('b4juin.raw','r');
IM = fread(fid,[3786,3424],'uint8') ;
fclose(fid);
IM=IM';
Image(IM);
```



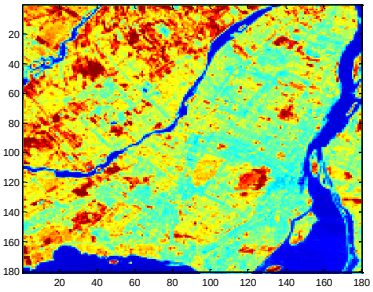
- découpage et affichage

```
IMTL=IM(1300:2200,200:1100);
image(IMTL)
```



- redimensionnement

```
size(IMTL)
ans =   901   901
IMTL2 = imresize(IMTL,0.2,'bilinear');
IMTL2=round(IMTL2);
size(IMTL2)
ans =  180   180
image(IMTL2)
```



- transformation de type :

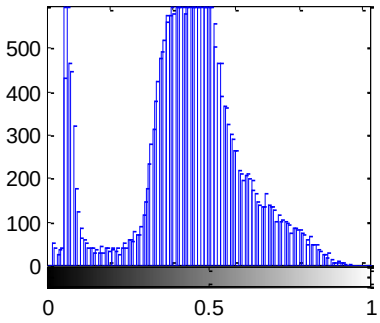
```
AMIN=min(min(IMTL2)) ; AMAX=max(max(IMTL2)) ;
I = MAT2GRAY(IMTL2,[AMIN AMAX]);
```

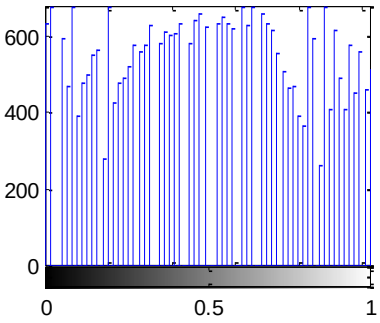
- égalisation d'histogramme

```
J = histeq(I);
imshow(I), figure, imshow(J)
```

- histogrammes

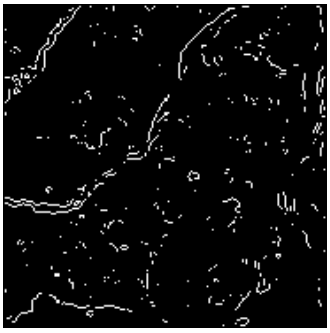
```
imhist(I), figure, imhist(J)
```





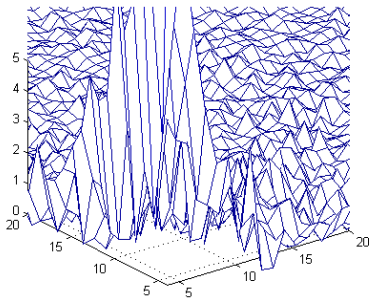
- Detection de contours

```
JE1 = edge(J,'prewitt');
imshow(JE1);
JE2 = edge(J,'canny');
figure;imshow(JE2);
subplot(121), imshow(JE1); subplot(122), imshow(JE2);
```



- transformée de Fourier fft2 :

