Introduction à SCILAB

MATLAB vient de MATrix LABoratory, SCILAB signifie SCIentific LABoratory, c'est un langage très *proche* de MATLAB, *libre*, développé par un institut de recherche français : l'I.N.R.I.A. (Institut National de Recherche en Informatique et Automatique), à ne pas confondre avec l'I.N.R.A.
On peut télécharger gratuitement SCILAB en allant sur le site (http ://www.scilab.org/). La version installée au lycée est la version 4.1.2. Elle existe pour Windows, mais aussi pour Linux et Mac OS X.

1 Mise en route

$-->$ est l'invite pour SCILAB, il apparaît automatiquement au début de chaque ligne et on ne le tape pas.

⋆ MATLAB et SCILAB sont des langages interprétés – contrairement à PASCAL qui est un langage compilé – on peut donc taper directement une commande (expression ou instruction), c'est à dire l'utiliser comme une calculatrice. On peut également utiliser l'éditeur intégré pour mettre au point des programmes ou créer de nouvelles fonctions.

⋆ La présente introduction se limite à SCILAB, les différences de notation entre les deux langages sont minimes.

⋆ Dans une première partie, on passera en revue les principaux éléments de construction, puis dans une deuxième partie, on étudiera le fonctionnement de l'éditeur. Enfin, on s'intéressera brièvement au format de stockage des nombres.

2 Eléments de construction :

les éléments de construction des expressions SCILAB sont :

- les nombres (réels ou complexes)
- les variables
- les opérateurs
- les fonctions

2.1 Nombres réels ou complexes

taper successivement :
$-->7\wedge 4$, $-->-3$, $-->2.75*10\wedge(-12)$, $-->5*3$, $-->2-3*\%i$, $-->$ sqrt (3), $-->3/4$, $-->$ sin (% pi /6), $-->$ log(10), $-->$ log10(10), $-->$ log2(10), $-->$ exp(10)

2.2 Opérateurs

Les opérateurs arithmétiques de base sont : $+$ $-$ $*$ $/$ $\wedge$ ainsi que les opérateurs terme à terme correspondants sur les vecteurs ou les matrices : $.*$ $./$ $.\wedge$ ($+$ et $-$ restent inchangés pour les opérations terme à terme).

2.3 Variables

- On ne déclare pas les variables, sauf exception ("persistent variable").
- Une variable peut être un nombre, mais aussi un vecteur, une matrice, une chaîne de caractères etc...
- Tout nouveau nom crée une nouvelle variable.
- Un nom de variable est constitué d'une lettre suivie d'un ensemble de caractères alphanumériques, par exemple «longueur», «alpha», «x1», «Volume», «abscisse»...
- SCILAB n'utilise que les 24 premiers caractères et distingue les minuscules des majuscules.
- $=$ est l'opérateur d'affectation.
- Pour afficher un nombre ou une matrice associé à une variable, il suffit d'entrer le nom de la variable.

Exemples :

⋆ On crée les variables suivantes par affectation d'une valeur :
$-->\ a=2$, $-->\ b=3\wedge 4$, $-->\ c=\%pi/2$

⋆ On peut effectuer des calculs avec les variables ainsi créées : $-->\ a+b$, $-->\ a/(b-c)$

⋆ On peut également affecter les résultats de ces calculs à de nouvelles variables : $-->$ Somme $=a+b$,
$-->$ Quotient $=a/(b-c)$

⋆ Le caractère point virgule (;) empêche l'affichage à l'écran, par exemple :
- effectuer successivement $-->a=12$
$-->b=4$;
$-->a*b$

⋆ Toute nouvelle affectation efface la précédente, en effet :
- que remarque-t-on si on exécute 3 fois de suite la commande : $-->a=a+2$?
- que se passe-t-il si l'on exécute successivement : $-->a=5$
$-->b=2$
$-->a=b$;
$-->a$

⋆ Il existe une variable prédéfinie, notée *ans*, qui est automatiquement créée par SCILAB, elle contient la valeur de l'expression courante, elle est modifiée par tout nouveau calcul et peut être réutilisée ;

effectuer par exemple : $-->a*b$
$-->ans+c$
$-->b*ans$

Taper successivement *who*, *whos*, *who_user*, *clear a*, *who_user*, *clear*, *who_user* ; conclusion ?

