Dictionnaire Pascal - Scilab

Variables et instructions	Pascal	Scilab	Remarques
Déclaration d'une variable	Nécessité de déclarer les variables	Pas de déclaration de variables	Scilab est un langage interprété (sans compilation), les variables sont automatiquement créées au moment de leur utilisation
Affectation	:= Exemple : `x :=x+1;`	= Exemple : `x=x+1;`	Cette instruction dans Scilab crée automatiquement la variable `x`
Affectation et écriture du résultat à l'écran	`x := 3;` `write(x);`	`x = 3`	Si l'on n'écrit pas de point-virgule, Scilab écrit automatiquement le résultat
Affectation sans écriture du résultat	`x := 3;`	`x = 3;`	Cette instruction dans Scilab crée automatiquement la variable `x`
Lecture à l'écran de la valeur de x	`write('x : ', x)` `read(x);`	`x=input('x : ');`	L'unique instruction de Scilab correspond aux deux instructions de Pascal
Affichage à l'écran d'une chaîne de caractères	`write('Resultat : ')`	`disp('Résultat : ') ;`	Le Pascal n'accepte pas les accents, Scilab les accepte
Fonctions usuelles et Pi	abs, sqrt, ln, exp, cos sin, trunc, Pi	abs, sqrt, log, exp, cos, sin, floor, %pi	En Scilab, log désigne le logarithme népérien
Insertion d'un commentaire	{commentaire}	// commentaire	Le commentaire est introduit à droite
Opérations élémentaires	$+, -, *, /$	$+, -, *, /, \hat{}$	Scilab dispose d'une opération puissance (ˆ)
Comparaison et tests	$=, <, <=, <>$	$==, <, <=, \tilde{} =$	
Opérateurs logiques	`and`, `or`	&, |	
Instruction if ... then	if (x = 0) then s :=s+1 ;	if (x = = 0) then s=s+1 ; end	Si x est égal à 0 alors on rajoute 1 à s

Instruction if ... then ... else	if (x = 0) then
	 s :=s+1
	 else
	 s :=s-1 ;
		if (x = = 0) then
		 s=s+1 ;
		 else
		 s=s-1 ;
		 end
			Si x est égal à 0 (test) alors on incrémente s de 1, sinon on décrémente s de 1

Boucle for	S :=0 ;
	 for k :=1 to n do
	 begin
	 S :=S+1/k ;
	 end ;
		S=0 ;
		 for k=1 :n
		 S=S+1/k ;
		 end
			Calcul de la somme partielle d'indice n de la série harmonique

Boucle while	r :=0 ;
	 while (r=0) do
	 begin
	 u :=random ;
	 n :=n+1 ;
	 if(u<p) then r :=1 ;
	 end ;
	 write('n : ', n) ;
		r=0 ;
		 while (r= =0) do
		 u=rand() ;
		 n=n+1 ;
		 if(u<p) then
		 r=1 ;
		 end
		 end
		 disp('n : ') ; disp(n) ;
			Simulation d'une variable aléatoire suivant la loi géométrique de paramètre p, r désigne le résultat du lancer, u suit la loi uniforme sur $[0, 1]$ n est le rang du premier succès, l'événement $[u < p]$ se produit avec une probabilité p

Boucle repeat until	repeat ... until (...)	Pas d'équivalent en Scilab	En Scilab on utilisera l'instruction while
Loi uniforme sur $[0, 1]$	U :=random ;	U :=rand() ;	
Loi uniforme sur $[[1, n]]$	U :=random(n)+1 ;	U :=grand(1,1,"uin",1,n) ;	M=grand(3,7,"uin",1,n) ; permet de de créer une matrice avec 2 lignes et 7 colonnes dont les coefficients sont égaux aux valeurs prises par une variable aléatoire suivant la loi uniforme sur $[[1, n]]$