```
Y=fft2(IMTL2);
mesh(abs(Y))
```



Voir demo pour d'autres exemples (notamment ipss001).

VII-4- Toolbox Ondelettes

Principe :

La transformée de Fourier joue un rôle prépondérant dans l'analyse et le traitement des signaux. Elle permet en outre de réaliser des opérations de filtrage par simple multiplication dans le domaine fréquentiel (domaine de Fourier). Cependant, les signaux possédant des régimes transitoires ou des parties non-stationnaires (signal parole, signaux sismiques, …) sont mal décrits ou représentés par une transformée de Fourier.
Pour pallier à cette déficience, on a eu l'idée d'introduire la représentation du signal en fonction, non pas d'ondes (sinus et cosinus), mais plutôt comme une somme d'ondelettes. Les ondelettes sont des petites ondes finies dans le temps, bien localisées dans le temps (facteur de translation) et de fréquences variables (facteur d'échelle, serrée ou étalée).

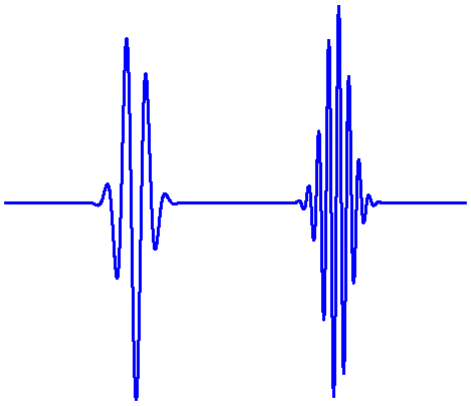


FIG. 1.1 — Fenêtre de la transformée de Fourier à court terme pour deux fréquences différentes

Ainsi, de la même façon que la transformée de Fourier qui peut se définir comme étant une projection sur la base des exponentielles complexes, on définirait la transformée en ondelettes comme la projection sur la base des fonctions ondelettes.

$$TO(a,b) = \frac{1}{\sqrt{(a)}} \int_{-\infty}^{+\infty} f(t)\, \psi(\frac{t-b}{a})\, dt \quad \text{avec} \quad a,b \in \mathbb{R}, a \neq 0$$

$$TO(a,b) = \int_{-\infty}^{+\infty} f(t)\, \psi_{a,b}(t)\, dt \quad \text{avec} \quad \psi_{a,b}(t) = \frac{1}{\sqrt{(a)}} \psi(\frac{t-b}{a})$$

Les coefficients d'ondelettes (TO(a;b)) dépendent de deux paramètres a et b : a étant le facteur d'échelle et b le facteur de translation.
Les fonctions a;b(t) sont obtenues à partir de la dilatation et de la translation de la fonction ***mère*** ψ(t). Les fonctions a;b(t) sont par conséquent parfois appelées les ondelettes ***filles***. Ces fonctions forment une base orthonormée. Tout comme la transformée de Fourier, la transformée en ondelettes est inversible.

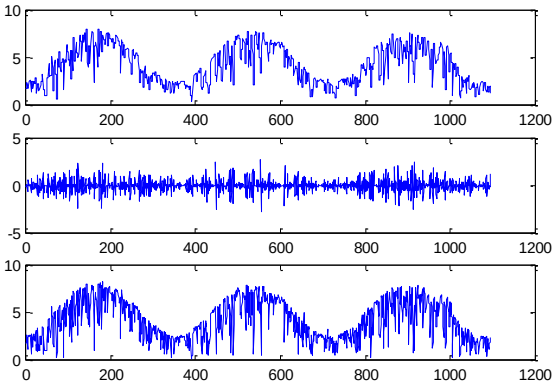
$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} < f\, |\psi_{a,b} > \ \psi_{a,b}\, da\, db$$

Le wavelet toolbox de Matlab (5.3 et plus) possède un nombre très important de fonctions spécialisées. Taper **`help wavelet`** pour voir toute la liste.

Exemples :

```
ghan91;ghan92;ghan93;
GH=[gha91' gha92' gha93']/1000;
l=length(GH);
```

```
[ca1,cd1]=dwt(GH,'db1');
a1=upcoef('a',ca1,'db1',1,1);
d1=upcoef('d',cd1,'db1',1,1);
subplot(311),plot(a1);subplot(312),plot(d1);subplot(313),plot(a1+b1);
```



Le toolbox wavelet possède également une interface graphique qui permet d'effectuer une analyse en ondelettes à un signal sauvegardé dans un fichier *.mat. Pour activer ce GUI, taper **`wavemenu`**.

VII-5- Mapping Toolbox

Ce Toolbox regroupe des fonctions relatives à la cartographie et à la géographie en général. Taper **`help map`** pour voir la liste complète des fonctions disponible dans la version de Matlab utilisée ou consulter le fichier map_ug.pdf pour un cours complet sur le mapping toolbox.

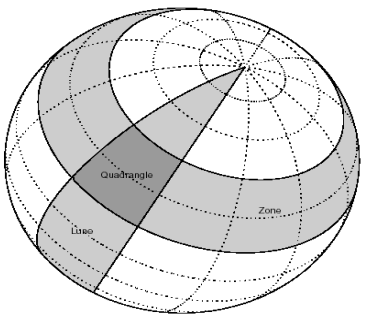
Exemple 1 : distance entre deux points

