TP4 : Matrices

1. Définition élément par élément

Soient $(n,p) \in \mathbb{N}^* \times \mathbb{N}^*$ et $(a_{i,j})_{\substack{1\leqslant i\leqslant n\\ 1\leqslant j\leqslant p}}$, $n \times p$ nombres réels ou complexes. Pour définir la matrice $A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,p} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,p} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,p} \end{pmatrix} \in \mathcal{M}_{n,p}(\mathbb{K})$ on crée un vecteur en séparant les lignes par un point-virgule.

Par exemple, pour affecter à la variable `M` la matrice $\begin{pmatrix} 1 & 0 & i \\ \pi & -\sqrt{2} & 42 \end{pmatrix}$ on écrit :

```
-->M=[1, 0, %i; %pi,-sqrt(2), 42]
```

Exercice 1 :

(a) Définir les matrices suivantes dans la console *Scilab*.

$$A = \begin{pmatrix} 1 & 4 \\ -2 & 5 \\ 3 & 0 \end{pmatrix} \quad ; \quad B = \begin{pmatrix} 1/2 & 1 & 3/2 \\ 1/3 & 2/3 & 1 \\ 1/4 & 1/2 & 3/4 \end{pmatrix} \quad ; \quad C = \begin{pmatrix} i & 0 & 1 \\ 0 & 1+i & 0 \\ 1 & 0 & i \end{pmatrix}$$

(b) Donner en argument aux fonctions `length` et `size`, chaque matrice ainsi définie. (Ces deux commandes sont à connaître !)

(c) Conclure quant à la différence entre ces deux fonctions ?

2. Accès et modification des coefficients

La commande `A(i,j)` permet d'accéder au coefficient situé à la i$^{\text{ème}}$ ligne et j$^{\text{ème}}$ colonne et éventuellement de le modifier.

La commande `A(i,:)` permet d'extraire ou modifier la i$^{\text{ème}}$ ligne et `A(:,j)` d'extraire ou modifier la j$^{\text{ème}}$ colonne.

Exemple (à tester et comprendre à partir de la matrice `C` précédente) :

```
-->C(2,2)=%pi
-->C(:,3)=(1:3)'
```

3. Définition globale

(a) zeros

`zeros(n,p)` permet de créer la matrice nulle de taille `n`×`p`. Exemple :

```
-->N=zeros(2,3)
N =
   0. 0. 0.
   0. 0. 0.
```

(b) ones

`ones(n,p)` permet de créer la matrice de taille `n`×`p` dont tous les coefficients sont égaux à 1. Exemple :

```
-->U=ones(2,2)
U =
   1. 1.
   1. 1.
```

(c) eye

`eye(n,p)` permet de créer la matrice $A = (a_{i,j})$ de taille `n`×`p` dont tous les coefficients $a_{i,i}$ sont égaux à 1 et les autres sont nuls. Exemple :

```
-->I=eye(3,4)
I =
   1. 0. 0. 0.
   0. 1. 0. 0.
   0. 0. 1. 0.
```

Remarque : dans le cas où `n=p`, `eye(n,n)` est alors la matrice identité d'ordre `n`.

(d) rand

`rand(n,p)` permet de créer une matrice de taille `n`×`p` dont tous les coefficients sont aléatoires, indépendants et suivent la loi uniforme sur $[0, 1]$. Exemple :

```
-->A=floor(10*rand(3,2))
A =
   2. 6.
   0. 7.
   8. 0.
```

4. Opérations sur les matrices

(a) Opérations élémentaires

Tester et comprendre la suite d'instructions suivantes :

```
-->A=eye(4,4);
-->B=2*ones(4,4);
-->C=B.*B-4*A
```

Attention : `B.*B` n'a pas du tout la même signification que `B*B` (testez-le). La première commande désigne le produit coefficient par coefficient tandis que la seconde (plus utile) désigne le produit matricielle.

(b) Transposition

Si `A` est une matrice alors `A'` désigne la matrice transposée de `A`. Exemples à tester et comprendre :

```
-->M=floor(10*rand(3,3));
-->S=(M+M')/2;
-->A=(M-M')/2;
-->A+S
-->M
```

Que donne la suite d'instructions suivantes :

```
-->M=floor(10*rand(3,3));
-->(M*M')'-M*M'
```

Comment l'expliquez-vous mathématiquement ?

(c) **Concaténation**
Si `A` $\in \mathcal{M}_{n,p}(\mathbb{C})$ et `B` $\in \mathcal{M}_{n,q}(\mathbb{C})$ alors `[A,B]` $\in \mathcal{M}_{n,p+q}(\mathbb{C})$ est la matrice obtenue en "collant" la matrice `B` à droite de la matrice `A`.

