Introduction à Scilab
Manipulations vectorielles

Jean-Philippe Chancelier & Michel De Lara
CERMICS, École des Ponts ParisTech

April 12, 2018

Contents

Ouvrir une fenêtre Scilab

Pour ces travaux pratiques d'introduction à Scilab, il vous faut lancer le logiciel Scilab et disposer ainsi d'une fenêtre permettant de saisir et d'exécuter des instructions.

Taper des instructions Scilab

Dans ces premiers travaux pratiques, vous trouverez une série de lignes de commandes Scilab précédées du signe `-->`. Pour commencer, il vous suffit de les recopier ou de les saisir par copier-coller (sans `-->`) pour les exécuter immédiatement dans la fenêtre Scilab.

Commentaires

Toute ligne débutant par // est une ligne de commentaires.

Pour récupérer le code des exemples, utiliser le lien : Code des exemples

1 Scalaires, vecteurs, matrices

- **scalaires**

```
-->5
// en tapant le chiffre 5, Scilab attribue la valeur 5 à la variable ans
// (pour answer)
-->ans^2
// ans élevé au carré donne 25
-->abs(-5)
// valeur absolue
-->m=10^6
-->sqrt(m)
// racine carrée
-->y=%e
-->log(y)
// %e est la constante e
-->sin(%pi)
// noter que le résultat n'est pas exactement zéro
-->1+%eps
// %eps est la précision : 1+%eps est indistinguable de 1
```

- **complexes**

```
-->%i
-->%i^2
-->x=3-4*%i
-->abs(x)
// module
```

- **vecteurs**

```
-->v=[3.8,-4,%pi/6]
-->size(v)
// dimensions de v : 1 ligne et 3 colonnes
-->w=v'
// transposition

-->x=[1,2]; y=[3,4,5]
-->z=[x,y];
```

```
// construction d'un vecteur par blocs

-->t=[4:9]
// vecteur des réels entre 4 et 9 par pas de 1 (pas implicite)
-->t=[4:1:9]
// vecteur des réels entre 4 et 9 par pas de 1 (pas explicite)
-->t=[0:0.1:1]
// vecteur des réels entre 0 et 1 par pas de 0.1

-->u=sqrt(2)*[1:2:8]'
-->size(u)
-->s=ones(u)
// un vecteur de même dimension que u et ne contenant que des 1
-->zeros(u)
// un vecteur de même dimension que u et ne contenant que des 0

-->t=rand(1,5)
// un vecteur à 1 ligne et 5 colonnes
// et contenant des nombres au hasard dans [0,1]
-->v=rand(s)
// un vecteur de même dimension que s
// et contenant des nombres au hasard dans [0,1]
```

- **matrices**

```
-->[1,2;3,4]
-->[11,12,13,14,15,16,17,18,19;21,22,23,24,25,...
-->26,27,28,29;31,32,33,34,35,36,37,38,39]
// dans une instruction trop longue pour tenir dans une ligne,
// mettre ... avant de passer à la ligne
-->diag([5 4 3])
// matrice diagonale
-->eye(6,6)
// des 1 sur la diagonale
-->B=eye(6,7)
-->A=ones(3,7)
-->C=[A;(-6)*B]
-->D=zeros(2,5)
-->E=rand(D)
-->rand(2,5)
```

Question 1

- ***Construire une matrice (4,9) (à 4 lignes et 9 colonnes) dont la première ligne est formée de 1, et dont tous les autres termes sont nuls.***
- ***Construire une matrice (3,5) dont la première colonne est formée de 2, la deuxième colonne des entiers de 1 à 3, et le reste de -1.***

2 Opérations vectorielles usuelles

- **fonctions (à arguments vectoriels)**

```
-->u=2*%pi*rand()
// un nombre au hasard dans [0,2*pi]
-->w=[cos(u) sin(u)]
-->norm(w)
-->norm(w,1)

-->t=[0:%pi/2:2*%pi]
-->v=sin(t)
-->[m,k]=maxi(v)
// la valeur maximale des éléments du vecteur v est m
// et elle est atteinte pour l'élément d'indice k : m=v(k)
-->[m,k]=mini(v)
-->sign(v)
// signe 1 (+) ou -1 (-) et sign(0)=0
```

- **opérations logiques**

```
-->1==0
// la réponse à l'assertion ``1 égale 0'' est F false
-->1~=0
// la réponse à l'assertion ``1 différent de 0'' est F false
// la réponse est T true
-->1==0 & 1~=0
// et : la réponse est F false
-->1==0 | 1~=0
// ou : la réponse est T true


-->t=[0:%pi/2:2*%pi]
```

```
-->v=sin(t)
-->v>0
// renvoie un vecteur de T (true) ou F (false) selon que
// l'élément correspondant de v est ou non >0
-->v>=0
-->bool2s(v>=0)
// convertit les T et F en 1 et 0
-->v(v>=0)
// extrait les élément positifs ou nuls de v
```