⋆ Quelques variables prédéfinies :

ans	dernière réponse	*%inf*	infini (supérieur à $2^{1024}\simeq 1,7977.10^{308}$)
%eps	précision virgule flottante (2^{-52})	*isinf*	vrai si infini
%e	2,718218...	*%i*	$\%i^2=-1$
%pi	3,14159265...	*%nan*	pas un nombre (not a number)
isnan	vrai si Not-a-number		plus petit réel ($2^{-1023}\simeq 2,551.10^{-308}$)
%T	True (vrai, booléen)	%F	False (faux, booléen)

2.4 Fonctions

⋆ Elles seront étudiées au fur et à mesure, on en a déjà rencontré quelques unes. Toutes les fonctions mathématiques que vous connaissez sont prédéfinies ; pour avoir la liste des fonctions élémentaires de SCILAB, ou pour connaître leur syntaxe, on peut consulter l'aide (*help*) et aller dans "Elementary functions". Il existe bien d'autres fonctions, par exemple les fonctions entrées-sorties, mais de toute façon seule une petite partie des fonctions SCILAB est au programme.

D'une façon générale, une fonction s'utilise en tapant le nom de la fonction, puis le ou les arguments (séparés par des virgules s'il y en a plusieurs) placés entre parenthèses. Ex : *exp*(3), *sin*(% pi /6).

Voici deux autres exemples :

⋆ Fonctions *fix*, *floor*, *ceil*, *round* : les appliquer aux nombres suivants : 3, −5, 3.4, 3.8, −3.8, −3.4 ; que font-elles ?

⋆ La fonction *rand* est un générateur de nombres aléatoires ; SCILAB peut utiliser au choix une loi uniforme sur [0 , 1], soit une loi normale centrée réduite (programme de deuxième année) : *rand*('uniform') ou *rand*('normal'). Par défaut, c'est en général la loi uniforme qui s'applique, pour s'en assurer, taper $-->$ *rand*('info'), la réponse devrait être *uniform* ; dans ce cas *rand* donne le même résultat que *rand*('uniform').

3 Vecteurs et matrices

3.1 Matrice (ou vecteur) ligne

$-->v=[1\quad 2\quad 3\quad 5]$, $-->u=[4\ ,\ 0\ ,\ -1\ ,\ \sin(\%pi/5)]$
Les composantes sont séparées par des blancs (un ou plusieurs) ou bien par des virgules.
On peut aussi définir un vecteur ligne de deux autres manières :
taper $-->w=-3:4$, $-->x=4:-3$, $-->y=-3.5:5.1$, $-->h=0:0.1:1$,
$-->z=6:-.5:3.4$, $-->t=10:-3:-6$ que remarque-t-on ?
puis $-->W=\text{linspace}(0,1,10)$, $-->X=\text{linspace}(-1,4,7)$, $-->Y=\text{linspace}(1,0,10)$
$-->T=\text{linspace}(0,1)$; il existe également une fonction *logspace* : $-->Z=\text{logspace}(x_1,x_2,n)$ fournit un vecteur dont les n composantes sont logarithmiquement réparties entre 10^{x_1} et 10^{x_2}.

3.2 Matrices à plusieurs lignes

Les lignes sont séparées par des points virgules :
$-->m=[1\ ,\ 2\ ,\ 3\ ;4\quad 5\ ,\ 7\ ;9\quad 0\quad \text{sqrt}(3);-6\quad \log(2)\quad 8]$ (matrice à 4 lignes et 3 colonnes).

Remarques 3.1.

- ⋆ *Si on souhaite rentrer une matrice dont les lignes sont assez longues, on peut faire un retour chariot aprés une ligne, par exemple :*
  $-->\ m = [1\ ,\ 2\ ,\ 3\ ;\ ..$
  $-->\ 4\quad 5\ ,\ 7\ ;9\quad 0\quad sqrt(3); -6\quad log(2)\quad 8]$
  Ici on a entré la première ligne, puis les 3 dernières lignes ensemble.
- ⋆ *De même pour n'importe quelle instruction, on peut la rentrer en plusieurs étapes ; pour cela il faut finir la ligne d'instruction par 2 points successifs, exemple :*
  $-->\ m = [1\ ,\ 2\ ,\ 3\ ;\ 4\quad 5\ ,\ ..$
  $-->\ 7\ ;9\quad 0\quad sqrt(3); -6\quad log(2)\quad 8]$
  Ici, on a coupé l'instruction en cours de ligne ; pour utiliser .. il n'est pas nécessaire d'avoir à faire à une matrice.

