

# Remise à niveau Mathématiques

## TD sur ordinateur

### Algèbre linéaire

Avant de commencer, quelques sites pouvant compléter votre formation d'initiation à Matlab :

Le site officiel de Matlab :

<http://www.mathworks.fr/>

<http://www.math.iitb.ac.in/~pde08/matlab1.pdf>

M-file database : <http://matlabdb.mathematik.uni-stuttgart.de/index.jsp>

***NB : Le but n'est pas d'améliorer sa "fréquence de frappe" sur clavier mais de comprendre ce que l'on fait.***

### **Consignes pour le compte-rendu**

Vous rédigerez un compte-rendu concernant ce TP. Il ne doit pas être la copie des programmes, plutôt un détail de ce qui aura été écrit *au tableau* par l'enseignant. Il devra contenir la réponse aux questions d'ordre général présentes dans le TP.

### **Utilisation de MATLAB**

Pour pouvoir utiliser correctement **MATLAB** et envisager de sauvegarder des programmes, créez un dossier à votre nom (Nom\_1\_Nom\_2\_CAO\_Matlab) et choisissez-le comme dossier de travail pour Matlab, en choisissant le chemin de ce dossier dans la barre d'adresse Matlab.

**MATLAB** est un logiciel qui travaille essentiellement sur des matrices et des vecteurs. Les matrices sont des tableaux de nombres de n lignes et p colonnes. Un vecteur est une matrice à une ligne ou à une colonne. Pour générer une matrice, on peut expliciter la liste de ses éléments. La liste est entourée par des crochets. Un point virgule, ; , sépare les rangées (lignes) et une virgule ou un espace sépare les éléments d'une même rangée (colonnes). Pour entrer une matrice 3\*3 tapez par exemple :

```
>> A=[ 1 2 3 ; 4 5 6 ; 7 8 0 ]
```

Un nombre est évidemment une matrice à une ligne et une colonne (vous n'avez pas dans ce cas à mettre des crochets).

## 1. Variables, données et sauvegarde

### 1.1 Premières commandes de calcul

\* Calcul de la transposée de la variable matricielle A (inversion des lignes et des colonnes)

```
>>B=A'
```

\* Inverse d'une matrice

```
>>C=inv(A)
```

\* Produit de matrices

```
>>D=A*C (Un produit matriciel ne peut s'effectuer que si le nombre de colonnes de A est identique au nombre de lignes de C)
```

Toutes les variables utilisées ci dessus sont gardées en mémoire vive et il est possible de visualiser à tout instant l'espace de travail (l'ensemble des variables utilisées). Pour cela tapez :

```
>>who
```

Taper après

```
>>whos
```

Pour visualiser le contenu de la variable B tapez

```
>>B
```

L'espace de travail peut être sauvegardé dans un fichier de données toto par l'instruction

```
>>save toto (création d'un fichier de données toto.mat)
```

On peut effacer l'espace de travail pour libérer de la mémoire en tapant

```
>>clear
```

Vérifier que l'espace de travail est vide (souvenez vous de la commande who)

Pour récupérer l'espace de travail sauvegardé dans la donnée toto.mat, tapez

```
>>load toto
```

Puis, pour vérifier que vos variables ont été chargées, tapez

```
>>who
```

On peut n'avoir à sauvegarder qu'un certain nombre de variables, il suffit alors de taper (par exemple pour sauvegarder la variable C)

```
>>save toto1 C
```

Vérifiez l'existence des fichiers toto.mat et toto1.mat. Tapez alors

```
>>dir
```

On peut sauvegarder la variable C sous forme ascii. Tapez pour cela

```
>>save toto2.dat C -ascii
```

Pour vérifier si l'on peut interpréter facilement toto et toto2, il faut les ouvrir avec *WordPad* ou avec *Bloc-notes*.

Vérifiez que toto est un fichier binaire qu'on ne peut pas interpréter.

Vérifier que toto2 est bien un fichier ascii que l'on peut interpréter.

## 1.2 Création d'un fichier .m

Afin de ne pas perdre votre travail à la mise hors tension de l'ordinateur, il faut enregistrer vos programmes dans un fichier .m

Ce fichier est rempli avec l'éditeur Matlab. Cliquez sur New, puis script.