- **addition**

```
-->w=1:9
-->sum(w)
// somme de tous les éléments de w
-->cumsum(w)
// vecteur donnant les sommes cumulées
```

```
-->A=rand(2,3)
-->B=sin(A)
-->A+B
```

```
-->G=[ones(1,4); 2*ones(1,4)]
-->sum(G,'c')
// somme sur les lignes : le résultat est un vecteur colonne ('c' pour column)
-->sum(G,'r')
// somme sur les colonnes : le résultat est un vecteur  ligne ('r' pour row)
```

- **transposition**

```
-->A'
```

- **rang**

```
-->rank(A)
```

- **multiplication**

```
-->A'*A
-->A*A'
-->C=eye(A)
-->A'*C
```

- **déterminant**

```
-->A=[1,2;3,4]
-->det(A)
-->u=%pi/4 -100*%eps
-->det([cos(u) sin(u); sin(u) cos(u)])
```

- **exponentiation**

```
-->D=rand(3,3)
-->expm(D) // exponentielle de matrice I + D + D^2/2! + D^3/3! + ...
-->exp(D) // attention : exponentielle terme à terme
```

Question 2

- ***Choisir un angle θ au hasard dans $[0, 2\pi]$. Écrire la matrice carrée A correspondant à la rotation d'angle θ dans le plan. Quelle est la matrice inverse de A ?***

- ***Taper*** `help inv`. ***Calculer l'inverse de la matrice carrée A par la commande*** `inv(A)`.

- ***Vérifier le calcul avec*** `A*inv(A)`. ***Que constate-t-on ? Calculer*** `norm(A*inv(A)-eye(A))` ***et*** `clean(A*inv(A))`. ***Conclure !***

- **extraction d'éléments d'un vecteur**

```
-->w=1:2:9
-->w(2)
-->w($) // dernier élément
-->w($-1) // avant-dernier élément
```

- **extraction de sous-matrices**

```
-->E=[11:19;21:29;31:39;41:49;51:59;61:69]
-->E(1,1) // l'élément de la ligne 1 colonne 1
-->E(3,4) // l'élément de la ligne 3 colonne 4
-->E(1,:) // la ligne 1
-->E(:,5) // la colonne 5
-->E(2:4,:)
// la sous-matrice formée des lignes allant de 2 à 4
-->E(2:3,7:9)
// la sous-matrice formée des éléments appartenant
// aux lignes allant de 2 à 3 et aux colonnes de 7 à 9
-->E([1,3,5],[2,4,6,8])
// la sous-matrice formée des éléments appartenant
// aux lignes 1 3 5 et aux colonnes 2 4 6 8
-->E(:,$) // dernière colonne
-->E(:,$-1) // avant-dernière colonne
-->E(2:$,:) // les lignes de la deuxième à la dernière
-->E(2:($-1),:) // les lignes de la deuxième à l'avant-dernière
```

- **autres fonctions**

```
-->A=int(20*rand(1,10))
// partie entière
-->[sa,ia]=sort(A)
// tri : sa est le résultat du tri, ia les indices correspondants
```

Question 3

- ***Créer une matrice A au hasard avec 7 lignes et 10 colonnes.***
- ***Extraire de la matrice A la matrice B formée des lignes paires et des colonnes impaires de A.***

3 Opérations vectorielles terme à terme

- **multiplication terme à terme**

```
-->x=[1 2 3]
-->x.*x
```

```
-->y=[-6 12 8]
-->x.*y

-->A=rand(2,3);
-->B=ones(A);
-->A.*B
```

- **division terme à terme**

```
-->x=[1 2 3]
-->y=1 ./x
// un blanc suit le 1, car sinon 1. serait interprété en 1.0 et
// l'opération serait / (résolution de système linéaire)
-->x.*y

-->A=rand(2,3);
-->B=rand(A);
-->A./B
```

- **puissance terme à terme**

```
-->x=[1 2 3]
-->y=x.^2
-->z=x.^[5 10 -2]

-->A=rand(2,3);
-->A.^3
-->B=rand(A);
-->A.^B
```

Question 4 • ***Créer une matrice*** `A=rand(5,3)`***. Cette matrice a trois colonnes ; on souhaite multiplier la première colonne par 5, la deuxième par 2 et la troisième par 7. Effectuer cette opération en utilisant uniquement *,*** `.*`***,*** `ones` ***et le vecteur [5,2,7].***

4 Valeurs propres et vecteurs propres

- **valeurs propres**

```
-->A=eye(5,5)
-->spec(A)
// le spectre de la matrice carrée A, ensemble des valeurs propres
-->spec([1,2;3,4])
-->A=[1,2;-3,4]
-->spec(A)
```

- **vecteurs propres et matrice de passage**

```
-->[Ab,X]=bdiag(A)
// bdiag (bloc diagonalisation) donne deux arguments en retour :
// Ab matrice diagnonalisée
// X matrice de passage
-->X1= X(:,1); lambda1=Ab(1,1);
-->norm(A*X1 - lambda1*X1)
-->X2= X(:,2); lambda2=Ab(2,2);
-->norm(A*X2 - lambda2*X2)
-->inv(X)*A*X // vérification
```

Question 5 • ***Taper*** `apropos bdiag`***.***

- ***Diagonaliser une matrice*** `A=rand(4,4)` ***dans*** $\mathbb{R}$ ***puis dans*** $\mathbb{C}$***.***