3.3 Matrices particulières

$-->\ n = \text{ones}\,(2,5)$, $-->\ p = \text{ones}\,(1,3)$, $-->\ q = \text{zeros}\,(4,3)$, $-->\ r = \text{eye}\,(6,6)$ créent des matrices dont tous les cœfficients sont soit des 1, soit des 0 ou bien définissent la matrice carrée identité de taille voulue. Que se passe-t-il si l'on tape $-->\ s = \text{eye}\,(4,7)$? comment obtenir une matrice dont tous les cœfficients sont égaux à 2 ? à $\sqrt{3}$?

3.4 Matrices diagonales

La fonction *diag* permet de fabriquer des matrices diagonales à partir d'un vecteur ligne ou d'un vecteur colonne, ou d'extraire la diagonale d'une matrice carrée. Appliquer cette fonction successivement à : un vecteur ligne, un vecteur colonne, une matrice carrée, une matrice quelconque ; que fait la fonction *diag(diag)* appliquée à un vecteur ? à une matrice carrée ? à une matrice quelconque ?

La fonction *diag* admet un deuxième argument facultatif qui est un entier relatif :
que donne $-->\ \text{diag}(m,1)$ appliqué à une matrice ? à un vecteur ? même question avec $-->\ \text{diag}(m,-2)$?

3.5 Matrices triangulaires

- ⋆ La fonction *triu* permet d'extraire une matrice triangulaire supérieure (u pour upper) et la fonction *tril* permet d'extraire une matrice triangulaire inférieure (ℓ pour lower) ; essayer tout d'abord avec une matrice carrée, puis quelconque. Qu'obtient-on pour un vecteur ligne ? pour un vecteur colonne ?
- ⋆ Que donne $-->\ \text{triu}(m,1)$? $-->\ \text{triu}(m,-2)$? mêmes questions avec *tril*.
- ⋆ Qu'obtient-on en composant les fonctions *triu*, *tril*, *diag* dans différents ordres ?

3.6 Transposée

Le caractère apostrophe ' est l'opérateur de transposition : par exemple, si v est un vecteur ligne, alors v' est un vecteur colonne ; et pour une matrice quelconque m, m' est la matrice transposée.

- ⋆ Vérifier sur quelques exemples la formule ${}^t(A.B) = {}^t B.{}^t A$.
- ⋆ Vérifier également que $A.{}^t A$ et ${}^t A.A$ sont des matrices symétriques.
- ⋆ Qu'en est-il de $A + {}^t A$, de $A - {}^t A$?

3.7 Opérations sur les matrices

- ⋆ Effectuer la somme, la différence de deux matrices de même format.
- ⋆ Effectuer le produit d'une matrice par un réel, exemple : $-->\ \text{sqrt}\,(3) * m$,
  ou par un complexe, exemple : $-->\ (2 + 5 * \%i) * n$
- ⋆ Effectuer le produit de deux matrices de formats compatibles, par exemple $>>\ v * m$. Vérifier qu'il s'agit bien du produit matriciel vu en cours de mathématiques.
- ⋆ Opérations terme à terme : Pour A et B deux matrices de même format (non carrées) ; vérifier que le produit $>>\ A.*B$ a un sens alors que $-->\ A*B$ n'en a pas. Si C et D sont deux matrices carrées de même format, vérifier que $-->\ C.*D$ et $-->\ C*D$ existent mais ne donnent pas le même résultat.
  On peut de même, si aucun cœfficient de B ou de D n'est nul, effectuer $-->\ A./B$ et $-->\ C./D$
  Attention : quels que soient les formats, on n'a pas défini $-->\ A/B$

3.8 Fonctions de matrices

⋆ Toutes les fonctions prédéfinies peuvent s'appliquer à des vecteurs ou des matrices aussi bien qu'à des réels ou des variables scalaires (sous réserve toutefois de compatibilité avec l'ensemble de définition).
Essayer par exemple $-->\ \sin(m)$, $-->\ \mathrm{sqrt}(v)$, $-->\ \exp(W)$

⋆ La fonctions *rand* – que l'on vue au paragraphe 2.4 utilisée avec un seul argument : 'uniform'– ou 'normal'– s'utilise aussi avec plusieurs variables pour fournir des matrices aléatoires : $-->$ *rand*(3,4) fournit une matrice aléatoire à 3 lignes et 4 colonnes d'éléments de $[0, 1]$ suivant une loi uniforme ; $-->$ *rand* (5,5,'normal') fournit une matrice aléatoire carrée de format 5×5, d'éléments de $\mathbb{R}$, suivant une loi normale centrée réduite.

