Maths/Ordi, TP 4: résolution de systèmes linéaires

Dans cette séance, l'accent est mis sur les différentes possibilités de construction et de manipulation de vecteurs et de matrices avec Scilab puis sur la résolution de systèmes linéaires.

1 Construction de vecteurs

Il existe avec Scilab différentes possibilités de construire un vecteur (appelé `v` par la suite) :

- Lorsque la taille du vecteur est connue et est petite, on peut lister les composantes du vecteur ; par exemple `v=[1,2]` pour un vecteur ligne et `v=[1;2]` pour un vecteur colonne.
- Lorsque les valeurs du vecteur suivent une progression arithmétique, en écrivant par exemple `v=2:0.1:4`. Dans cet exemple, `2` est le premier terme de la suite et `0.1` sa raison (par défaut, la raison est égale à un). Le dernier paramètre (ici `4`) ne fait pas toujours partie de la suite, comme dans l'exemple `v=2:0.3:4`.
- En utilisant une boucle `for` :
  ```
  for k=1:4
     v(k)=k*k
  end;
  ```
  Cet exemple crée un vecteur `v` de dimension 4. Il est recommandé d'initialiser les vecteurs de grande dimension (avec l'instruction `zeros` ou `ones`, . . .).
- En initialisant `v` au vecteur vide : `v=[]` puis en utilisant une boucle `for` pour "concaténer" `v` et de nouveaux éléments
  ```
  for k=1:4 v=[v,k*k] end; ,
  ```
- En effectuant des opérations sur des vecteurs déjà définis.
  ```
  u=rand(3,1), v=rand(3,1), 2*u-3*v,
  ```
- En utilisant la fonction `linspace` (voir l'aide de cette commande).

On accède à un élément du vecteur `u` de dimension `n` par `u(k)` où `k` est un entier compris entre `1` et `n`.

Exercice 1 Tapez les instructions suivantes et en commenter les résultats.

```
n=5, u=rand(n,1), u(3), u(2:n-1), u($), u', length(u)
```

Exercice 2 Construire de plusieurs manières, le vecteur ligne de taille n qui comporte les carrés des n premiers nombres entiers.

2 Construction de matrices

Les vecteurs étant pour `Scilab` des cas particuliers de matrices de taille $n \times 1$ (pour un vecteur colonne) ou $1 \times n$ (pour un vecteur ligne), il est naturel que la construction d'une matrice A s'effectue de manière similaire à celle d'un vecteur, en l'occurrence :

- lorsque la taille de la matrice est connue et petite, en écrivant par exemple `A=[1,2,3;3,4,5]` pour une matrice de taille 2×3,
- en initialisant A à la matrice nulle (ou à la matrice identité avec `eye`) puis en effectuant une double boucle sur les indices avec des affectations du type `A(i,j)=2`,
- en initialisant `A` à un vecteur ligne (ou colonne) puis en concaténant dans une boucle chaque nouvelle ligne (ou colonne) avec des affectations du type `A=[A;v]` (respectivement `A=[A,v]`). A noter que cette méthode s'étend à la concaténation entre matrices,
- En effectuant des opérations de somme, de multiplication et de division matricielle.
  `u=rand(3,2), v=rand(3,2), 2*u-3*v,  u*v'`
- En effectuant des opérations composante par composante (utiliser les opérations `.*`, `./` ou `.^`) à partir de vecteurs déjà définis
  `u.*v, u./v, u.^v`

On accède à un élément de la matrice `u` de dimension `m`$\times$ `n`, c'est-à-dire à `m` lignes et `n` colonnes, par `u(i,j)` où `i` est une entier compris entre `1` et `m` et `j` est un entier compris entre `1` et `n`.

Exercice 3 Tapez les instructions suivantes et en commenter les résultats.

```
m=5;n=4;
for i=1:m
   for j=1:n
      A(i,j)=i-j;
   end;
end;
A, size(A),  length(A),
u=A(3,:), size(u)
v=A(:), size(v), length(v)
```

Exercice 4 Construire la matrice de taille 9×9 dont tous les éléments sont nuls sauf les éléments du "bord" $i \in \{1, 9\}$ ou $j \in \{1, 9\}$, et les éléments du "centre" $(i, j) \in \{4, 5, 6\} \times \{4, 5, 6\}$ qui valent 1.

3 Extraction de sous-matrices

Il est possible d'extraire facilement une sous-matrice d'une matrice quelconque simplement en construisant le vecteur formé par les indices de lignes et celui formé par les indices de colonnes à sélectionner. Par exemple, l'instruction `B=A(1:2:5,1:3)` extrait de la matrice A, l'intersection des lignes 1, 3 et 5 et des colonnes 1, 2 et 3 pour former une matrice de taille 3×3. Ainsi une solution de l'exercice 4 est

```
A=zeros(9,9); A(4:6,4:6)=1;A([1,9],:)=1;A(:,[1,9])=1;
```

4 Résolution de systèmes linéaires

La résolution de systèmes linéaires du type $Ax = b$ avec Scilab peut être réalisée à l'aide d'instructions préprogrammées comme \ ou `linsolve` (voir l'aide de ces deux instructions pour plus de détails) mais elle peut aussi se faire directement en reprogrammant une des méthodes existantes (Gauss, LU, Choleski, Jacobi, etc...).

Avant d'aborder la résolution de systèmes linéaires, il est essentiel de s'assurer qua la matrice utilisée est bien conditionnée comme le montre l'exercice suivant.

Exercice 5 Construire en dimension n quelconque la matrice de Hilbert H_n de terme général $\frac{1}{i+j+1}$ et résoudre un système linéaire avec la matriceH_{15} et l'instruction \ de Scilab. Le résultat obtenu est-il satisfaisant ? Calculer le conditionnement de la matrice H_{15} et conclure.

Exercice 6 En profitant de la structure matricielle vue dans les exercices précédents, programmer la méthode de résolution du pivôt de Gauss (avec ou sans stratégie de pivôt).

Vérifier la précision et la rapidité de cette méthode sur un cas aléatoire en grande dimension en prenant par exemple `A=rand(n,n);b=A*ones(n,1)`.