

09 févr. 15 11:41	scilab-intro.txt	Page 1/8
1) Introduction		
- matlab => scilab et octave : versions libres origine INRIA		
- logiciel de calcul matriciel de représentation graphique + un peu de calcul symbolique (fractions rationnelles)		
- interprété, sauf les primitives, mais possibilité d'appeler du fortran ou du C pour les calculs lourds		
- programmable : - scripts *.sce => exec - fonctions *.sci => getf		
2) Premiers pas		
Scalaire		
a = 1	affectation et affichage	
b = 2 ;	affectation sans affichage	
a + b		
ans	variable par défaut	
	stockant le dernier résultat	
who	liste des variables	
clear a; who	supprime a	
%pi	constante pi	
%e	constante e	
%eps	précision (double du fortran)	
%i	sqrt(-1) => complexes	
exp(1)/%e	e/e = 1	
1 - 2 * %i	= 1 - 2i	
exp(-%i * %pi)	-1 + partie imaginaire en 10 ⁽⁻¹⁶⁾	
//	introducteur de commentaires	
Chaînes de caractères		
'message'		
"commande"		
"ls "+SCI+"/macros/elem"	concaténation avec +	
Opérateurs		
+, -, *, /, ^ ou ** (élévation à la puissance)		

09 févr. 15 11:41	scilab-intro.txt	Page 2/8
2) Vecteurs/Matrices : constructeurs		
V = [1 10 100]	vecteur ligne ! 1 10 100 !	
W = [1; 10; 100]	vecteur colonne ! 1 ! ! 10 ! ! 100 !	
M = [1, 2, 3; 4, 5, 6]	matrice (2,3) ! 1 2 3 ! ! 4 5 6 !	
M = [1, 2, 3; ... 4, 5, 6]	continuation de ligne	
Autres générateurs de vecteurs et matrices		
u = début [:pas] : fin	pas de 1 par défaut problème d'arrondi si non entier	
u = linspace(début, fin, nombre)	préféré vecteur en progression arithmétique	
u = logspace(ldéb, lfin, nombre)	où début =10 ^{ldéb} fin=10 ^{lfin} vecteur en progression géométrique	
eye(n_lignes, p_colonnes)	= matrice de 0 avec diagonale de 1	
zeros(n_lignes, p_colonnes)	avec des 0 partout	
ones(n_lignes, p_colonnes)	avec des 1 partout	
rand(n_lignes, p_colonnes)	tirage aléatoire sur [0, 1[
M = diag(u)	diagonale avec les valeurs de u	
v = diag(M)	vecteur des éléments de la diagonale	
triu(M)	triangulaire supérieure	
tril(M)	triangulaire inférieure	
Opérations sur les matrices		
Opérations globales		
A + B	somme terme à terme	
A * B	produit matriciel ! diff. de f90	
A \ B	division matricielle	
A'	ajointe de A si complexe transposée de A si réelle	
x = linspace(...)	vecteur ligne	
y = linspace(...)		
x * y'	produit scalaire (lign x col)	
x' * y	produit "extérieur" = matrice	
expm(A)	exponentielle matricielle	

09 févr. 15 11:41	scilab-intro.txt	Page 3/8
Opérations terme à terme		
avec un scalaire	M + a	
	M * a	
	M / a	
avec une fonction	fct(M) => exp(M), sqrt(M)	
entre deux matrices	=> interposer un point "."	
pour forcer une opération terme à terme		
	M .* N	
	M ./ N	
	M.^2 => ((m_{ij})^2))	
	a.^M => ((a^{m_{ij}}))	
Fonctions opérant sur les matrices		
Réduction:		
size(M)	= [n_lignes, n_colonnes]	
size(M, 'c')	= size(M, 2) = n_colonnes (columns)	
size(M, 'r')	= size(M, 1) = n_lignes (rows)	
length(M)	= size(M, '*') nb total d'éléments	
sum(M)	somme = scalaire	
sum(M, 'r')	selon les lignes => vecteur colonne	
sum(M, 'c')	selon les colonnes => vecteur ligne	
cumsum(M)	matrice des sommes cumulées	
prod(M)	produit = scalaire	
cumprod(M)	matrice des produits cumulés	
diag(M)		
norm(M)		
Sous-matrices		
M(2,:)	2ème ligne	
M(1:2,:)	lignes 1 et 2 (début:pas:fin) diff. de f90	
M([1,3], [2,3])	éléments de M dont les indices sont donnés par la matrice d'indexation	
Restructuration		
matrix(mat, n, m)	ou mat(vect, n, m) réarrange la matrice mat en empilant ses colonnes ou le vecteur vect => matrice n lignes x m colonnes	

