

## TP6 : Matrices – Suite

### Définition de matrices

On a vu au TP précédent comment définir une matrice donc les coefficients sont donnés. Par exemple, pour définir  $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ , on entre **A=[1,2;3,4]**.

Ceci est rapidement inconfortable lorsque la taille de la matrice grandit. On ne peut pas envisager, par cette méthode, définir une matrice de taille 100, chose qui est pourtant indispensable.

Scilab dispose heureusement de fonctions qui définissent des matrices de taille donnée. Il en existe trois :

- **zeros(n,p)** qui définit la matrice nulle de  $\mathcal{M}_{n,p}(\mathbb{R})$
- **ones(n,p)** qui définit la matrice ne contenant que des 1 de  $\mathcal{M}_{n,p}(\mathbb{R})$
- **eye(n,n)** qui définit la matrice identité de  $\mathcal{M}_n(\mathbb{R})$ .

Pour définir une matrice sur Scilab, on utilisera donc l'une des trois fonctions précédentes et on utilisera des boucles **for** pour donner aux coefficients les valeurs désirées.

**Q.1** Définir dans Scilab la matrice **A** de taille 10x10 contenant des 1 sur la sur-diagonale et des 0 ailleurs :

$$\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}$$

Calculer  $A, A^2, A^3, \dots, A^{10}$ . Que remarque-t-on ?

**Q.2** La fonction **size** renvoie le vecteur-ligne formé du nombre de ligne et du nombre de colonnes. Pour l'utiliser, on écrira **[n,p]=size(A)** et **n** sera le nombre de lignes de **A** et **p** son nombre de colonnes. Tester avec la matrice précédente.

### Résolution de systèmes linéaires

Soient  $A = (a_{ij})_{1 \leq i, j \leq n} \in \mathcal{M}_n(\mathbb{R})$  une matrice **inversible** et  $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathcal{M}_{n,1}(\mathbb{R})$ . On rappelle

que le système

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + \dots + a_{nn}x_n = b_n \end{cases} \iff AX = b$$

d'inconnue  $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathcal{M}_{n,1}(\mathbb{R})$  a pour unique solution  $X = A^{-1}b$ .

**Q.3** Soit  $A = \begin{pmatrix} 2 & 1 \\ -3 & 2 \end{pmatrix}$ . Montrer que  $A$  est inversible et calculer son inverse. Résoudre  $AX = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ . Vérifier à l'aide de Scilab de deux manières :

1. à l'aide de l'instruction  $\mathbf{A} \wedge (-1) * \mathbf{b}$  ;
2. à l'aide de l'instruction  $\mathbf{A} \setminus \mathbf{b}$ .

**Q.4** Définir la matrice  $\mathbf{A}$  de taille 1000x1000 contenant des 1, sauf sur la diagonale où les coefficients sont 0. Définir le vecteur colonne  $\mathbf{b}$  de taille 1000 ne contenant que des 1. On s'intéresse à la méthode la plus efficace pour résoudre  $AX = b$ . Exécuter les instructions suivantes :

```
timer()
X=A^(-1)*b;
timer()
Y=A\b;
timer()
```

Quelle est la méthode de résolution la plus rapide ?

L'objectif de la fin du TP est de programmer l'algorithme du pivot de Gauss pour la résolution de système linéaire.

**Q.5** Résolution d'un système triangulaire. Écrire une fonction d'en-tête

**function x=SolTriangulaireSup(A,b)**

qui prend en paramètre une matrice triangulaire supérieure  $\mathbf{A}$  et un vecteur colonne  $\mathbf{b}$  et renvoie le vecteur colonne  $\mathbf{x}$ .

L'idée du pivot de Gauss pour la résolution de  $Ax = b$  est de transformer la matrice  $A \in \mathcal{M}_n(\mathbb{R})$  en une matrice triangulaire supérieure puis d'appliquer la fonction précédente pour la résolution.

On considère en fait une matrice  $M \in \mathcal{M}_{n,p}(\mathbb{R})$  à priori rectangulaire. L'idée est la suivante :

– On copie la matrice  $M$  dans la variable  $\mathbf{B}$

– Pour  $k$  allant de 1 jusqu'à  $n - 1$  :

\* on détermine la ligne  $i_0$  telle que  $|b_{i_0,k}| = \max_{k \leq i \leq n} |b_{i,k}|$

\* on échange les lignes  $k$  et  $i_0$

\* on élimine les coefficients  $b_{i,k}$  pour  $i > k$  par opérations élémentaires sur les lignes : pour  $i$  allant de  $k + 1$  à  $n$ ,

- la ligne  $i$   $L_i$  est remplacée par  $L_i - \frac{b_{i,k}}{b_{k,k}} L_k$ .

**Q.6** Écrire une fonction d'en-tête

**function numLigne = pivot(B,k)**

qui prend en paramètres une matrice  $B \in \mathcal{M}_{n,p}(\mathbb{R})$  et un entier  $k \in \llbracket 1, n \rrbracket$  et qui renvoie l'indice  $i_0$  de la ligne telle que  $|b_{i_0,k}| = \max_{k \leq i \leq n} |b_{i,k}|$ .

On utilisera la fonction **size** pour connaître  $n$ , puis on parcourra la colonne  $k$  de la ligne  $k$  à la ligne

$n$  en gardant en mémoire l'indice de la ligne qui est, pour l'instant, de valeur absolue maximale.

Q.7 Écrire une fonction d'en-tête

**function B = Echange(A,i,k)**

qui prend en paramètres une matrice  $A \in \mathcal{M}_{n,p}(\mathbb{R})$  et des entiers  $i, k$  et qui renvoie la matrice  $B$ , copie de  $A$  sur laquelle on aura échangé les lignes  $i$  et  $k$ .

Q.8 Écrire une fonction d'en-tête

**function B = Elimine(A,k)**

qui prend en paramètres une matrice  $A \in \mathcal{M}_{n,p}(\mathbb{R})$  et un entier  $k$  et qui renvoie la matrice  $B$ , copie de  $A$  sur laquelle on aura éliminé les coefficients  $b_{i,k}$  selon la méthode décrite ci-dessus.

Q.9 Écrire une fonction d'en-tête

**function B = Gauss(A)**

qui, à l'aide des fonctions précédentes, applique l'algorithme décrit ci-dessus et renvoie la matrice  $B$  obtenue de  $A$  par la méthode du pivot de Gauss.

*On pourra tester sa fonction avec des matrices carrées inversibles. On doit obtenir des matrices triangulaires supérieures.*

Q.10 Écrire une fonction d'en-tête

**function x = Solution(A,b)**

qui à une matrice  $A \in \mathcal{M}_n(\mathbb{R})$  inversible et un vecteur colonne de taille  $n$  associe la solution  $x$  de l'équation  $Ax = b$ .

*On utilisera la fonction précédente sur la matrice  $C = (A, b) \in \mathcal{M}_{n,n+1}(\mathbb{R})$  dont les colonnes sont celles de  $A$  suivies de la colonne  $b$ . On obtiendra une matrice de taille  $n \times (n+1)$  dont les  $n$  premières lignes et colonnes forment une matrice de taille  $n \times n$  qui est triangulaire supérieure. On copiera les coefficients d'indices  $(i, j)$  avec  $1 \leq i, j \leq n$  dans une matrice  $T$  et les coefficients  $(i, n+1)$  dans un vecteur colonne  $V$ . On utilisera enfin la fonction **SolTriangulaireSup** pour conclure.*