

Corrigé TP1 MATLAB

Exercice 1

Tester et comprendre les deux lignes suivantes

$A = \begin{bmatrix} 1 & 2 & -1 & 1 \\ -1 & 1 & 0 & 3 \end{bmatrix}$

`find(A>0)`

Corrigé de l'exercice 1

% On tape dans l'invite de Matlab

```
>> A=[1 2 -1 1; -1 1 0 3]
```

A =

```
1 2 -1 1
```

```
-1 1 0 3
```

% La commande précédente permet de créer une variable de type tableau (matrice) à deux

% lignes et quatre colonnes. La première ligne contient les termes spécifiés par la liste

% 1 2 -1 1 qui sont séparés par un blanc, alors que les termes de la deuxième ligne

% commencent après le point virgule de séparation des lignes et sont définis par la liste

% -1 1 0 3.

%

% On tape dans l'invite de Matlab

```
find(A>0)
```

ans =

```
1
```

```
3
```

```
4
```

```
7
```

```
8
```

rouver les indices des termes d'un tableau selon le critère

% spécifié entre parenthèses. Ici le critère est $A > 0$ qui signifie donc que l'on veut les termes
% du tableau A qui sont strictement positifs. La numérotation avec un seul indice adoptée par
% Matlab est telle que les termes sont numérotés par colonne de haut en bas et de gauche à
% droite. Le premier terme de A est donc 1, le deuxième est -1, le troisième est 2, ..., le
% septième est 1 et le huitième est 3. La commande *find* produit comme résultat la liste
% 1 3 4 7 8 qui représente les indices des termes (et non pas les termes) de la matrice A qui
% satisfont le critère.

Exercice 2

On note u, v et w les vecteurs suivants : $u = \begin{bmatrix} 1 & -1 & 2 \end{bmatrix}^t$, $v = \begin{bmatrix} 10 & -1 & 3 \end{bmatrix}^t$,
 $w = \begin{bmatrix} 5 & -1 & 4 \end{bmatrix}^t$

1. Calculer $3u$, $\|u\|_2$, $2u - v + 5w$, $\|2u - v + 5w\|_1$, $\|w - 4v\|_\infty$.
2. Déterminer l'angle de droites formé par les vecteurs v et w .

Corrigé de l'exercice 2

1.

% On tape dans l'invite de Matlab

```
>> u=[1 -1 2].'
```

u =

1

-1

2

% Noter la présence du symbole *'* en fin de l'instruction qui signifie le transposé selon la

% syntaxe de Matlab.

% On tape ensuite dans l'invite de Matlab

```
>> v=[10 -1 3].'
```

10

-1

3

>> w=[5 -1 4].'

w =

5

-1

4

% Les trois instructions précédentes ont permis d'affecter les valeurs aux trois variables de
% type tableaux u, v et w. On aurait pu placer un point virgule, c'est-à-dire le symbole ; en
% fin de chaque instruction si l'on n'était pas intéressé par l'affichage des valeurs de ces trois
% variables sur la fenêtre de commandes.

% Pour calculer 3u, on tape dans l'invite de Matlab

>> 3*u

ans =

3

-3

6

% On obtient donc le résultat qui s'écrit en notations mathématiques $[3 \ -3 \ 6]^t$

% Avant de faire la suite faisons un petit rappel sur les normes que l'on peut définir sur un

% espace vectoriel. Nous avons la norme euclidienne dite aussi norme 2 qui est définie

% pour un vecteur $a = [x \ y \ z]^t$ par $\|a\|_2 = \|a\| = \sqrt{x^2 + y^2 + z^2}$. Il y a aussi la norme 1

% notée $\|a\|_1 = |x| + |y| + |z|$ et la norme uniforme (norme infinie) notée $\|a\|_\infty = \max\{|x|, |y|, |z|\}$.

% Nous pouvons aussi définir la norme p (pour $p > 1$) par $\|a\|_p = \left(|x|^p + |y|^p + |z|^p\right)^{\frac{1}{p}}$.