Enregistrez immédiatement votre fichier .m dans votre dossier de travail, sous le nom

Nom\_1\_Nom\_2\_CAO\_TP1.m

A partir de maintenant, les exercices que vous réaliserez pendant le TP seront rédigés sur ce fichier .m. Pour exécuter ce programme, tapez son nom au prompt (>>) puis tapez sur Entrée.

## 1.3 Opérations sur les matrices et les vecteurs

**☒ Stop Exercice :** Le produit scalaire de deux vecteurs A et B ayant le même nombre de coordonnées est le nombre défini par :  $\text{scalaire}=A' * B$ .  
Effectuer le produit scalaire des vecteurs  $A=[-1;1]$  et  $B=[1;1]$ .

**☒ Stop Exercice :** Essayer les commandes suivantes et trouver leur interprétation :

```
A=[1 2 3; 4 5 6; 7 8 0];
B=[1 2 0
3 1 4
4 1 2];
x=[-1.3 sqrt(3) (1+2+3)*4/5];
x(2)
x(5)=abs(x(1))
B=[B; [2 3 4]]
B=B(1:3, :)
C=[1 2; 3 4] + sqrt(-1)*[5 6; 7 8]
i=sqrt(-1)
C=[1+5*i 2+6*i; 3+7*i 4+8*i]
w=2*exp(i*3)
D=eye(5)
D=ones(5)
D=zeros(5)
D=rand(5)
```

**☒** *Stop Exercice* : Essayer les commandes suivantes et trouver leur interprétation :

```
B=A '  
y=x '  
C=C '  
C=A+B  
z=y-1  
z '*y  
y '*z  
z*y '  
x=[1 0 2] '  
b=A*x  
z=A\b  
z=(b '/A') '
```

Précédées du symbole « . », les opérations suivantes peuvent être effectuées élément par élément : **multiplication, division, puissance**. Ce type d'opération existe déjà implicitement pour le + et le -.

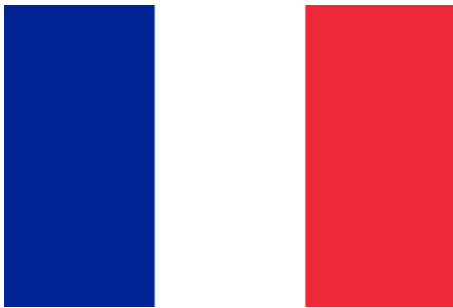
**☒** *Stop Exercice* : Essayer et interpréter les commandes suivantes :

```
x=[1 2 3] ; y=[4 5 6] ;  
z=x.*y  
z=x.^y  
z=x.^2
```

## 1.4 Notion de rang

On considère maintenant la notion de rang, qui est étroitement liée aux décompositions de matrices.

Les figures 1 et 2 représentent les drapeaux Français et Allemands. On montrera que ce sont des matrices de rang 1.



**☒ Stop Exercice :** Donnez une représentation matricielle des drapeaux Français et Allemand.

Sur papier: exprimez F et G par le produit d'un vecteur colonne par un vecteur ligne.  
Vous prendrez les valeurs 1, 2 et 3 à chaque fois pour distinguer les couleurs.  
Quel est le rang de ces matrices ?

Sur Matlab : Créez les matrices F et G qui correspondent aux deux drapeaux.  
Vous choisirez une taille 3 x 3 (3 lignes et 3 colonnes) à chaque fois.  
Créez les matrices F et G en appliquant la formule  $u*v'$  où vous choisirez adéquatement u et v à chaque fois.

Nous considérons maintenant les formes dites canonique et matricielle des décompositions de matrices.

Exemple de forme canonique pour une matrice de rang 2:

$$M = [a; c] * [1 \ 2] + [b; d] * [3 \ 4]$$

Exemple de forme matricielle pour une matrice de rang 2:

$$M = [a \ b; c \ d] * [1 \ 2; 3 \ 4]$$

 **Stop Exercice :**

Sur papier: développez la matrice **M**.