```
help distance,
distance(46,-73,37,3)
ans =  55.45017803314318
deg2km(55.45)
ans =   6.165758682440783e+003
deg2nm(55.45)
ans =   3.329243349050099e+003
```

Exemple 2 : surface d'un territoire

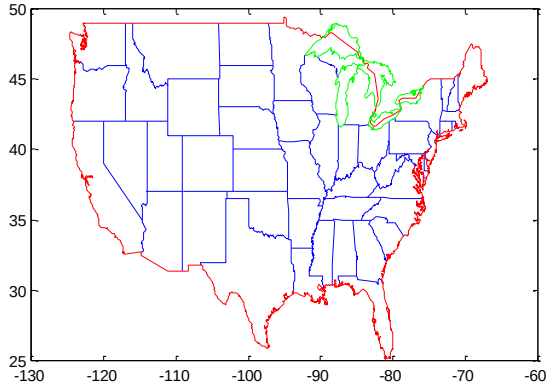
- aire délimitée par latmin-lonmin et latmax-lonmax

```
area=areaquad(15,0,45,30)
area =   0.01867
```



- aire délimitée par un polygone

```
load usalo
plot(statelon,statelat),hold on,plot(uslon,uslat,'r'),hold on,
plot(gtlakelon,gtlakelat,'g')

areaint(uslat,uslon)

ans =
   0.01553842116460
   0.00000677438424
   0.00000070866308
```



VII-6- Toolbox de Maths Symboliques

Les fonctions du symbolic toolbox permettent de faire des opérations mathématiques sur des variables ou des fonctions symboliques (i.e. non numériques). On peut citer les sous-familles suivantes :

- Calcul : diff, int, limit, taylor, symsum,…
- Algèbre linéaire : diag, triu, inv, det, eig, …
- Simplification : simplify, simple, numden, …
- Solution d'équations : solve, dsolve, finverse, compose, …
- Transformées : fourier, laplace, ztrans, ifourier, ilaplace, iztrans,
- Demonstrations : symintro, symcalcdemo, symlindemo, symvpademo, symrotdemo, …

Exemple 1 : intégration et différentiation

```
syms x

int(x^2+3*x^3-2*x)
ans = 1/3*x^3+3/4*x^4-x^2

diff(log(x^2)-exp(x^3)+2*x)
ans = 2/x-3*x^2*exp(x^3)+2
```

Exemple 2 : résolution d'un système d'équation

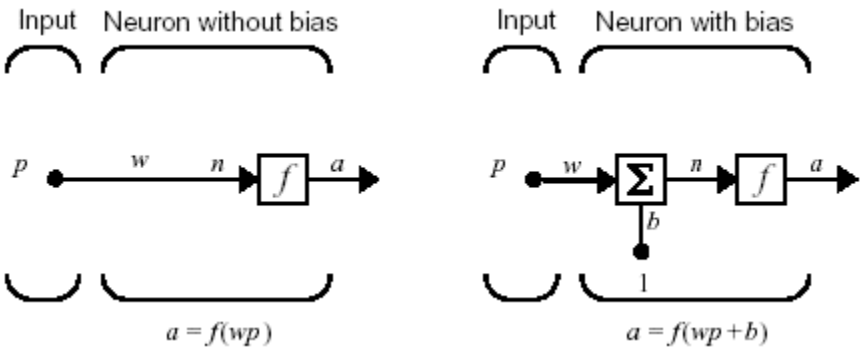
```
syms x y
[x,y] = solve('x^2 + x*y + y = 3','x^2 - 4*x + 3 = 0')

x =
[ 1]
[ 3]