% Matlab permet de calculer toutes ces normes pour un vecteur $a \in \mathbb{R}^n$ représenté par

n tableau ligne ou colonne par la simple commande

% **$norm(a,p)$** où a est le tableau dont on veut calculer la norme et p l'ordre de la norme: $p=1$

% pour la norme 1, $p=2$ ou bien carrément **$norm(a)$** pour calculer la norme euclidienne,...,

% $p=\infty$ pour la norme uniforme.

% Pour calculer $\|u\|_2$, on tape dans l'invite de Matlab

```
>> norm(u,2)
```

```
ans =
```

```
2.4495
```

% La réponse est donc $\|u\|_2 = 2.4495$. On aurait pu taper **$norm(u)$** et obtenir le même résultat,

% car la norme par défaut est la norme euclidienne.

% Pour calculer $2u-v+5w$, on tape dans l'invite de Matlab

```
>> 2*u-v+5*w
```

```
ans =
```

```
17
```

```
-6
```

```
21
```

% Le résultat est donc $\begin{bmatrix} 17 & -6 & 21 \end{bmatrix}^t$. Noter qu'il faut bien mettre * qui représente

% l'opération de multiplication et qu'on ne peut pas faire le calcul en utilisant la

% notation mathématique $2u-v+5w$ qui produirait un message d'erreur.

% Pour calculer $\|2u-v+5w\|_1$, il suffit de taper à l'invite de Matlab

```
>> norm(2*u-v+5*w,1)
```

```
ans =
```

```
44
```

% Le résultat est donc $\|2u-v+5w\|_1 = 44$.

% Pour calculer $\|w-4v\|_\infty$, on tape dans l'invite de Matlab

ans =

35

% Avec Matlab l'infini ∞ est noté Inf (il s'agit bien sûr de l'infini numérique et non pas % de l'infini mathématique).

2.

% Pour pouvoir calculer l'angle de droite défini par les vecteurs v et w, il suffit de calculer

% le cosinus de l'angle formé par les deux vecteurs v et w. Pour cela, on fait appel à la

% formule: $v \cdot w = \|v\| \|w\| \cos(\angle(v, w))$, qui donne: $\cos(\angle(v, w)) = (v \cdot w) / (\|v\| \|w\|)$. Pour calculer le

% produit scalaire des deux vecteurs v et w qui apparaît dans la formule précédente, avec ces

% deux vecteurs qui sont représentés par leurs deux matrices des composantes

% (tableaux colonnes), on peut soit appliquer l'opération $v \cdot w$ qui permet de transformer le

% tableau colonne v en tableau ligne et qui donne ensuite par multiplication algébrique

% avec w le produit scalaire. On peut aussi utiliser la commande $\text{dot}(v, w)$ qui signifie point

% en anglais et qui produit comme résultat le produit scalaire que l'on veut calculer. Il ne faut

% surtout pas utiliser la notation mathématique habituelle du produit scalaire $\cdot$. En fait, cette

% notation a été réservée par Matlab pour déclarer une variable rattachée à une variable de

% type structure. Comme remarque générale avec Matlab, il faut se méfier de chercher à

% obtenir le résultat par application des notations mathématiques.

% Il faut donc apprendre la syntaxe correcte afin d'éviter les messages d'erreurs.

% Matlab ne reconnaît donc que sa propre syntaxe qui n'est pas parfois intuitive (sauf bien

% sûr pour ceux qui l'ont élaborée: l'équipe de *Mathworks*.

% Ayant calculé $\cos(\angle(v, w))$, on peut récupérer l'angle de droites qui est défini à π près par

% inversion de la fonction cosinus, c'est-à-dire avec les notations de Matlab grâce à la

% commande: $\text{acos} = \cos^{-1}$.