⋆ Pour un vecteur comme pour une matrice quelconque, la fonction *size* fournit dans l'ordre le nombre de lignes, puis le nombre de colonnes ; *length* donne le nombre total d'éléments.
Enfin, *size*$(A, 1)$ donne juste le nombre de lignes et *size*$(A, 2)$ le nombre de colonnes de A et pour une matrice ou un vecteur, *length*(A) est équivalent à *size*$(A,"\star")$.

⋆ Le symbole \$ représente le dernier élément d'une matrice ou d'un vecteur (essayer $m(\$)$, $v(\$)$).

⋆ Les fonctions *sum*, *prod* et *diff* s'appliquent à un vecteur, à une matrice ; que font-elles ?

⋆ Que font les fonctions *cumsum*, *cumprod* appliquées à une matrice ? à un vecteur ?

⋆ Comment, avec ces fonctions, calculer la moyenne arithmétique de n termes ? la moyenne géométrique de n termes positifs ?

⋆ Soient u et v deux vecteurs lignes de même longueur ; comment peut-on obtenir le produit scalaire de u et v en utilisant la transposition ?

3.9 Construction de matrices à partir d'autres matrices

À partir d'un vecteur ligne v et d'une matrice m, construire les matrices ou vecteurs suivants : $-->[v \quad v]$, $-->[v\ ;\ v]$, $-->[m \quad m]$, $-->[m\ ;\ m]$. A quelle condition sur les formats des matrices m_1 et m_2 les instructions $-->[m_1\ ,\ m_2]$ et $-->[m_1\ ;\ m_2]$ ont-elles un sens ?

3.10 Extraction de matrices ou de vecteurs

⋆ Extraire un élément : $-->A(3, 4)$ extrait l'élément se trouvant en troisième ligne et quatrième colonne de la matrice A ; que se passe-t-il si A possède moins de 3 lignes ou moins de 4 colonnes ?

⋆ Extraire une ligne ou une colonne : $-->A(3, :)$ extrait la troisième ligne de A et $-->A(:, 4)$ extrait la quatrième colonne.

⋆ Extraire une sous-matrice : $-->A(2 : 4, 3 : 6)$ extrait (lorsque c'est possible) la sous-matrice de A formée des éléments situés aux intersections des lignes 2 à 4 et des colonnes 3 à 6.
Comment extraire une sous-matrice formée à partir de lignes et (ou) de colonnes de A non consécutives ?

⋆ Supression ou ajout d'une ligne ou d'une colonne : pour supprimer la troisième ligne de A, on tape : $-->A(3, :) = [\,]$ (matrice vide), de même pour supprimer la deuxième colonne : $-->A(:, 2) = [\,]$
Pour ajouter une ligne ou une colonne, il suffit d'en énumérer les éléments : par exemple si A est carrée d'ordre 4, $A(5, :) = [1 \quad 1 \quad 1 \quad 1]$ rajoute une ligne de 1 ; que se passe-t-il si on tape $A(2, :) = [1 \quad 1 \quad 1 \quad 1]$ pour une matrice carrée d'ordre 4 ?

3.11 Exercice de construction de matrices

1. Construire à partir des fonctions vues précédemment :

$$m = \begin{pmatrix} 1 & -3 & 2 & 0 & 0 & 0 \\ 4 & 1 & -3 & 2 & 0 & 0 \\ 0 & 4 & 1 & -3 & 2 & 0 \\ 0 & 0 & 4 & 1 & -3 & 2 \\ 0 & 0 & 0 & 4 & 1 & -3 \\ 0 & 0 & 0 & 0 & 4 & 1 \end{pmatrix} \quad , \quad n = \begin{pmatrix} 2 & 2 & 3 & 3 & 3 & 8 \\ 2 & 2 & 3 & 3 & 3 & 8 \\ 4 & 4 & 3 & 3 & 3 & 8 \\ 4 & 4 & 3 & 3 & 3 & 8 \\ 4 & 4 & 6 & 6 & 6 & 6 \end{pmatrix} \quad , \quad p = \begin{pmatrix} 2 & 2 & 3 & 3 & 3 & 8 \\ 2 & 2 & 3 & 3 & 3 & 8 \\ 6 & 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 3 & 3 & 8 \\ 4 & 4 & 6 & 6 & 6 & 6 \end{pmatrix} .$$