Tableaux, matrices Pascal & Scilab

Variables et instructions	Pascal	Scilab	Remarques
Déclaration d'une variable tableau	tableau : array[1..n] of real mat : array[1..n,1..p] of real	Pas de déclaration	
Tableau (à une ligne)	X[0] :=2 ; X[1] :=3 ; X[2] :=0 ;	X=[2, 3, 0] ;	L'instruction dans Scilab crée automatiquement le tableau $X = \begin{pmatrix} 2 & 3 & 0 \end{pmatrix}$
Matrice	A[1,1] :=2 ; A[1,2] :=3 ; A[2,1] :=0 ; A[2,2] :=1 ; A[3,1] :=4 ; A[3,2] :=-1 ;	A=[2, 3 ; 0, 1 ; 4,-1] ;	Création de $A = \begin{pmatrix} 2 & 3 \\ 0 & 1 \\ 4 & -1 \end{pmatrix}$
Tableau à une ligne ou vecteur ligne (1)	Pas d'équivalent	A=0 :2 :10 ; A=linspace(0,10,6)	Création d'un tableau à une ligne, 6 colonnes, formé des nombres entiers $0, 2, 4, 6, 8, 10$
Vecteur ligne à n colonnes (2)	Pas d'équivalent	L=zeros(1,n) ; L=ones(1,n) ;	Création d'un vecteur ligne $L = (0, 0, \ldots, 0)$ ou $L = (1, 1, \ldots, 1)$
Vecteur colonne à n lignes	Pas d'équivalent	C=zeros(n,1) ; C=ones(n,1) ;	Création des vecteurs colonne correspondants
$k^{ème}$ élément d'un vecteur ligne ou colonne	X[k]	L(k) C(k)	
Taille du tableau	Pas d'équivalent	length(L)	Donne le nombre d'éléments du tableau L
Matrices	Pas d'équivalent	M=zeros(n,p) ; M=ones(n,p) ; M=eye(n,p) ;	$M = (0)_{1\leq i\leq n, 1\leq j\leq p}$ $M = (1)_{1\leq i\leq n, 1\leq j\leq p}$ $M = I_n$
Coefficient d'une matrice	M[1,2]	M(1,2)	
Extraction d'une ligne	Pas d'équivalent	L=A(3,1 :2)	Extraction de la troisième ligne de A, on obtient : $L = \begin{pmatrix} 4 & -1 \end{pmatrix}$
Extraction d'une colonne	Pas d'équivalent	C=A(1 :3,2)	Extraction de la deuxième colonne de A
Opérations sur les matrices	Pas d'équivalent	$+,\ -,\ *,\ /,\ \hat{}$ $.*,\ ./,\ .\hat{}$	$*$ est utilisé pour le produit matriciel ou le produit par un scalaire, $.*$ donne le produit coefficient par coefficient
Fonctions matricielles	Pas d'équivalent	rank(M), size(M), inv(M), transpose(M)	rang, taille, inverse, transposée
Tracé de la courbe représentative d'une fonction	Pas d'équivalent	x=[-2 :0.1 :4] ; plot(x,x^2+1)	Tracé de la courbe représentative de $x \mapsto x^2 + 1,$ pour $x \in [-2, 4]$ 0.1 corresepond au pas choisi
Tracé d'une ligne brisée	Pas d'équivalent	x=[0,1,2,3] ; y=[2,5,7,1] ; plot(x,y)	
Tracé de plusieurs courbes avec couleurs	Pas d'équivalent	x=[-2 :0.1 :4] ; plot(x,x^2+1,'red', x,2*x,'black')	On trace sur un même graphique les courbes de $x \mapsto x^2 + 1$ et $x \mapsto x^2 + 1$. x=[-2 :0.1 :4] peut être remplacé par x=linespace(-2,4,60)
Tracé de courbes avec légende	Pas d'équivalent	fonction plot2d	On peut choisir la taille de la fenêtre d'affichage
Histogramme (1)	Pas d'équivalent	x=rand(1,1000,'normal') ; histplot(20,x)	x est un vecteur ligne contenant 1000 valeurs prises par une variable aléatoire suivant une loi normale centrée réduite 20 est le nombre de classes
Histogramme (2)	Pas d'équivalent	x=rand(1,1000,'normal') ; c=linespace(-5,5,100) histplot(c,x)	c est vecteur ligne définissant les classes

Remarque : La syntaxe précise de toutes les fonctions avec leur arguments est accessible dans l'aide de Scilab