y =
[    1]
[ -3/2]
```

VII-7- Toolbox Réseaux de Neurones

Principe :

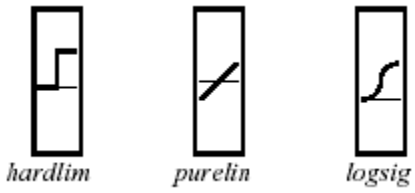
Les réseaux de neurones artificiels ont été conçus à la base du modèle du neurone naturel. Ce dernier étant excité par un stimuli, il y répond selon sa fonction de transfert par une sortie (réponse) qui peut être une atténuation ou une amplification de l'entrée (stimuli).

Neurone Simple :

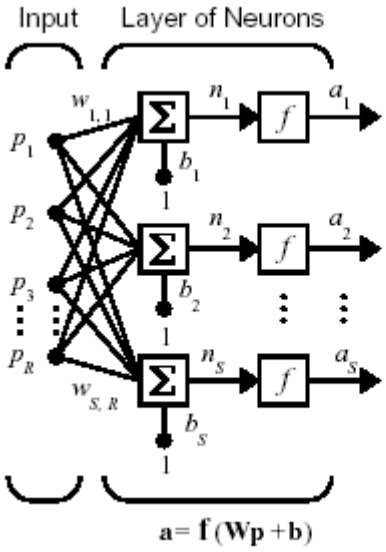


Le poids *w* sert à amplifier ou à atténuer l'effet du stimuli sur la réponse.

Fonctions de transfert : on peut trouver différentes formes de fonctions de transfert



Une couche de neurones multi-entrées multi-sorties :



Where...

R = # of elements in input vector

S = # Neurons in Layer

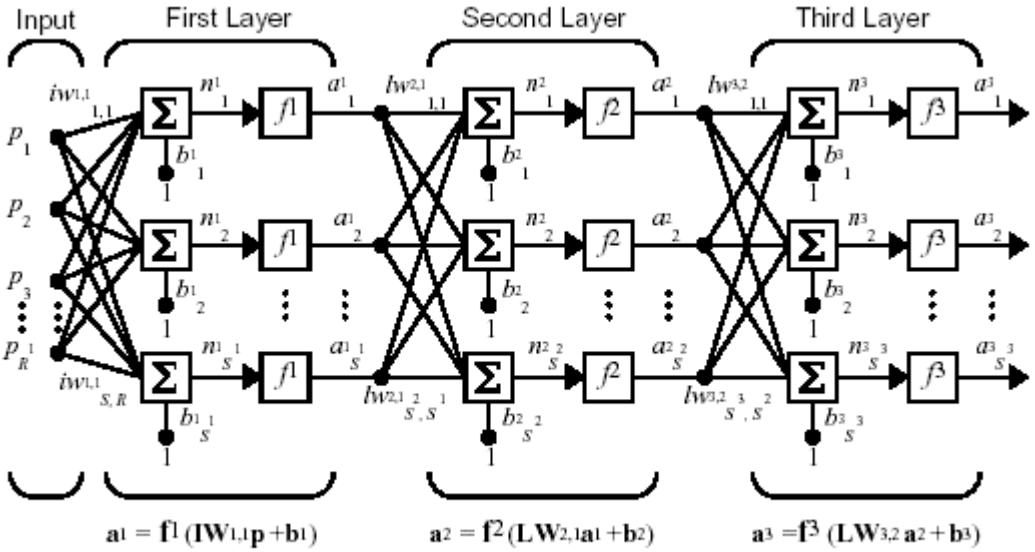
Matrice des poids

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

Dans un réseau de neurones, les sorties sont donc reliées aux entrées via des fonctions et des poids.

Réseau de neurones multi-couches

Afin de pouvoir détecter ou reproduire d'éventuelles relations complexes entre les sorties et les entrées, il est possible de devoir recourir à un réseau à plusieurs couches. Plus on multiplie la complexité du réseau, plus on le dote "d'intelligence" et de "mémoire".



$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\,\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

Utilisation dans la modélisation :

Pour concevoir un modèle à base de réseaux de neurones, on doit d'abord choisir une structure (nombre d'entrées, nombre de sorties et nombre de couches), sélectionner des fonctions de transfert et enfin calculer les poids *w* qui lient les différents neurones.
Les poids sont calculés grâce à une procédure (algorithmes) d'entraînement du réseau à partir d'entrées et de sorties déjà existantes. Une fois ces poids calculée, le réseau pourra être utilisé pour simuler des réponses à d'autres stimuli.

NNET de Matlab

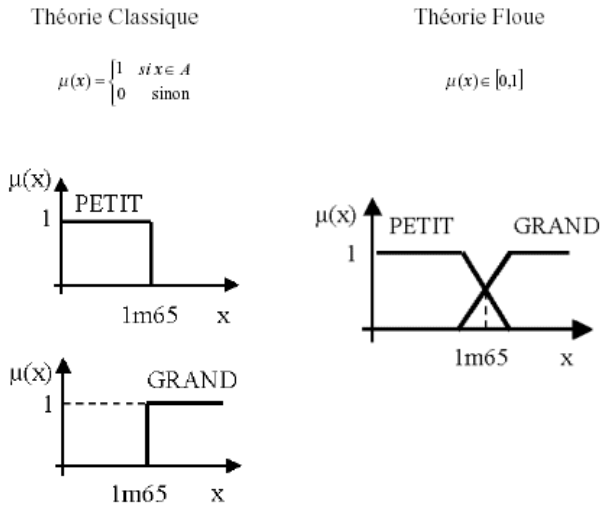
Taper **`help nnet`** pour voir les possibilités qu'offre Matlab pour cette technique. Consulter également le fichier *nnet.pdf* pour un cours simple et détaillé avec des exemples d'application. Voir aussi **`demo`** - toolbox - neural network pour des exemples illustrés.