Si `A` $\in \mathcal{M}_{n,p}(\mathbb{C})$ et `B` $\in \mathcal{M}_{m,p}(\mathbb{C})$ alors `[A;B]` $\in \mathcal{M}_{m+n,p}(\mathbb{C})$ est la matrice obtenue en "collant" la matrice `B` sous la matrice `A`.
Exemple à tester et comprendre :

```
-->A=floor(10*rand(3,3))
-->B=floor(10*rand(3,3))
-->C=[A(:,2),B(:,3)]
-->D=[B(1:2,:);A(2:3,:)]
```

(d) **diag**
Si `A`$= (a_{i,j}) \in \mathcal{M}_{n,p}(\mathbb{C})$ alors la commande `diag(A)` renvoie le vecteur contenant les coefficients diagonaux : $[a_{11}, a_{22}, ...]$.
`diag([x1,...xn])` renvoie la matrice diagonale correspondante.
Exemple à tester et comprendre :

```
-->A=floor(10*rand(4,6))
-->diag(A)
-->diag(A,1), diag(A,-1)
-->diag([2,4,6,8])
```

(e) **inv**
Si `A`$= (a_{i,j}) \in \mathcal{M}_{n}(\mathbb{C})$ est une matrice carrée inversible alors la commande `inv(A)` renvoie la matrice carrée $\texttt{A}^{-1}$ inverse de `A`.
Exemple :

```
-->A=[0 1 1; 1 0 1; 0 1 0]
-->B=inv(A)
-->A*B
```

Remarque : On peut aussi écrire `B/A`, ce qui revient à `B*inv(A)`.

(f) **find**
L'instruction `find(test A)` renvoie, sous forme de vecteur, la liste des indices de `A` vérifiant le `test`. Exemple à tester :

```
-->A=floor(10*rand(3,3))
-->find(A>5)
```

5. Exercices

Exercice 1. On considère la matrice $A = \begin{pmatrix} -1 & 0 & 3 & 5 \\ 0 & 4 & -2 & 0 \\ 3 & 0 & 1 & 0 \end{pmatrix}$.

1. Affecter cette matrice à une variable `A`.
2. Quelle commande donne la taille de la matrice `A` ?
3. Extraire la première colonne de `A`, puis la seconde ligne et enfin la *diagonale* de `A`.
4. À l'aide d'une seule commande, extraire la matrice composée des trois dernières lignes et des trois dernières colonnes de `A` et l'affecter à une matrice `B`. Calculer l'inverse de `B`.

Exercice 2. Inverse d'une matrice par polynôme annulateur

On pose $A = \begin{pmatrix} 1 & 3 & -2 \\ 0 & 4 & -3 \\ -2 & 1 & 0 \end{pmatrix}$

1. Calculer $A^3 - 5A^2 + 3A$ à l'aide de *Scilab* et exprimer le résultat à l'aide de I_3.
2. En déduire une expression de l'inverse de A en fonction de A^2, A et I_3.
3. Vérifier à l'aide de la console que la matrice ainsi trouvée est bien l'inverse de A.

Exercice 3. Résolution d'un système linéaire

On veut résoudre le système linéaire suivant :

$$(S) \begin{cases} 3x & -2y & & +t & = & 3 \\ -5x & +3y & -2z & +2t & = & 3 \\ x & -y & +z & -t & = & -2 \\ 4x & -10y & +7z & -4t & = & -11 \end{cases}$$

1. Montrer que le système (S) est équivalent au produit matriciel $AX = B$ avec $A \in \mathcal{M}_4(\mathbb{R})$ et $X, B \in \mathcal{M}_{4,1}(\mathbb{R})$ que l'on explicitera.
2. Exprimer alors le quadruplet inconnu X en fonction de A et B.
3. Résoudre à l'aide de *Scilab* le système (S) et donner une solution. Y en a-t-il d'autres ?

Exercice 4. Matrice de rotation dans un *r.o.n.d.*

1. Écrire un script *Scilab* qui demande à l'utilisateur un nombre θ et crée la matrice $M(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta \end{pmatrix}$.
2. Ajouter quelques lignes de code pour le programme demande aussi à l'utilisateur un vecteur colonne $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.
3. Modifier encore le programme pour qu'il calcule ensuite le vecteur colonne $Y = M(\theta)X$.

