MEMENTO SUR SCILAB

1 Utilisation du logiciel

— **F5** (ou via le menu) : sauver et exécuter sans écho
— **ctrl-L** (ou via le menu) : exécuter avec écho
— `help` (ou via le menu) : ouvrir l'aide
— `apropos sujet` : cherche dans l'aide les pages contenant le mot `sujet`

2 Calculs sur les scalaires

— constantes usuelles : `%pi`, `%i`, `%e`
— `sqrt`, `atan`, `asin`, `acos`, `log` (logarithme naturel), `abs` (valeur absolue), `floor` (partie entière (inférieure))
— pour les nombres complexes : `real`, `imag`, `conj`
— `x^y` (exponentiation x^y)
— comparaisons : `<`, `>`, `<=`, `>=`, `==`, `<>` (différent de)
— opérations logiques : `&`, `|` (ou), `~` (non)

3 Matrices

3.1 Définition

— `[a b c;d e f]` (pour $\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$), matrice vide : `[]`
— `zeros(m,n)`, `ones(m,n)`, `eye(n,n)` (matrice identité) ; avec même taille que `x` : `zeros(x)`, `ones(x)`, `eye(x)`
— `1:n` ($(1,2,\dots,n)$), `x1:h:x2` ($(x_1,x_1+h,\dots,x_2)$), `linspace(x1,x2,n)` ($(x_1,\dots,x_2)$ équiréparti, de longueur n)
— `size(A)`, `length(x)`
— `A = fscanfMat('toto.txt')`, lit le fichier `toto.txt` et en extrait la matrice `A` (le fichier texte donne la matrice ligne par ligne, les colonnes étant séparées par des espaces ou des tabulations)

3.2 Extraction de valeurs, sous-matrices

— pour une matrice `A`, `A(i,j)`, `A(i,:)` (toute la ligne `i`), `A(:,j)` (toute la colonne `j`)
— pour un vecteur `x`, `x(i)`, `x(3:$)` (indices de 3 à la fin)
— dans tous les cas, `A(I)` (vecteur des indices spécifiés dans `I`)

3.3 Opérations

— `A'` (transconjuguée de `A`)
— `[A,B]` (matrice formée de `A` et `B` côte-à-côte), `[A;B]` (matrice formée de `A` et `B` l'une au-dessus de l'autre)
— `5+A` (ajoute 5 à chaque coefficient), `5*A` (multiplie chaque coefficient par 5)
— opérations matricielles : `A*B`, `A^2` (carré matriciel), `A^(-1)` (inverse de matrice)
— opérations **coefficient par coefficient** pour matrices ou vecteurs : `A.*B` (matrice $(a_{ij}b_{ij})_{ij}$), `A.^2`, `A.^B` (matrice $(a_{ij}^{b_{ij}})_{ij}$), `1 ./A` (Matrice $(1/a_{ij})_{ij}$, attention à l'espace !), `A./B`, et toutes les fonctions usuelles, `sqrt(A)` (matrice $(\sqrt{a_{ij}})_{ij}$), `sin(A)`, `3^A`, etc.
— `sum(A)`, `sum(A,1)` (vecteur-ligne des sommes des colonnes), `sum(A,2)`, `cumsum(A)` (sommes cumulées), `prod(A)`
— trouver les indices vérifiant une condition : `find((x<0)|(x>9))`
— compter les coefficients vérifiant une condition : `sum((x>0)&(x<3))`
— `gsort(x)` (trier `x` par ordre décroissant), `-gsort(-x)` (par ordre croissant)

3.4 Algèbre linéaire

— `det(A)`, `kernel(A)`
— `[P,D]=spec(A)`
— `linsolve(A,b)` (renvoie une solution de $AX + b = 0$)

4 Représentations graphiques

— `disp('x='); disp(x);` (affiche le texte `x=` puis la valeur de `x` dans la console)

4.1 Gestion des fenêtres graphiques

— `scf(i)` (ouvre la fenêtre numéro `i`)
— `clf` (efface la fenêtre courante)
— `subplot(m,n,k)` (découpe la fenêtre graphique en `m` lignes et `n` colonnes, et rend active la `k`-ième sous-fenêtre dans l'ordre de lecture)

4.2 Graphiques

— `plot(x,y,'option')` représente la courbe formée des points `(x(i),y(i))`, avec un style précisé par `option` (une lettre pour la couleur, parmi `kbrgcym`; et éventuellement une lettre pour le style de courbe, parmi `.+xo-`; par exemple `'r'` dessine en trait continu rouge, `'b.'` dessine des points bleus, etc.)
— `xtitle('titre','axe X','axe Y')`
— `legend(['courbe 1','courbe 2','courbe 3'],2)` donne des légendes pour chaque courbe (dans l'ordre de dessin), et `2` indique la position de la légende (en haut à gauche)
— `isoview(xmin,xmax,ymin,ymax)` (pour avoir la même échelle sur les deux axes)
— `histplot(n,x)` (histogramme des valeurs de `x`, découpées en `n` classes), `histplot(n,x,2)` par exemple pour changer la couleur
— `plot2d2(x,y)` (fonction constante par morceaux, égale à `y(i)` sur [`x(i)`,`x(i+1)`])

5 Programmation

```
// Commentaires
if(x>0)
    ...
else
    ...
end

for i=0:10
    ...
end

while((x<5)|(x>0))
    ...
end

function a=nom_de_la_fonction(x1,x2)
  ...
endfunction  // Renvoie la valeur de la variable "a" à la fin de la fonction
```

6 Probabilités et statistiques

— `mean(x)`, `variance(x)` : moyenne et variance empiriques de `x` (et aussi `mean(A,1)` : moyenne par colonne)
— `rand()` (un réel uniforme dans [0,1]), `rand(m,n)` (un tableau de `m`×`n` réels indépendants uniformes dans [0,1])
— `grand(m,n,'type',paramètres)` (un tableau de `m`×`n` réels tirés selon la loi `type` (`nor`, `poi`, `geom`, `bin`, `exp`, `unf` uniforme continue, `uin` uniforme discrète,...) de paramètres donnés (attention aux paramètres des lois exponentielle et normale)
— `cdfnor`, `cdfpoi`, `cdfchi` (fonction de répartition **et** fonction de répartition réciproque)
— `N=length(x); plot2d2(-gsort(-x),(1:N)/N)` (fonction de répartition empirique d'après les données `x`)
— `[a,b]=reglin(x,y)` : régression linéaire (détermine des matrices `a` et `b` telles que `y`$\simeq$`a*x+b`, minimisant l'erreur quadratique), `[a,b,s]=reglin(x,y)` fournit aussi l'estimateur `s` de l'écart-type, cf. cas gaussien. **Attention :** nombre de données = nombre de *colonnes* de `x` et `y` (donc ce sont des vecteurs-*lignes* dans le cas de la dimension 1)
— `pr=binomial(p,n)` : vecteur des probabilités binomiales `pr(k+1)`$=\binom{n}{k}p^k(1-p)^{n-k}=P(X=k)$ où $X\sim\mathcal{B}(n,p)$ (attention au décalage d'indice)