VII-8- Toolbox Logique Floue

Principe :

La logique floue est un concept introduit par Lotfi Zadeh en 1965 et qui utilise la notion de variables linguistiques (le problème est donc posé en langage humain). Le concept du floue viole le principe du tiers exclu d'aristote.

Exemple :



On parlera dans cet exemple de variable linguistique taille et de valeurs linguistiques grand et petit.

Variables floues :
Une variable floue est décrite donc par une fonction d'appartenance à un domaine de sa gamme dynamique et non pas des valeurs numériques ou des attributs fixes.

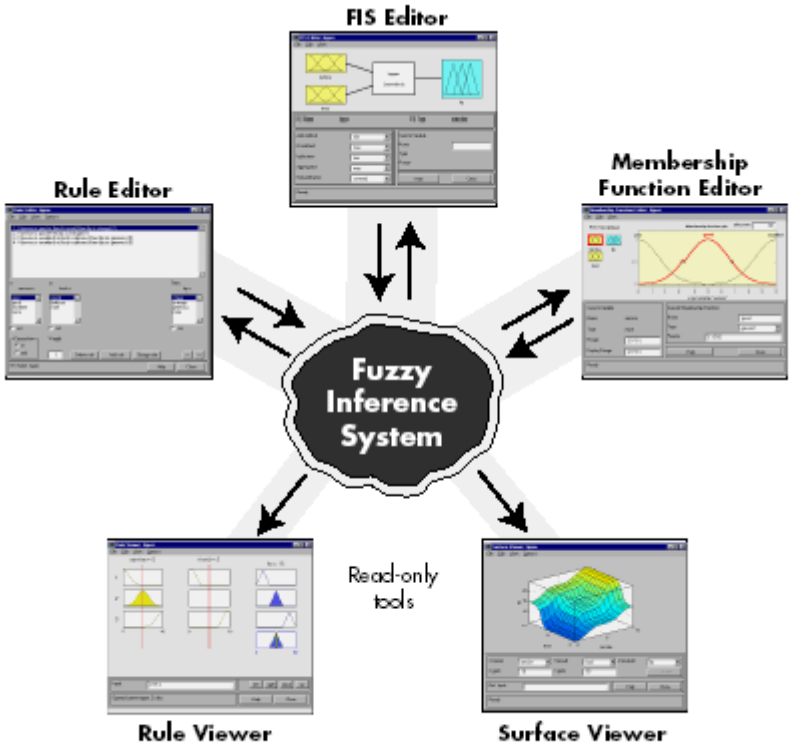
Règles "if then entre" variables floues :
Les relations entre les variables floues se font par des règles Si Alors (If Then). Par exemple deux variables x et y sont reliées par la règle :

if x is A then y is B

ou encore si on a d'autres autre variable en jeu.

if (x1 is A1 and x2 is A2) or x3 is A3 then y is B

La logique floue sous Matlab
Le "fuzzy toolbox" de Matlab possède des fonctions qui peuvent être utilisées de façon classique comme les autres fonctions ou via une interface graphique activée par l'instruction **`fuzzy`**. L'utilisation du fuzzy-GUI de Matlab pour la modélisation de variables floues est organisées en plusieurs étapes. D'abord le choix des fonctions d'appartenance, ensuite l'écriture des règles et enfin la visualisation des résultats.



Le fichier *fuzzy_tb.pdf* contient une introduction à la théorie des ensembles floues ainsi qu'un exemple complet sur l'utilisation du GUI pour résoudre un problème de modélisation avec cette technique. Le demo - toolbox -fuzzy contient également des exemples d'utilisation de cette technique.

CONCLUSION

Matlab est un langage de programmation, un logiciel, un monde de fonctions et de techniques. Matlab est un outil très simple, très puissant, très complet et qui ne nécessite pas une formation spéciale pour le maîtriser, juste de ne pas hésiter à utilise **`help`**, à naviguer dans **`demo`** et à consulter les fichiers de documentation pdf.