% En résumé, le calcul de l'angle de droites défini par les deux vecteurs v et w peut se faire

% en tapant dans l'invite de Matlab, l'instruction suivante :

w)))

ans =

0.3848

% La mesure de l'angle est donnée par défaut en radians. L'angle des droites en question

% mesure donc 0.3848 rd (avec une précision de 4 chiffres significatifs). Noter en particulier

% la nécessité de placer correctement les parenthèses. Les opérations se font avec un ordre

% décidé par les parenthèses. On peut s'amuser à ce titre à exécuter les opérations suivantes

% $12/3*4 = 16$ et $12/(3*4)=1$. La première opération correspond à $\frac{12}{3} \times 4$, alors que la

% deuxième correspond à $\frac{12}{3 \times 4}$. Noter qu'on peut calculer la deuxième avec $12/3/4$.

Exercice 3

On pose $A = \begin{bmatrix} 1 & -1 & 7 \\ -4 & 2 & 11 \\ 8 & 0 & 3 \end{bmatrix}$, $B = \begin{bmatrix} 3 & -2 & -1 \\ 7 & 8 & 6 \\ 5 & 1 & 3 \end{bmatrix}$

Que font les instructions suivantes?

$3 * A$; $A * B$; $A ./ B$; $\cos(A)$; $\exp(B)$.

Corrigé de l'exercice 3

% Il s'agit d'opérations terme à terme, opérations dites de tableaux (array opérations). Ces

% opérations n'ont pas de sens algébrique propre comme c'est le cas de la multiplication des

% matrices. Les arguments des fonctions de Matlab peuvent être des variables de type tableau.

% Ainsi la fonction $\cos$ peut calculer en bloc tous les cosinus des termes du tableau qui lui

% est présenté comme argument. On peut de la sorte faire des opérations en bloc et éviter

% de distinguer les cas. Cela permet de gagner en vitesse de programmation. L'une des

% qualités de Matlab c'est justement ça: Matlab est bien **Matrix Laboratoty**.

% Les opérations terme à terme ne sont possibles que si les deux tableaux admettent

% les mêmes dimensions.

dans cet exercice est obtenue par les instructions suivantes :

```
>> A=[1 -1 7; -4 2 11; 8 0 3]
```

```
A =
```

```
1 -1 7
```

```
-4 2 11
```

```
8 0 3
```

```
%
```

```
>> B=[3 -2 -1; 7 8 6; 5 1 3]
```

```
B =
```

```
3 -2 -1
```

```
7 8 6
```

```
5 1 3
```

```
%
```

```
>> 3*A
```

```
ans =
```

```
3 -3 21
```

```
-12 6 33
```

```
24 0 9
```

```
%
```

```
>> A.*B
```

```
ans =
```

```
3 2 -7
```

```
-28 16 66
```

```
40 0 9
```

```
%
```

```
>> A./B
```

```
ans =
```

```
0.333333333333333 0.500000000000000 -7.000000000000000
```

```
-0.571428571428571 0.250000000000000 1.833333333333333
```

0 1.0000000000000000

% On voit dans ce dernier résultat qu'en cas de besoin Matlab affiche le résultat avec le
% maximum de précision possible, c'est-à-dire la double précision. Vous verrez qu'avec
% Maple, la précision pourra être choisie a priori. Maple est plus précis que Matlab. Mais,
% Maple ne permet pas de faire la même chose que Matlab. A chacun de ces deux logiciels
% correspond son domaine d'application. Ils sont complémentaires.

>> cos(A)

ans =

0.540302305868140 0.540302305868140 0.753902254343305
-0.653643620863612 -0.416146836547142 0.004425697988051
-0.145500033808614 1.0000000000000000 -0.989992496600445

%

>> exp(B)

ans =

1.0e+003 *
0.020085536923188 0.000135335283237 0.000367879441171
1.096633158428459 2.980957987041728 0.403428793492735
0.148413159102577 0.002718281828459 0.020085536923188

% Ne pas oublier ici que tous les termes sont multipliés par 1000 comme l'indique la ligne

% en surbrillance jaune. 1000 en notation Matlab s'écrit 1.0 e+003, on peut aussi l'écrire

% tout simplement 1.e3 qui signifie la même chose que 10^3 .

