Université Pierre et Marie Curie — Année 2013-2014
Licence — période 1

LM206 : Initiation à `Scilab`

3 Entrées, sorties, fichiers

Dans cette séance, l'accent est mis sur le pré-traitement des données (entrée des données par l'utilisateur) et le post-traitement des résultats (présentation des résultats) à travers un exemple modélisant l'évolution d'une population.

Modèle de Leslie On s'intéresse à la modélisation de l'évolution d'une population. Il s'agit d'une version simplifiée du modèle dit de Leslie. On note $u^{(n)}$ le vecteur de $\mathbb{R}^k$ représentant (en pourcentage) la population au temps n (calculé en années par exemple) qu'on a regroupé en k tranches d'âge :

- $u_1^{(n)}$ est la tranche formée des individus les plus jeunes,
- $\dots$
- $u_k^{(n)}$ est la tranche formée des individus les plus vieux.

On note f_i la taux de fécondité de la classe i et s_i le nombre d'individus de la classe i qui passent à la classe $i+1$ ($i = 1, \cdots, k-1$). Pour $k = 5$, l'évolution des classes de population entre les temps n et $n+1$ est

$$\left\{\begin{array}{lcl} u_1^{(n+1)} & = & f_1 u_1^{(n)} + f_2 u_2^{(n)} + f_3 u_3^{(n)} + f_4 u_4^{(n)} + f_5 u_5^{(n)} \\ u_2^{(n+1)} & = & s_1 u_1^{(n)} \\ u_3^{(n+1)} & = & s_2 u_2^{(n)} \\ u_4^{(n+1)} & = & s_3 u_3^{(n)} \\ u_5^{(n+1)} & = & s_4 u_4^{(n)}. \end{array}\right. \tag{1}$$

Commenter ces équations (et se convaincre de leur pertinence !).

Exercice 1 Écrire le système (1) sous forme matricielle $u^{(n+1)} = Au^{(n)}$ où A est une matrice de taille $k \times k$ appelée matrice de transistion à déterminer.
Solution.

3.1 Entrée de données

Exercice 2 Pour $k = 3$, $f_1 = 0.3$, $f_2 = 1$, $f_3 = 0.1$, $s_1 = .6$, $s_2 = .8$, calculer la matrice A. Partant des données initiales $u(0) = (100, 0, 0)^T$, calculer $u(20)$. Mêmes questions pour $s_1 = .9$, c'est-à-dire en supposant qu'il y a plus d'individus de la première tranche d'âge qui survivent et passent à la deuxième tranche. Commenter.
Solution.

Exercice 3 Utiliser la commande `input` pour lire les valeurs de k, f_i et s_i de l'exercice 2. Même question avec la commande `x_dialog` qui permet de lire ces valeurs en proposant à l'utilisateur des valeurs par défaut.
Solution.

3.2 Affichage

L'instruction la plus simple est `disp` qui permet d'afficher un texte. Exemple : `disp('Calcul de la solution')`.

L'instruction `mprintf` permet un affichage plus élaboré en utilisant des `format` d'affichage. Exemples

1. Le format `f` pour afficher des nombres réels : `15.5f` affiche un nombre réel sur 15 "cases" dont 5 sont reservés pour les chiffres après la virgule :

```
-->a=sqrt(2);
-->mprintf('a =  %10.4f  ',a)
a =      1.4142
-->mprintf('a = %10.6f  ',a)
a =   1.414214
-->mprintf('a = %10.9f  ',a)
a = 1.414213562
```

2. Le format `i` pour afficher des entiers

```
-->n=int(10000*a);
-->mprintf('partie entière de dix mille fois a = %10i ',n)
partie entière de dix mille fois a =      14142
```

3. Le format `s` pour afficher des chaînes de caractère

```
-->t='valeur trouvée pour a =';
-->mprintf('%s %10f',t,a);
valeur trouvée pour a =   9.869604
```

4. Utiliser `\n` pour indiquer un retour à la ligne

```
-->a=%pi*%pi;b=10;
-->mprintf('a = %10f et b = %10f', a,b)
a =   9.869604 et b =  10.000000
-->mprintf(' a = %10f \n b = %10f', a,b)
 a =   9.869604
 b =  10.000000
```

   La concaténation des chaînes de caractères `j'aime beaucoup` et `le calcul scientifique` donne :

```
-->t='j''aime beaucoup ';y='le calcul scientifique';
-->mprintf('La concaténation de %s et %s donne : \n %s',t,y,t+y);

j'aime beaucoup le calcul scientifique
```

Il existe d'autres formats, voir l'aide de `printf_conversion`.

Exercice 4 Soigner les sorties de l'exercice 3.
Solution.

3.3 Écriture/lecture dans un fichier

Il est essentiel de pouvoir écrire des résultats (importants !) dans un fichier pour pouvoir les conserver d'une session à l'autre. Les procédures de lecture et écriture de fichiers peuvent être très compliquées, surtout si on y ajoute des formats d'écriture/lecture. On se contentera ici du minimum : comment sauvegarder des variables dans un fichier (en format dit libre), et relire ces variables.

1. Ouverture.
   Utiliser la fonction `file` comme dans `fid=file('open','monfichier1')` qui crée un fichier nommé 'monfichier1' et lui associe l'entier `fid`. Attention : préciser le chemin complet pour accéder au fichier, si nécessaire. [1]

   Par la suite, on utilisera la variable `fid` pour effectuer des opérations sur ce fichier. Si le fichier existe déjà, écrire `fid=file('open','monfichier1','old')`. Il y a d'autres arguments d'entrée ou de sortie de la fonction `file` qu'on peut ignorer à ce stade de l'initiation à `Scilab`.
2. Fermeture.
   le fichier par `file('close',fid)`
3. Écriture.
   Se fait avec la fonction `write`. Par exemple

```
-->A=rand(5,3)
 A  =
    0.4829179    0.8607515    0.9923191
    0.2232865    0.8494102    0.0500420
    0.8400886    0.5257061    0.7485507
    0.1205996    0.9931210    0.4104059
    0.2855364    0.6488563    0.6084526
A=rand(5,3)
fid=file('open','monfichier1');
write(fid,size(A));
write(fid,A)
file('close',fid);
```

4. Lecture.
   Se fait avec la fonction `read`. Par exemple

```
fid=file('open','monfichier1','old');
dim=read(fid,1,2);
A=read(fid,dim(1),dim(2))
file('close',fid);
```

Exercice 5 Utiliser un éditeur de texte pour créer un fichier contenant les valeurs ci-dessous

```
3
.3 1 .1
.6 .8
100,0 0
```

1. par exemple C :/Utilisateurs/ . . .

qui correspondent respectivement au nombre k et aux vecteurs f, s et u_1 de l'exercice 5. Lire ce fichier et écrire le résultat des calculs dans un autre fichier.
Solution.

Remarque. La fonction `save` permet d'écrire des variables dans des fichier binaires qui ne peuvent être lus que pas `Scilab`. La lecture est effectuée par la fonction `load`. Exemple :

```
x=linspace(0,2*%pi,200);y=sin(x);
x0=x;y0=y;
save('SauvegardeSession','x','y');
clear x, clear y,
x
y
load('SauvegardeSession','x','y');
norm(x-x0), norm(y-y0)
plot(x,y)
```