Calcul matriciel

Commande	Scilab	Remarques
Tableau à une ligne ou vecteur ligne	`A=0 :2 :10 ;` A=linspace(0,10,6)	Création de $A = \begin{pmatrix} 0 & 2 & 4 & 6 & 8 & 10 \end{pmatrix}$ Le vecteur obtenu comporte 6 valeurs Le segment [0,10] est divisé en 5 sous-segments
Matrice quelconque	A=[2, 3, 0 ;1, 4,-1] ;	Création de $A = \begin{pmatrix} 2 & 3 & 0 \\ 1 & 4 & -1 \end{pmatrix}$
Matrices particulières Initialisation	M=zeros(n,p) ; M=ones(n,p) ; M=eye(n,n) ;	$M = (0)_{1\leq i\leq n, 1\leq j\leq p}$ $M = (1)_{1\leq i\leq n, 1\leq j\leq p}$ $M = I_n$
Longueur d'un vecteur ligne ou colonne	Longueur = length(L)	L est un vecteur ligne ou colonne Longueur est un entier
Taille de la matrice quelconque	[nb_ligne,nb_col]=size(A)	size renvoie une matrice ligne comportant deux coefficients Dans l'exemple ci-dessus on aurait : [nb_ligne,nb_col]=[2, 3]
Coefficient d'une matrice	A(i,j) = 2	On affecte la valeur 2 au coefficient de la matrice A situé à la i^{ème} ligne et à la j^{ème} colonne
Extraction d'une ligne Extraction d'une colonne	L=A(2,1 :3) ou L=A(2, :) C=A(1 :2,3) ou C=A(:,3)	Extraction de la deuxième ligne de la matrice A, on obtient : $L = \begin{pmatrix} 1 & 4 & -1 \end{pmatrix}$ Extraction de la troisième colonne de A. : désigne toutes les lignes (ou toutes les colonnes)
Extraction - exemple	v=[1,3] B=A(:,v)	Extraction des colonnes 1 et 3 de la matrice A, le résultat est : $B = \begin{pmatrix} 2 & 0 \\ 1 & -1 \end{pmatrix}$
Extraction - cas général A $\in \mathcal{M}_{n,m}(\mathbb{R})$ B $\in \mathcal{M}_{k,p}(\mathbb{R})$	u=[u_1,u_2,. . . ,u_k] v=[v_1,v_2,. . . ,v_p] B=A(u,v)	La matrice B est constituée des éléments de A situés aux lignes u_1,u_2,. . . ,u_k et aux colonnes v_1,v_2,. . . ,v_p.
Concaténation horizontale Concaténation verticale	C=[A,B] E=[F ;G]	Si nb_lignes_A = nb_lignes_B Si nb_colonnes_A = nb_colonnes_B
Tri des éléments	gsort(A,'r','i')	Tri sur les lignes ('r' pour "row"), par ordre croissant ("increase")
Opérations Opérations pointées	$+,\ -,\ *,\ /,\ \hat{}$ $.*,\ ./,\ .\hat{}$	Opérations usuelles, ˆ désigne la puissance Opérations coefficient par coefficient
Fonctions matricielles	rank(M), inv(M), M'	rang, inverse, transposée (adjointe)
Utilisation d'une fonction	B = f(A)	Si $A = (a_{i,j})$ alors $B = (f(a_{i,j}))$ où f est une fonction numérique comme $\cos, \exp, \log$ etc.
Résolution d'un système	X=A\B (division à gauche)	Résolution de $AX = B$
Commande find	A=rand(2,3) x=find(A<0.5)	Renvoie les numéros (cf. (*)) des éléments vérifiant la condition
Valeurs propres	val_propres=spec(A)	val_propres est un vecteur colonne dont les coefficients sont les valeurs propres de A
Diagonalisation d'une matrice carrée	[vect_p,mat_diag]=spec(A) ;	vect_p : matrice P des vecteurs propres de A mat_diag : matrice diagonale D obtenue vect_p et mat_diag ont la même taille que A
Bloc-diagonalisation d'une matrice carrée	[bloc_diag,P]=bdiag(A) ;	P : matrice de passage bloc_diag : matrice bloc-diagonale
Répartition en classes Lois quelconques Commande : dsearch(x,b,"c")	x =grand(1,100,"exp",3) ; b=[0,1,2,2.5,3,3.5,4,6,10] ; [pos,eff]=dsearch(x,b,"c") ;	Recherche parmi les éléments du vecteur x, ceux qui se trouvent dans une des classes définies par b. pos est un vecteur de même taille que x, qui indique le numéro de la classe à laquelle appartient chaque élément. eff donne l'effectif de chaque classe.
Répartition en classes Lois discètes Commande : dsearch(x,b,"d")	x =grand(1,100,"exp",3) ; v=[0,1,2,3,4,5,6,7,8,9,10] ; [pos,eff]=dsearch(x,b,"d") ;	Même fonction que ci-dessus, sauf que la recherche se fait par rapport aux valeurs entières définies dans le vecteur v