Exercice 5. On souhaite, uniquement à l'aide des commandes `diag` et `ones`, créer la matrice carrée d'ordre 8 suivante :

$$A = \begin{pmatrix} -1 & 2 & 0 & \cdots & 0 \\ 3 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 2 \\ 0 & \cdots & 0 & 3 & -1 \end{pmatrix}$$

1. Écrire et comprendre les instructions suivantes :

```
-->2*ones(7,1)
-->diag(2*ones(7,1))
-->diag(2*ones(7,1),1)
```

2. En déduire une suite de commandes permettant de créer la matrice A.
3. La matrice A est-elle inversible ?
4. Écrire un script qui demande à l'utilisateur quatre nombres a, b, c et n et qui créer ensuite la matrice carrée d'ordre n suivante :

$$\begin{pmatrix} a & b & 0 & \cdots & 0 \\ c & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b \\ 0 & \cdots & 0 & c & a \end{pmatrix}$$

Exercice 6. Interpolation polynomiale et matrice de Vandermonde

On se donne $(n+1)$ couples de nombres $(x_i, y_i) \in \mathbb{R}^2$, avec $i \in [\![0, n]\!]$ et on cherche un polynôme $P \in \mathbb{R}_n[X]$ passant par tous les points (x_i, y_i). Autrement dit, on recherche $(n+1)$ coefficients $(a_0, \ldots, a_n) \in \mathbb{R}^{n+1}$ tels que le polynôme $P(X) = \sum_{k=0}^{n} a_k X^k$ vérifie pour tout $i \in [\![0, n]\!]$, $P(x_i) = y_i$.

1. Expliquer pourquoi cela revient à résoudre le système matricielle d'inconnues $(a_0, \ldots, a_n)$ suivant :

$$\begin{pmatrix} 1 & x_0^1 & \cdots & x_0^n \\ 1 & x_1^1 & \cdots & x_1^n \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_n^1 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

2. Plus particulièrement, on recherche un polynôme

$$P = a_0 + a_1 X + a_2 X^2 + a_3 X^3 \in \mathbb{R}_3[X]$$

passant par les points $E : (-2, -2)$, $F : (-1, 2)$, $G : (0, -4)$, $H : (1, 4)$. Poser le problème sous forme matricielle, *i.e.* déterminer une matrice X de taille 4×4 et un vecteur colonne Y de taille 4×1 tels que les coefficiens recherchés (a_0, a_1, a_2, a_3) forme un vecteur colonne A solution du système $XA = Y$.

3. Créer la matrice X et le vecteur colonne Y dans *Scilab* et résoudre le problème (on donnera l'écriture explicite du polynôme P passant par les points E, F, G et H).
4. Tracer le graphe de P sur $[-2, 2]$ (*voir TP2*) et vérifier que la courbe tracée passe bien par les points E, F, G et H. Pensez-vous que P admette une racine complexe ?

Exercice 7. Script pour la matrice de Vandermonde

Créer un script qui demande à l'utilisateur un vecteur colonne $[x_0; \ldots; x_n]$ de taille quelconque et qui crée la matrice de Vandermonde associée.

Indication : on pourra utiliser la fonction `length()` *et une boucle for pour concaténer les 'puissances' du vecteur.*

Exercice 8. Étude de la trajectoire d'un drone à l'aide d'un système récurrent

Un drone sans pilote se déplace de manière automatique selon la méthode suivante :

Sa position est donnée à un instant $n \in \mathbb{N}$ par ses coordonnées $\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}$ dans un repère orthonormal. On pose $X = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \in \mathbb{R}^3$ et pour tout $n \in \mathbb{N}$, la position du drone à l'instant $n+1$ dépend de sa position à l'instant n de la manière suivante :

$$\begin{cases} x_{n+1} = & 2x_n & -3y_n & +1,5z_n \\ y_{n+1} = & -0,4x_n & +1,6y_n & -0,3z_n \\ z_{n+1} = & -0,8x_n & +2,2y_n & -0,1z_n \end{cases}$$

1. Déterminer une matrice A telle que AX donne les coordonnées du drone à l'instant $n = 1$, puis affecter cette matrice à une variable `A` dans la console.
2. On cherche à déterminer la position du drone à un instant $n \in \mathbb{N}$ quelconque en fonction du point de départ. Déterminer la position du drone aux instants $n = 5, 10, 50, 70$ et 80 lorsque $X = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$. Le drone semble-t-il se stabiliser ?
3. On donne $P = \begin{pmatrix} 2 & -1 & 3 \\ 0 & 2 & -1 \\ -1 & 2 & -1 \end{pmatrix}$. Démontrer mathématiquement que P est inversible et déterminer son inverse. Vérifier ensuite vos calculs avec *Scilab*.
4. Calculer $D = PAP^{-1}$ puis vérifier à l'aide de *Scilab*.
5. En déduire une expression simple de A^n en fonction de n, D et P. Vérifier avec *Scilab*.
6. On cherche à savoir si ${}^t(0,0,0)$ est la seule position stable pour le drone. Extraire le premier vecteur colonne de P^{-1} et l'affecter à X, puis calculer la position du drone à divers instant $n \in \mathbb{N}$. Que remarquez-vous ?
7. Faire de même avec le deuxième et le troisième vecteur colonne de P^{-1}. Où semble se diriger le drone ?