Si on note  $u_1, v_1$  et  $u_2, v_2$  les couples de vecteurs de la décomposition canonique on peut écrire:

$$M = u_1 * v_1' + u_2 * v_2'$$

Exprimez la forme matricielle en fonction de  $u_1, v_1, u_2, v_2$ , puis en fonction des matrices **U** et **V** dont les vecteurs colonnes sont respectivement  $u_1, u_2$ , et  $v_1, v_2$ .

 **Stop Exercice :**

Sur Matlab:

créez les vecteurs  $u_1, u_2, v_1, v_2$ . Construisez la matrice **M** à partir de ces vecteurs.

Créez les matrices **U** et **V** dont les colonnes sont respectivement  $u_1, u_2$ , et  $v_1, v_2$ .

Construisez la matrice **M** à partir de ces matrices.

## 2. Décomposition en valeurs singulières

### 2.1. Matrices simples de format 1 x 2

Nous souhaitons vérifier que le calcul d'une svd passe par le calcul de deux décompositions en éléments propres.

#### *Stop Exercice :*

Sur Matlab :

On considère la matrice  $SSimple = [3 \ 4]$

Avec les lignes de code Matlab ci-dessous, calculez la svd de  $SSimple$  de deux façons : directement et en passant par la décomposition en éléments propres :

```
[U, D, V]=svd(SSimple)
```

```
[E1, d1]=eig(SSimple*SSimple')
```

```
[E2, d2]=eig(SSimple'*SSimple)
```

Les vecteurs singuliers gauche sont stockés dans les colonnes de  $U$ , les vecteurs singuliers droite sont stockés dans les colonnes de  $V$ .

Comparez les vecteurs singuliers aux vecteurs propres : quel est le format des vecteurs dans chaque cas ?

Comparez les valeurs singulières aux valeurs propres. Qu'en déduisez-vous ?



## 2.2 Application images drapeaux

Nous souhaitons maintenant comparer la décomposition ‘intuitive’ des matrices drapeaux et la décomposition en valeurs singulières.

**☒** *Stop Exercice :*

Sur Matlab : Créez les matrices F et G en utilisant les vecteurs suivants :  
 $u=[1 \ 1 \ 1]$  et  $v=[1 \ 2 \ 3]$ .

**☒** *Stop Exercice :*

Sur Matlab : appliquez la fonction SVD aux deux matrices F et G.  
Par exemple pour la matrice F :

$[U,D,V]=svd(F)$

Quels sont les vecteurs singuliers gauche et droite de F et G ? Les valeurs singulières ?

Pour vérifier que la décomposition est correcte, calculez Fverif et Gverif en utilisant la formule suivante, par exemple pour la matrice Fverif :

$F_{verif}=U*S*V'$

Affichez F-Fverif et G-Gverif de la façon suivante, par exemple pour F :

$disp(F-F_{verif})$

Faites quelques commentaires, comparez la décomposition en valeurs singulières à la décomposition intuitive.

### 3. Théorème d'Eckart-Young

L'objectif de l'exercice est de montrer que, parmi les décompositions en somme de matrices de rang 1, la décomposition en valeurs singulières est celle qui conduit à l'erreur quadratique minimale.

#### *Stop Exercice :*

Nous souhaitons maintenant illustrer le théorème d'Eckart-Young.

Considérez la somme des deux matrices drapeau  $F+G$ .

Calculez la troncature de rang 1 de  $F+G$  de deux façons : en tronquant la somme des décompositions intuitives, et en tronquant la décomposition en valeurs singulières.

Pour réaliser la troncature, sélectionnez le premier vecteur colonne de  $U$  et le premier vecteur colonne de  $V$ .

Pour analyser les résultats de troncature obtenus, calculez la norme de Frobenius de l'erreur entre résultat de la troncature et matrice de départ ( $F+G$ ).

*Indice :* la norme de Frobenius d'une matrice  $M$  est calculée sous Matlab par `norm(M,'fro')`.

Affichez avec la fonction `imagesc` de Matlab les matrices tronquées obtenues. Par exemple :

```
figure(1)
imagesc(F)
```