



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Ibn Khaldoun – Tiaret
Faculté des Mathématiques et de l'Informatique
Département d'Informatique



METHODES NUMERIQUES

Travaux Pratiques POUR :

2ème ANNEE LICENCE INFORMATIQUE

Par : Mr. KARIM MEZZOUG

karim.mezzoug@univ-tiaret.dz

TP 1 : Rappels sur Scilab

1. Navigation au sein de l'arborescence des dossiers

```
-->dir // afficher le contenu du dossier courant
-->chdir('.') // remonter au dossier immédiatement au-dessus
-->pwd // Arborescence du dossier courant
-->chdir ('../..') //fait remonté de deux crans
-->pwd
-->chdir('CHEMIN.....')
-->ls // afficher les fichiers du dossier courant
-->help // afficher l'aide Scilab sur une commande
-->apropos('root') // afficher l'aide sur un mot clef
-->help ls // afficher l'aide sur la commande ls
-->clc // effacer la fenêtre de commande
```

2. Manipulations de bases

```
-->12.256 // valeur déclarée
-->2*ans // utiliser le dernier résultat
-->A=1000 // affectation et affichage
-->A=1000 ; B=2000 ; // affectation sans affichage
-->C=A+B // Addition
-->C=A-B
-->C=A/B
-->C=A*B
-->1.2E-5 // nombre flottant
-->%pi // valeur de pi
-->%eps // valeur de l' epsilon machine
-->%e // valeur de exponentiel
-->A=1+2*i // Nombre complexe
-->real(A) , imag(A) // réel et imaginaire complexe
-->afge=sin(2*pi) ; disp(afge) ; // affichage du résultat
-->v=exp(0.1*pi)
-->format('e',20); // forcer le format scientifique à 20 caractères
-->format('v',6); // forcer le format ordinaire à 6 caractères
-->exp(1000) // affichera inf (L'infini machine)
-->%inf-%inf // affichera Nan (Note a Number)
--> 2+5+6+.. //.. si la commande est trop long
```

> 5

-->clear // efface tout l'espace de travail

3. Quelques Fonctions

| Fonction classique : | Syntaxe dans Scilab | Exemple |
|---|---------------------|------------|
| Racine carrée : \sqrt{x} | sqrt | sqrt(2) |
| Carré : x^2 | ^2 | 5^2 |
| Puissance : x^P | ^ | 5^7 |
| Exponentielle : e^x | exp | exp(0) |
| Logarithme : $\ln(x)$ | log | log(1) |
| Logarithme en base 10 : $\log_{10}(x)$ | log10 | log10(10) |
| Cosinus : $\cos(x)$ | cos | cos(0) |
| Sinus : $\sin(x)$ | sin | sin(0) |
| Tangente : $\tan(x)$ | tan | tan(0) |
| Valeur absolue : $ x $ | abs | abs(-1) |
| Partie entière inférieure : $\lfloor x \rfloor$ | floor | floor(1.4) |
| Partie entière supérieure : $\lceil x \rceil$ | ceil | ceil(1.4) |
| Arrondi entier le plus proche | round | round(1.4) |

4. Chaines de caractères

```
-->'Une chaîne de caractères'
-->"Une autre"
-->'L''été'
-->s='L''été' , length(s)
-->s+' plus long'
-->a=string(2+3)
-->d=evstr('sqrt(3)/2')
-->part('abcdefg',2)
-->part('abcdefg',[1 2 6])
-->strindex('Mississippi','ss')
-->ascii('abc')
-->ascii([97 98 99])
-->lettres(771)
```

5. Vecteurs et matrices

Vecteurs

```
-->V = [1 2 3 4 5 6 7 8 9 10]
-->x = [-1.3 sqrt(3) (1+2+3)*4/5]
-->V(6)
-->V(13)=63
```

```
-->M=length(V)
-->U=6*V
-->(U.*V)./(U-V)
-->U.^3
-->sin(U)
-->[ U [1 2 3] ] // concaténation de vecteurs
```

