

Chapitre 3

Manipulations des matrices et des vecteurs

3.1 Comment générer des matrices et des vecteurs

On peut facilement générer une matrice nulle de taille quelconque avec la commande `zeros` qui s'utilise comme ceci :

```
--> A=zeros(3,2)
```

A =

```
0.  0.  
0.  0.  
0.  0.
```

Cette commande peut par exemple servir pour créer et initialiser une matrice.

On peut générer la matrice identité avec la commande `eye` de la façon suivante :

```
--> I=eye(3,3)
```

I =

```
1.  0.  0.  
0.  1.  0.  
0.  0.  1.
```

On peut aussi générer des vecteurs particuliers à l'aide du caractère `:` (deux points), comme le montre l'exemple suivant :

```
--> x=1:5
```

```
x =
```

```
1.    2.    3.    4.    5.
```

Cette commande a permis d'affecter au vecteur x les entiers de 1 à 5. On peut aussi spécifier un incrément particulier autre que 1 de la façon suivante :

```
--> y=0:%pi/4:%pi
```

```
y =
```

```
0.    0.7853982    1.5707963    2.3561945    3.1415927
```

On notera que cette commande produit toujours un vecteur ligne. Il est bien sûr possible d'utiliser un incrément négatif :

```
--> y=6:-1:1
```

```
y =
```

```
6.    5.    4.    3.    2.    1.
```

On peut aisément créer des tables en utilisant cette commande (on verra au chapitre suivant que cela permet de créer rapidement des représentations graphiques). Par exemple :

```
--> x=(0:0.2:3)';
--> y=exp(-x).*sin(x);
--> [x y]
```

```
ans =
```

```
0.    0.
0.2    0.1626567
0.4    0.2610349
0.6    0.3098824
0.8    0.3223289
1.    0.3095599
1.2    0.2807248
1.4    0.2430089
1.6    0.2018104
```

```

1.8    0.1609759
2.     0.1230600
2.2    0.0895840
2.4    0.0612766
2.6    0.0382881
2.8    0.0203707
3.     0.0070260

```

Il existe une commande permettant de spécifier uniquement les valeurs minimum et maximum et le nombre de valeurs désirés :

```
--> k=linspace(-%pi,%pi,5)
```

```
k =
```

```
- 3.1415927 - 1.5707963  0.  1.5707963  3.1415927
```

3.2 Comment manipuler les éléments d'une matrice

On peut faire référence aux éléments individuels d'une matrice en précisant leurs numéros de ligne et de colonne dans des parenthèses suivant le nom de la matrice. Prenons la matrice

```
--> A=[1  2  3;  4  5  6;  7  8  9]
```

La commande suivante permet de remplacer a_{33} par $a_{13} + a_{31}$:

```
--> A(3,3)=A(1,3) + A(3,1)
```

```
A =
```

```

1.    2.    3.
4.    5.    6.
7.    8.   10.

```

On peut aussi facilement extraire une ligne ou une colonne d'une matrice ; par exemple, la commande

```
--> v=A(:,1)
```

```
v =
```

```

1.
4.
7.

```

permet de mettre dans le vecteur v la première colonne de la matrice A . De la même manière, la commande

```
--> v=A(1,:)
```

permettra d'extraire la première ligne de A .

Un indice de ligne ou de colonne peut être un vecteur. Par exemple si nous prenons le vecteur

```
--> x=0:2:8;
```

la commande suivante permet d'effectuer une permutation des éléments de x :

```
--> v=[3 5 1 2 4];
```

```
--> x(v)
```

```
ans =
```

```
4.    8.    0.    2.    6.
```

Il est possible de construire une matrice à partir de matrices plus petites. Par exemple, pour ajouter une nouvelle ligne à la matrice A :

```
--> l=[10 11 12];
```

```
--> A=[A; l]
```

Cela donne

```
A =
```

```
1.    2.    3.
4.    5.    6.
7.    8.   10.
10.   11.   12.
```

On peut aussi facilement manipuler les sous-matrices d'une matrice, on l'a déjà vu plus haut pour le cas d'une ligne ou d'une colonne :

```
--> A(1:2,1:2)=eye(2,2)
```

```
A =
```

```
1.    0.    3.
0.    1.    6.
7.    8.   10.
10.   11.   12.
```

Ici on a remplacé la sous-matrice principale d'ordre 2 de A par la matrice identité. On peut extraire une sous-matrice d'une matrice plus grande. Par exemple

```
--> B=A(1:2, :);
```

sélectionne les deux premières lignes et toutes les colonnes de A et les stocke dans une matrice B.

Pour en finir avec les matrices, on peut obtenir automatiquement leur taille (nombre de lignes et de colonnes) avec la fonction `size` :

```
--> size(A)
```

```
ans =
```

```
4.    3.
```

```
--> [n,m]=size(v)
```

```
m =
```

```
5.
```

```
n =
```

```
1.
```

La première de ces deux commandes renvoie un vecteur à deux composantes : la première est le nombre de lignes et la deuxième est le nombre de colonnes. La deuxième commande permet d'assigner ces deux valeurs à deux variables différentes, `n` pour les lignes et `m` pour les colonnes.