Exercice 4

On pose $A = \begin{bmatrix} 1 & -1 & 7 \\ -4 & 2 & 11 \\ 8 & 0 & 3 \end{bmatrix}$.

Que font les instructions suivantes : $v = [1, 2, 4, 7]$; $A(v) = A(v) + 0.01$;

$B(v) = \text{abs}(B(v))$ **Corrigé de l'exercice 4**

es suivantes d'affectation dans l'invite de Matlab

```
>> A=[1 -1 7; -4 2 11; 8 0 3]
```

```
A =
```

```
1 -1 7
```

```
-4 2 11
```

```
8 0 3
```

```
%
```

```
>> v=[1, 2, 4, 7]
```

```
v =
```

```
1 2 4 7
```

**% On n'était pas obligé de mettre la virgule comme séparation des colonnes de v car elle est
% équivalente à l'espace qui remplit exactement la même fonction de séparation.**

% Tapons d'abord

```
>> A(v)
```

```
ans =
```

```
1 -4 -1 7
```

**% Cette commande permet d'extraire tous les termes de la matrice A dont les indices sont
% spécifiés dans la liste v. Bien sûr conformément à la convention de numérotation des
% termes de la matrice avec un seul indice. Remarquer que la syntaxe de Matlab ne respecte
% absolument pas la notation mathématique habituelle réservée à $A(v)$ et qui signifie en
% maths, l'image de v donnée par A. Prudence donc de vouloir utiliser Matlab avec ses
% seules connaissances des notations mathématiques, il faut apprendre ceux choisis par les
% développeurs de Matlab.**

```
>> A(v)=A(v)+0.01
```

```
A =
```

```
1.0100000000000000 -0.9900000000000000 7.0100000000000000
```

```
-3.9900000000000000 2.0000000000000000 11.0000000000000000
```

```
8.0000000000000000 0 3.0000000000000000
```

% La commande $A(v)=A(v)+0.01$ s'interprète de la façon suivante: modifier tous les termes

ont spécifiés dans la liste des indices v en les augmentant de

% la quantité 0.01.

% Pour étudier ce que fait la commande $B(v)=abs(B(v))$ prenons comme exemple le tableau

% A. Mais revenons au tableau A non modifié par l'instruction précédente $A(v)=A(v)+0.01$.

```
>> A=[1 -1 7; -4 2 11; 8 0 3]
```

A =

```
1  -1  7
```

```
-4   2 11
```

```
8   0  3
```

%

```
>> A(v)=abs(A(v))
```

A =

```
1   1  7
```

```
4   2 11
```

```
8   0  3
```

% On remarque que les termes de A dont les indices sont spécifiés dans la liste v, ont été

% substitués par leurs valeurs absolues. Les autres n'ont pas été touchés. Pour mieux s'en

% convaincre prenons cet exemple :

```
>> B=[1 -1 7; -4 2 11; 8 0 -3]
```

B =

```
1  -1  7
```

```
-4   2 11
```

```
8   0 -3
```

```
>> B(v)=abs(B(v))
```

B =

```
1   1  7
```

```
4   2 11
```

```
8   0 -3
```

% -3 dont l'indice est 9 qui n'est pas dans la liste des indices v n'a pas été modifié.

1. Créer un vecteur de 11 composantes contenant les nombres $-5, -4, \dots, 4, 5$
2. Créer un vecteur de 1001 composantes contenant les nombres $-500, -499, -498, \dots, 499, 500$
3. Créer un vecteur u contenant 10 valeurs entre 0 et π séparées par un incrément constant.

Corrigé de l'exercice 5

1.

% La commande à utiliser est *linspace*. On écrit alors dans l'invite de Matlab.

```
>> linspace(-5,5,11)
```

ans =

```
-5 -4 -3 -2 -1 0 1 2 3 4 5
```

% Matlab utilise par défaut la variable ans pour stocker le résultat. ans est le diminutif du mot

% anglais answer qui signifie réponse. Mais, il est préférable de donner un autre nom à la

% variable créée, par exemple v1.

```
>> v1=linspace(-5,5,11)
```

v1 =

```
-5 -4 -3 -2 -1 0 1 2 3 4 5
```

2.