Matrices

```
-->A = [1 2 3;4 5 6;7 8 9]
-->B= sin(A)
-->(A-6*B).*(cos(A).^2)
-->A' // transposé de A
-->A = [1 2 3;4 5 6];
-->B = [7;8;9];
-->A*B // Solution de AX=B
-->x = [1;2],y = [3;4], x'*y // Produit scalaire
-->x = 3:9
-->x = 1:0.5:4
-->x = 6:-1:0
-->V=[10:-1:0];
-->V(1:2:length(V)) // extraire tous les nombres d'indices pairs de
-->tic; cos(1:1000000) ; toc // calcul le temps d'exécution écoulé
entre tic et toc.
```

6. Autres

- $A(i,:)$ extrait la i éme ligne de A .
- $A(:,j)$ extrait la j éme colonne de A . On considère successivement toutes les lignes de A et on choisit le j éme élément de chaque ligne.
- $A(:)$ reforme la matrice A en un seul vecteur colonne en concaténant toutes les colonnes de A . $A(:,j:k)$ extrait la sous-matrice de A formée des colonnes j à k .
- $A(j:k,:)$ extrait la sous-matrice de A formée des lignes j à k .
- $A(j:k,q:r)$ extrait la sous-matrice de A formée des éléments situés dans les lignes j à k et dans les colonnes q à r .

7. Graphiques 2D

Une seule courbe

```
-->x=0.1:0.1:10; ; y=sin(x)./x; ;xbasc(); // efface le contenu de la
fenetre graphique ;plot2d(x,y) ;
```

Deux courbes superposées

```
-->x=0.1:0.1:10; y=sin(x)./x; z=cos(x)./x; xset("window",1);  
plot2d(x,y); plot2d(x,z) ;
```

Plusieurs courbes

```
-->x=x';y=y';z=z'; xbasec();plot2d(x,[y z]) ; // vecteurs colonnes  
-->x=1:10; xbasec();plot2d2(x,x); // courbes creneaux  
-->xbasec();plot2d3(x,x); // courbes barres
```

8. Les conditions

```
--> a=5;  
--> b=5;  
--> (~isnan(a)) & (b>=a)  
    if a<=0 b=0  
    else b=log(a)  
    end
```

9. Les Boucles

FOR :

```
T(1)=1; T(2)=1;  
for k=3:25  
T(k)=T(k-1)+T(k-2); // fibonacci  
End
```

WHILE :

```
k=0;  
while ~isinf(exp(k))  
k=k+1;  
end // l'infinie machine de exponentiel
```

10. Scripts et Fonctions

Les scripts :

Un script est une suite d'instructions. Scilab lit le fichier ligne après ligne ce qui revient donc à exécuter les commandes à la suite, comme si elle étaient tapées. Par exemple :

```
-->v = [0:.01:10]*2*%pi;  
-->fv = sin(v);  
-->plot(fv)
```

Peut être remplacé par un fichier 'essai.sce' contenant les lignes :

```
v = [0:.01:10]*2*%pi;  
fv = sin(v); plot(fv)
```

et lancée grâce à :

```
--> exec('essai.sce');
```

Les fonctions.

Une fonction est une suite d'instructions prenant un ou plusieurs arguments en entrée et renvoyant un ou plusieurs arguments en sortie. On la définit à l'aide de l'instruction `function` et on l'enregistre dans un fichier `.sci`. Ainsi, si nous voulons écrire une fonction prenant un vecteur et élevant chacun de ses éléments au carré, on peut écrire dans un fichier dénommé `'carre.sci'` :

```
function resultat=carre(argument)
n=2;
// n=input('donner la valeur de n')
resultat = argument.^n;
endfunction
```

Pour utiliser la fonction, il faut d'abord la charger dans Scilab :

```
--> getd // charge toutes les fonctions du dossier courant.
```

```
--> exec 'carre.sci' // charge uniquement la fonction carre.sci
```

Puis, on s'en sert exactement comme d'une fonction classique de Scilab :

```
-->a=carre([1 2;3 4])
```