(*) Les matrices de Scilab sont stockées colonne par colonne, ce qui donne l'ordre dans lequel les éléments sont comptés

Probabilités et statistiques

Commandes	Scilab	Remarques
Moyenne, variance empirique, écart-type, médiane,maximum. Commandes : mean, variance, st_deviation, median, max.	lambda = 2 ; x =grand(1,100,"exp",1/lambda) ; m = mean(x) ; V_emp = variance(x) ; ecart_type = st_deviation(x) ; mediane = median(x) ; maximum = max(x) ;	Simulation d'une variable X suivant une loi exponentielle de paramètre 2 avec grand (x est une matrice ligne contenant 100 valeurs prises par X). Calcul de la moyenne, de la variance empirique, ($\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2$), de l'écart-type, de la médiane et du maximum
Cas d'une matrice quelconque : moyenne, écart-type, médiane, maximum, selon les lignes et les colonnes. Commandes : mean, st_deviation, median, max	x =grand(50,100,"nor",4,2) ; moy_l = mean(x, "r") ; moy_c = mean(x, "c") ; ecart_l = st_deviation(x, "r") ; ecart_c = st_deviation(x, "c") ; med_l = median(x, "r") ; med_c = median(x, "c") ; max_l = max(x, "r") ; max_c = max(x, "c") ;	Simulation d'une variable X suivant une loi normale d'espérance 4 et d'écart-type 2, x est une matrice contenant 50 lignes et 100 colonnes, dont les coefficients sont les valeurs prises par X. Calcul de la moyenne par colonnes (le résultat est une matrice ligne) et par lignes (le résultat est une matrice colonne). Calcul de l'écart-type, de la médiane, du maximum, par colonnes et par lignes.
Somme et somme cumulée commandes sum et cumsum	x =grand(50,100,"nor",4,2) ; somme_l = sum(x, "r") ; somme_c = sum(x, "c") ; s_cumul_l =cumsum(x, "r") ; s_cumul_c =cumsum(x, "c") ;	On reprend la matrice x précédente Calcul de la somme par colonnes et de la somme par lignes. Somme cumulée par colonnes : $z_{i,k}=\sum_{j=1}^{k}x_{i,j}$ pour $1\leq i\leq 50$ et $1\leq j\leq 100$, puis par lignes.
Fonction de répartition Commande : P=cdfnor("PQ",X,Mean,Std) ("cumulative distribution function - normal"), cdfbin (binomiale), cdfpoi (Poisson) ...	mu = 2 ; sigma = 5 ; x = 6 ; P = cdfnor('PQ',x,mu,sigma) ;	Calcule pour une loi normale de paramètres 2 et 25, la probabilité $P=\Phi(x)$ connaissant le seuil x
Quantiles, inversion des fonctions de répartition. Commande : X=cdfnor("X",Mean,Std,P,Q)	alpha = 0.05; x = cdfnor('X',0,1,1-alpha/2,alpha/2)	Calcul de la valeur de x tel que $\Phi(x)=1-\frac{\alpha}{2}$ Si Y suit la loi normale $N(0,1)$, $P(\|Y\|\leq x)=1-\alpha$.
Covariance empirique d'une série statistique à deux variables	x =grand(1,100,"nor",0,1) ; y =grand(1,100,"nor",0,1) ; covar = corr(x,y,1)	On peut vérifier que : variance(x) est égal à $\frac{n}{n-1}$*corr(x,x,1)
Fonction dsearch	(cf. aide de Scilab)	Répartition des données en classes
Simulation avec grand : lois binomiales, de Poisson, normales, uniformes à densité, uniformes discrètes, exponentielles, géométriques, gamma,	y1 = grand(m, n, "bin", N, p) ; y2 = grand(m, n, "poi", lambda) ; y3 = grand(m, n, "nor", mu, sigma) ; y4 = grand(m, n, "unf", a, b) ; y5 = grand(m, n, "uin", 1, n) ; y6 = grand(m, n, "exp", 1/lambda) ; y7 = grand(m, n, "geom", p) ; y8 = grand(m, n, "gam", b, nu) ;	Génère une matrice de taille m $\times$ n constituée de nombres aléatoires distribués selon la loi spécifiée. Remarques : pour la loi normale c'est l'écart-type qui est donné, pour la loi exponentielle c'est l'espérance.
Diagramme en bâtons et diagramme en barres Commandes plot2d3 et bar Commandes binomial et subplot	x = binomial(0.6, 20) ; subplot(1,2,1) ; plot2d3(0 :20, x) ; subplot(1,2,2) ; bar(0 :20, x) ;	La commande binomial permet d'obtenir les coefficients d'une loi binomiale, subplot permet de diviser la fenêtre graphique en deux sous-fenêtres
Tracés d'histogrammes Commandes : histplot(n,x) et histplot(b,x)	x =grand(1,100,"nor",1,2) ; subplot(121) ; histplot(10, x) ; subplot(122) ; b=[-9,-4,0,1,2,5,10] ; histplot(b,x) ;	Trace l'histogramme de x en utilisant n classes de même largeur, l'effectif de chaque classe est normalisé par l'effectif total. Trace l'histogramme de x en utilisant les classes dont les bornes sont définies par le vecteur b ($[b_1,b_2],[b_2,b_3]$, ...).

