

Voici un panorama des fonctions dont vous devez connaître l'utilisation.

Pour toutes les autres fonctions que nous avons rencontrées, la syntaxe d'utilisation vous sera donnée.

1 Opérations et fonctions usuelles

1.1 Les symboles usuels

1.1.1 Le symbole d'affectation « = »

- $x=3$ signifie que la valeur 3 est affectée à la variable x ($x \leftarrow 3$).
- $x=x+3$ signifie que le résultat de l'opération $x+3$ écrase l'ancienne valeur de x et devient la nouvelle valeur de x ($x \leftarrow x+3$).

Remarque : chronologiquement, la quantité du membre de droite est d'abord calculée, et est ensuite affectée à la variable indiquée à gauche.

1.1.2 Le délimiteur « ; »

En mode console, le « ; » marque la fin d'une instruction.

En mode éditeur sci, la fin d'une instruction peut être marquée par une nouvelle ligne.

1.1.3 Opérations algébriques

+ - * /

1.1.4 Nombres et valeurs usuels

%e (e) %pi (π) %i (i)
ans (variable qui stocke le dernier
résultat calculé (« last answer »))

1.1.5 Comparaison de deux nombres

< > <= ($\leq$) >= ($\geq$) <> ($\neq$) == (=)

1.1.6 Connecteurs logiques

& (et) | (ou inclusif) ~ (non)

1.2 Fonctions usuelles

log (ln) exp ^ cos sin tan
atan (arctan) sqrt ($\sqrt{\cdot}$) floor ([.]) abs ([.])

1.3 Déclaration d'une fonction

- `deff("y=f(x)","y=...")` permet de définir une fonction pas trop compliquée, par exemple `deff("y=f(x)","y=sin(2*x)")` définit $f : x \mapsto \sin(2x)$.
- Et pour une fonction plus élaborée :

```
function y=f(x)
:
y=...
endfunction
```

2 Matrices

2.1 Vecteurs ou Matrices-lignes

- $A=[0 \ 5 \ 4]$ ou $A=[0,5,4]$
- $A=[1:100]$ ($A=[1 \ 2 \ \dots 100]$)
- $A=[1:0.5:100]$ ($A=[1,1.5,2,\dots,100]$)
- $A(i)$: i -ème coefficient de la matrice A .
- $B=A(n1:n2)$ donne la sous-matrice des coefficients d'indices $n1$ à $n2$.

2.2 Matrices à plusieurs lignes et/ou plusieurs colonnes

À retenir : le séparateur de colonnes est « , » et le séparateur de lignes est « : ».

- $A=[1,2,3;4,5,6]$ définit

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

- Une matrice à plusieurs lignes peut être obtenue par concaténation de matrices à une ligne, toutes de même longueur. Syntaxe : $A=[L1;L2;\dots;Ln]$
- Une matrice à plusieurs colonnes peut être obtenue par concaténation de matrices à une colonne, toutes de même longueur. Syntaxe : $A=[C1,C2,\dots,Cn]$

- $A(i,j)$: coefficient i -ème ligne, j -ème colonne de la matrice A .

2.3 Taille d'une matrice

- $[n,p]=\text{size}(A)$ matrice à n lignes et p colonnes.

2.4 Création de matrices

- $\text{zeros}(m1,m2)$: renvoie une matrice nulle à $m1$ lignes et $m2$ colonnes.
- $\text{ones}(m1,m2)$: renvoie une matrice remplie de « 1 » à $m1$ lignes et $m2$ colonnes.
- $\text{eye}(n,p)$: renvoie une matrice à n lignes et p colonnes avec des « 1 » sur la diagonale principale et des « 0 » partout ailleurs (matrice échelonnée).
- $\text{eye}(A)$: renvoie une matrice de même dimension que la matrice A avec des « 1 » sur la diagonale principale et des « 0 » partout ailleurs (matrice échelonnée).

2.5 Fonctions algébriques sur les matrices

- +, - et * : addition, soustraction et produit matriciels.
- $\text{inv}(A)$ donne l'inverse de la matrice A .
- $\text{rank}(A)$ donne le rang de la matrice A .
- $\text{spec}(A)$ donne le spectre de A . Par exemple $\text{vp}=\text{spec}(A)$ retourne le tableau des valeurs propres et $[\text{vp},X]=\text{spec}(A)$ retourne valeurs et vecteurs propres de A .
- A' donne la transposée de A .

