

Initiation à Scilab

On se placera dans un répertoire qu'on aura préalablement créé et on lancera Scilab (sous UNIX) ou bien après lancement à partir de l'icône (sous Windows) on tapera dans la fenêtre de commandes : `cd repertoire`.

1 Saisie manuelle de commandes

Scilab peut être utilisé en tapant des commandes dans la fenêtre principale. Tous les objets de Scilab sont considérés comme des matrices. Par exemple un scalaire est une matrice une ligne, une colonne. Une fonction est représentée comme un vecteur de valeurs associé à un vecteur d'abscisse etc... L'ajout d'un point virgule en fin de ligne inhibe l'affichage.

Vecteurs et matrices

--> x=[1,0,5]	--> y=[-1;2;3]	--> A=[1 2 4; 2 3 -2; 1 0 1];
--> A	--> z=0:0.1:1	--> u=linspace(0,4,10)
--> B=diag(y)	--> C=triu(A)	--> D=tril(A)

Quelques matrices prédéfinies

--> I=eye(3,3)	--> N=zeros(3,2)	--> U=ones(2,3)
----------------	------------------	-----------------

Accéder à une sous-matrice

--> x(3)	--> A(1:2,2:3)	--> A(2,:)
--> diag(A)	--> A([1 3],[2 1])	--> A(\$,2)

Opérations d'algèbre matricielle

--> x'	--> A^2	--> x*y
--> y*x	--> A\y	--> x*A*y

Opérations élément par élément

--> x.^2	--> x.*y'	--> sin(z)
--> (1)./y	--> x+2	

Nombres complexes

--> c=1+2*i	--> c^2	--> sin(c)
--> imag(c)	--> real(c)	

Visualiser le graphe de la fonction $f(x) = e^{-x} \sin 4x$ sur $[0, 2\pi]$

```
--> x=linspace(0,2*pi,101);
--> y=exp(-x).*sin(4*x);
--> plot(x,y,'x','y','y=exp(-x)*sin(4x)')
--> a=gca(); a.x_location="middle";
```

Exercices

1. construire les matrices A et B de taille 4×4 telles que

$$a_{ij} = 2 \text{ si } i \neq j \text{ et } a_{ii} = 1 \quad \text{et} \quad b_{ij} = j$$

2. Soit A une matrice carrée; que vaut $\text{diag}(\text{diag}(A))$?
3. Définir une matrice carrée A quelconque (par exemple avec `rand`) et construire en une seule instruction, une matrice triangulaire inférieure T telle que $t_{ij} = a_{ij}$ pour $i > j$ (les parties strictement triangulaires inférieures de A et T sont égales) et telle que $t_{ii} = 1$ (T est à diagonale unité).

2 Programmation

Exécution d'un script Dès que le nombre de commandes dépasse quelques unités, il est intéressant de créer un fichier contenant ces commandes pour ne pas avoir à les retaper sans cesse. Pour cela on crée un fichier avec l'extension `.sce`, appelé fichier script. Ceci peut être fait avec n'importe quel éditeur. Cependant, les versions récentes de Scilab ont un éditeur intégré. Voici un exemple de script :

```
// mon premier script Scilab

a = input(" Rentrer la valeur de a : ");
b = input(" Rentrer la valeur de b : ");
n = input(" Nb d'intervalles n : ");

// calcul des abscisses
x = linspace(a,b,n+1);

// calcul des ordonnees
y = exp(-x).*sin(4*x);

// un petit dessin
plot(x,y)
```

Après avoir sauvegardé ceci dans un fichier `essai.sce`, l'exécution du script est lancée par la commande

```
--> exec essai.sce
```

Création d'une fonction Une fonction, contrairement à un script, ne s'exécute pas, mais au contraire on l'appelle à partir d'une autre fonction, d'un script ou de la ligne de commande. Créons par exemple un fichier `f.sci` contenant

```
function res=f(x)
    res = x.^3;
endfunction
```

Avant d'utiliser la fonction, elle doit être chargée dans Scilab. On peut alors l'appeler.

```
--> getf f.sci
--> f([1 2 3])
```

3 Aide en ligne

Une aide en ligne est disponible grâce à l'instruction `help nom_de_commande`. Quand on ne connaît pas le nom exact de la commande, on peut utiliser l'instruction

```
--> apropos mot_clef
```

qui effectue une recherche dans les entêtes des rubriques d'aide (en anglais).

Exercice : quelles commandes permettent de

- définir des points à l'aide de la souris ;
- calculer les valeurs propres d'une matrice ;
- changer le format d'affichage des nombres ;
- tracer un graphe en coordonnées polaires ;
- calculer la longueur d'un vecteur ;
- ouvrir une boîte de dialogue pour entrer des paramètres ;
- calculer le temps d'exécution d'une tâche.