Tracés de figures avec plot2d

<table>
<tr><th>Commandes</th><th>Scilab</th><th>Remarques</th></tr>
<tr><td>Tracé du graphe de
$f : x \mapsto xe^{-x}$
Première méthode :
utilisation des
opérations coefficient
par coefficient
(.*, ./, .^, .+)
commandes : plot2d
et xtitle</td><td>x = [-1 :0.1 :5] ;
y = x .* exp(-x) ;
plot2d(x, y, style = 2) ;
xtitle("Graphe de f, avec f(x)=xexp(-x)") ;</td><td>x est une matrice ligne dont les coefficients
varient de -1 à 5 par pas de 0.1
y est une matrice ligne de même taille que x
Penser à utiliser .*, le paramètre 2 donne
la couleur de la courbe :<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>noir</td><td>bleu</td><td>vert</td><td>cyan</td><td>rouge</td><td>magenta</td><td>rouge</td></tr></table>xtitle permet de rajouter un titre</td></tr>
<tr><td>Tracé du graphe de
$f : x \mapsto xe^{-x}$
Deuxième méthode :
pour éviter les
opérations coefficient
par coefficient,
utilisation de deff
et de fplot2d</td><td>deff('y=f(x)', 'y= x*exp(-x)')
x = [-1 :0.1 :5]' ;
fplot2d(x, f, style = 2) ;
xtitle("Graphe de f, avec f(x)=xexp(-x)") ;</td><td>La commande deff permet de définir
rapidement une fonction simple (sinon on
peut utiliser function y=f() ... endfunction)
x est une matrice colonne
fplot2d permet d'éviter de définir la matrice
colonne y (sinon on peut aussi écrire :
y = feval(x,f) ; plot2d(x, y, 2) ;)</td></tr>
<tr><td>Tracé du graphe de
$f : x \mapsto xe^{-x}$
Troisième méthode :
Utilisation de la
commande : feval
et de la déclaration
générale d'une fonction</td><td>x = [-1 :0.1 :5] ;
function y=f(x)
y = x * exp(-x) ;
endfunction
y = feval(x, f) ;
plot2d(x, y, 3, rect=[0.7,0.8,4,3]) ;
xtitle("Graphe de f ") ;</td><td>On commence par définir la fonction f
x est ici une variable locale (x est un réel)
La commande feval permet de créer
la matrice ligne y</td></tr>
<tr><td>Tracé du graphe de
$f : x \mapsto \frac{x^2}{2x-1}$ en vert
et de l'asymptote
(en rouge)
dans la même fenêtre
avec un titre et une
légende</td><td>x = [1 :0.01 :4] ;
y1 = (x.^2) ./ (2*x-1) ;
y2 = 0.5 * x + 0.25 ;
plot2d(x, y1, style=3) ;
plot2d(x, y2, style=5, rect=[1,0,4,3]) ;
xtitle("Graphe de f et asymptote") ;
legend("Graphe de f", "asymptote") ;</td><td>Pour tracer plusieurs courbes sur un même
graphique, on peut soit définir x, $y1$ et $y2$
comme des matrices lignes comme ici,