2.6 Opérations sur les coefficients

- **A+1** : on ajoute 1 à tous les coefficients de la matrice A.
- **2*A** : tous les coefficients de A sont multipliés par 2.
- **sqrt(A)**, **log(A)**,... : calcule les images par la fonction considérée de tous les coefficients de A.
- **.*** : on multiplie point par point les coefficients du premier objet avec les coefficients du deuxième objet (l'espace avant le point est important).
- **./** : on divise point par point les coefficients du premier objet avec les coefficients du deuxième objet. Par exemple : **1 ./A** calcule l'inverse de tous les coefficients de A, mais ne calcule pas l'inverse de A (cf. algèbre linéaire).
- **.^** : met à la puissance point par point les coefficients du premier objet par les coefficients du deuxième objet.
- **sum(A)** : somme de tous les coefficients de la matrice A.

2.7 Recherche d'éléments et tri

- **min(A)** : donne le plus petit des coefficients de la matrice réelle A.
- **max(A)** : donne le plus grand des coefficients de la matrice réelle A.
- **gsort(A)** : trie dans l'ordre décroissant les valeurs d'une matrice et **gsort(A,'g','i')** permet un tri dans l'ordre croissant.
- **find(A==s)** : recherche parmi les coefficients de A ceux qui sont égaux à s, et donne la liste des indices correspondant.

2.8 Fonctions statistiques

- **mean(A)** : fait la moyenne arithmétique de tous les coefficients de la matrice.
- **cumsum(A)** : somme cumulative des éléments d'un tableau.

Par exemple **cumsum([1,2,3,4])** donne **[1,3,6,10]**

3 Interface

3.1 Saisie au clavier et affichage à l'écran

3.1.1 Input

L'instruction **x=input("message")** permet de stocker dans la variable x la saisie réalisée par l'utilisateur. Le message doit être écrit entre guillemets.

3.1.2 Disp

L'instruction **disp(...)** permet d'afficher un message pouvant contenir éventuellement (même fréquemment) le contenu d'une variable. L'affichage s'établit de droite vers la gauche. Par exemple **disp(x,"x=")**. Le séparateur entre les objets à afficher est la virgule « , ».

3.2 Représentations Graphiques

3.2.1 plot2d et fplot2d

- **plot2d(x,y)** permet de tracer la courbe rejoignant les points dont les abscisses sont les coefficients de x et les ordonnées sont les coefficients de y.
- **plot2d(x,y,-1)** permet de tracer le nuage de points dont les abscisses sont les coefficients de x et les ordonnées sont les coefficients de y.

3.2.2 fplot2d

fplot2d(x,f) trace la courbe de la fonction f en joignant les points $(x, f(x))$ où les abscisses x sont les coefficients de x. La fonction f doit avoir été définie au préalable.

3.2.3 plot3d et fplot3d

Même principe que **plot2d** et **fplot2d** mais en dimension 3, ce qui permet donc le tracé des surfaces.

Syntaxe :

- **plot3d(x,y,z)**
- **fplot3d(x,y,f)**

4 Probabilités

4.1 Générateur aléatoire

- **rand** simule une loi uniforme sur $[0; 1[$. **rand(m,n)** permet de créer directement une matrice de taille $m \times n$ dont les coefficients suivent la loi uniforme sur $[0; 1[$ et sont indépendants.

4.2 Répartition de la loi normale

- **cdfnor("PQ",x,m,s)** calcule $F_X(x)$ pour une variable X de loi normale $\mathcal{N}(m; s^2)$. En particulier **cdfnor("PQ",x,0,1)** calcule $\Phi(x)$.
- **cdfnor("X",m,s,P,Q)** avec $P + Q = 1$ calcule la réciproque de F_X pour une variable X de loi normale $\mathcal{N}(m; s^2)$: **cdfnor("X",m,s,P,Q)** donne la valeur x telle que $F_X(x) = P = 1 - Q$.

Par exemple, **cdfnor("X",0,1,0.975,0.025)** donne **1.959964** car $\Phi(1,96) \simeq 0,975$.

5 Structures conditionnelles et répétitives

- **if ... then ... end**
- **if ... then ... else ... end**
- **for k =n1:n2 ... end** : k prend toutes les valeurs entières de n1 à n2 (inclus)
- **for k =n1:pas:n2 ... end** : k prend toutes les valeurs de n1 à n2 en avançant de pas à chaque fois ($k = n1$, puis $k = n1 + pas$, puis $k = n1 + 2 \times pas$, etc.)
- **while ... then ... end** : répète les instructions comprises entre **then** et **end** tant que le test effectué entre **while** et **then** est positif.