09 févr. 15 11:41	scilab-intro.txt	Page 4/8
Booléens		
%t %f	opérateurs booléens & (et) (ou) ~ (non)	
tests logiques	== < > <= >=	
	~= ou <> (différent)	
M = rand(3,5)		
M < .4	=> matrice de booléens	
find(M < .4)	=> vecteur des indices où le booléen est vrai (utilisé pour les boucles)	
3) Structures de contrôle		
if condition then	
else	
end		
aussi avec elseif		
select var	case expr1	
	
	case expr2	
	
	[else]	
	
end		
for var = matrice	
end		
while condition	
end		

09 févr. 15 11:41	scilab-intro.txt	Page 5/8
4) Fonctions		
Résultat matriciel => entre []		
a) Définitions "courtes" avec deff deff('y1, y2, ...)=ma_fct(x1, x2, ...)', y1=...)		
b) Création d'un fichier de fct d'extension .sci function [y1, y2, ...] = ma_fct(x1, x2, ...) y1 = ... ; // ne pas afficher les calculs y2 = ... ; // internes à la fonction endfunction puis getf('fichier.sci') recharger la fonction à chaque modification		
c) Remarques Passage d'arguments par référence si modifiés mais par copie si en lecture seule. Les variables internes sont locales à la fonction error("message") impression d'un message et arrêt warning("message") impression d'un message		
d) Appel [u1, u2, ...] = ma_fct(t1, t2, ...) attention: sans variable pour stocker le résultat seule la lère composante est récupérée par ans ma_fct(t1,t2,...) => affiche u1 seul (u2 perdu)		

lundi 09 février 2015

5/8

09 févr. 15 11:41	scilab-intro.txt	Page 6/8
5) Graphique		
courbes y = f(x) plot2d(x,y,...) attend des vecteurs colonnes (penser à transposer plot2d(linspace(...)', ...) ou des matrices pour 2 courbes par exemple plot2d([x,x], [y1, y2], leg='légende1@légende2') arguments optionnels : style, leg, rect (limites), logflag ex: logflag="nl" pour du semi-log xtitle("titre", "libellé axe x", "libellé axe y")		
contour2d(x, y, z, nz) courbes de niveau (valeurs dans z et nz niveaux) fcontour2d(x, y, f, nz) courbes de niveau de la fonction f		
fchamp tracé d'un champ de vecteurs		
gestion des fenêtres graphiques xset('window', 2) => 2 devient la fenêtre courante xdel(2) ferme la fenêtre graphique 2 xbasc(2) efface la fenêtre graphique 2 xbasc() efface la fenêtre graphique courante		
Attention: les versions récentes de scilab proposent un nouveau mode graphique utilisant une structure hiérarchique d'objets. Par exemple, pour examiner les propriétés des axes on accédera à cet objet par axes=get("current_axes") Puis pour modifier le libellé de l'axe x on pourra écrire axes.x_label.text="axe x" puis sa taille axes.x_label.font_size=4 puis sa position axes.x_location="top"		
Exportation en format eps (postscript encapsulé) Menu File -> Export -> Postscript indiquer le nom sans le suffixe (.eps)		

6/8

lundi 09 février 2015

09 févr. 15 11:41	scilab-intro.txt	Page 7/8
6) Probabilités et statistique		
grand = générateur aléatoire, plusieurs lois : beta, binomiale, chi2, gamma, Gauss, Poisson, uniforme... cdf* = fonctions de répartition de nombreuses lois + outils statistiques histplot : calcule et dessine un histogramme		
7) Divers		
help fonction	manuel de la fonction	
apropos mot-clef	recherche par mot-clef	
whos()	liste longue des variables	
whos -type type1	de type type1	
whos -name name1	de nom nom1*	
diary('journal')	journal de la session	
diary(0)	arrête le journal	
whereis(fonction)	localise la macro, mais pas les primitives	
pwd()	affichage du répertoire courant	
getcwd()	affichage du répertoire courant	
chdir("chemin")	changement du répertoire courant	
unix_w("commande")	passé une commande au système avec retour dans la fenêtre	
unix_w("ls "+SCI+"/macros/elem/interpln.sci")		
Listes = équivalent des structures		
Lancement en mode console sans fenêtre graphique (sans java en v5) scilab -nwni (no window)		

lundi 09 février 2015

7/8

09 févr. 15 11:41	scilab-intro.txt	Page 8/8
8) Fichiers		
unit = file("open", "fic", "old"/"new") ouvre le fichier "fic" existant(old)/à créer(new) et lui associe le numéro unit jusqu'au close file("close", unit) ferme le fichier associé à l'unité unit => à faire le plus rapidement possible		
lecture en format libre matrice = read("fic" ou unité, nb_lignes, nb_col) nb_lignes=-1 => jusqu'à la fin du fichier		
lecture de chaînes de caractères => format A texte=read("fic" ou unité,nb_lignes,nb_col,"(A)")		
NB: read directement sur un fichier sans passer par file puis le numéro d'unité => on lit toujours à partir de la lère ligne		
9) Calcul Symbolique: Polynômes et fractions rationnelles		
z = poly(0,"z") => monôme de racine 0 p = poly([rac1, rac2, ...], "z") => polynôme défini par ses racines p = poly([a0, a1, ...], "z", "coeff") => polynôme défini par ses coefficients p = a0 + a1*z + a2*z^2 + ... v = roots(p) => vecteur des racines (complexes) c = coeff(p) => vecteur des coefficients du polynôme y = horner(p,x) évalue le polynôme p en x		
f = p/q => fraction rationnelle symbolique df = derivat(f) dérivée de la fraction rationnelle f		
10) En savoir plus		
Une introduction à Scilab de Bruno Pinçon (ancien : révision incomplète sur les graphiques) http://www.iecn.u-nancy.fr/~pincon/scilab/scilab.html		
La page d'accueil de scilab (versions 5.* récentes) http://www.scilab.org/ (voir "Ressources")		

8/8

lundi 09 février 2015