soit procéder comme ci-dessous (*)
rect[$x_{\min}, y_{\min}, x_{\max}, y_{\max}$] permet de définir
la taille de la fenêtre graphique
On rajoute un titre et une légende</td></tr>
<tr><td>(*) Même exemple
que ci-dessus, mais
avec des matrices
colonnes, frameflag,
axesflag, xgrid</td><td>x = [1 :0.01 :4]' ;
y1 = (x.^2) ./ (2*x-1) ;
y2 = 0.5 * x + 0.25 ;
plot2d([x x],[y1 y2], [3 5], ..
frameflag=6, axesflag=5) ;
xgrid(1) ; xtitle('Variante') ;
legend("Graphe de f", "asymptote") ;</td><td>x est ici une matrice colonne de même que
y1 et y2, ce qui permet d'utiliser une seule
instruction plot2d, .. permet de passer à
la ligne dans une commande (lorsqu'elle
est trop longue à écrire sur une ligne),
frameflag=3 : échelle isométrique,
frameflag=6 : graduations simples,
axesflag=5 : axes qui se croisent au milieu
xgrid(1) : ajout d'une grille</td></tr>
<tr><td>Tracé de points
sans ligne brisée
entre les points
Paramètre
style = n</td><td>x = [-1 :0.2 :5] ;
y = x .* exp(-x) ;
plot2d(x, y, style = -1) ;
xtitle("Tracé des points") ;</td><td>style = -1 permet de remplacer
les points par des croix :<table><tr><td>0</td><td>-1</td><td>-2</td><td>-3</td><td>-4</td><td>-5</td></tr><tr><td>$\cdot$</td><td>$+$</td><td>$\times$</td><td>$\oplus$</td><td>$*$</td><td>$\diamond$</td></tr></table><table><tr><td>-6</td><td>-7</td><td>-9</td><td>-10</td><td>-11</td></tr><tr><td>$\triangle$</td><td>$\triangledown$</td><td>$\circ$</td><td>$*$</td><td>$\square$</td></tr></table></td></tr>
<tr><td>Tracés de figures
dans des fenêtres
différentes
commande :
xset('window', n)</td><td>x = [1 :0.01 :4] ;
y1 = x.^2 ; y2 = 3 * x + 1 ;
xset('window', 1) ; plot2d(x, y1, 3) ;
xset('window', 2) ; plot2d(x, y2, 5) ;</td><td>Trace la courbe représentative de $f : x \mapsto x^2$
dans la fenêtre numéro 1, et la courbe
représentative de $g : x \mapsto 3x + 1$
dans la fenêtre 2</td></tr>
<tr><td>Utilisation de
sous-fenêtres,
commande :
subplot(1,2,n)</td><td>x = [1 :0.01 :4] ;
y1 = x.^2 ; y2 = 3 * x + 1 ;
subplot(1,2,1) ; plot2d(x, y1, 3) ;
subplot(1,2,2) ; plot2d(x, y2, 5) ;</td><td>Création de deux sous-fenêtres sur une ligne,
tracés de la première courbe dans la première
sous-fenêtre et de la deuxième courbe dans
la deuxième sous-fenêtre</td></tr>
<tr><td>Commande : clf</td><td>clf() ;</td><td>Efface toutes les figures précédentes</td></tr>
</table>