% On peut vérifier que le nombre total des termes est $500 - (-500) + 1 = 1001$.

% Il suffit d'utiliser la commande *linspace*. Mais, ici l'affichage du résultat est encombrant.

% On va donc placer un point virgule en fin de la ligne de la commande. On peut tester

% l'affichage obtenu en retirant le point virgule.

```
>> v2=linspace(-500,500,1001) ;
```

% On peut se permettre de vérifier le premier terme par $v2(1) = -500$, puis $v2(2) = -499$ et enfin

th donne le nombre des termes d'un tableau (sa dimension).

% En tapant `length(v2)`, on obtient

```
>> length(v2)
```

ans =

1001

% Nous avons vu ici comment créer des grands tableaux avec la commande *linspace*. Avec

% Matlab ça se fait en bloc. Dans l'exemple précédent le faire à la main est une punition !

3.

% Il suffit de taper

```
>> v3=linspace(0,pi,10)
```

v3 =

Columns 1 through 3

0 0.349065850398866 0.698131700797732

Columns 4 through 6

1.047197551196598 1.396263401595464 1.745329251994330

Columns 7 through 9

2.094395102393195 2.443460952792061 2.792526803190927

Column 10

3.141592653589793

% Au passage le nombre π est bien connu dans Matlab. C'est la variable pi (attention ! ne pas

% la modifier en choisissant d'appeler une autre variable pi, Matlab ne vous le dira pas !). En

% particulier, dès que vous lancez Matlab, taper *clear* sur la fenêtre de commande pour

% nettoyer la mémoire et retrouver les valeurs par défaut de π et de l'imaginaire pur i.

% Matlab est un peu lâche à ce sujet, il ne vous préviendra pas. Maple vous empêchera

% d'écraser Pi (avec un P majuscule qui sert à désigner le nombre π et non pas minuscule

% comme c'est le cas avec Matlab). C'est un autre esprit qui a animé ceux qui ont

n jardin secret.

Exercice 6

On considère l'approximation numérique de la dérivée d'une fonction par différences finies de premier ordre avant

$$f'(x) \approx \frac{f(x+h)-f(x)}{h} \quad (1)$$

1. Définir un tableau d'abscisses x pour discrétiser l'intervalle $[0, \pi]$ avec le pas $h=0.1$.
 2. Calculer dans un tableau d la dérivée exacte de la fonction $x \mapsto \sin(x)$ aux points x et dans un tableau $d1$ les valeurs approchées par la formule de l'équation (1).
 3. Calculer l'erreur maximale commise en faisant cette approximation sur l'intervalle $[0, \pi]$.
- Diminuer h , que remarque-t-on ?

Corrigé exercice 6

1.

% On tape dans l'invite de Matlab

```
>> x=0:0.1:pi
```

x =

Columns 1 through 3

0 0.100000000000000 0.200000000000000

Columns 4 through 6

0.300000000000000 0.400000000000000 0.500000000000000

Columns 7 through 9

0.600000000000000 0.700000000000000 0.800000000000000

0.9000000000000000 1.0000000000000000 1.1000000000000000

Columns 13 through 15

1.2000000000000000 1.3000000000000000 1.4000000000000000

Columns 16 through 18

1.5000000000000000 1.6000000000000000 1.7000000000000000

Columns 19 through 21

1.8000000000000000 1.9000000000000000 2.0000000000000000

Columns 22 through 24

2.1000000000000000 2.2000000000000000 2.3000000000000000

Columns 25 through 27

2.4000000000000000 2.5000000000000000 2.6000000000000000

Columns 28 through 30

2.7000000000000000 2.8000000000000000 2.9000000000000000

Column 31

3.0000000000000000

% Comme je ne vais pas imprimer le présent fichier, je me suis permis de ne pas écrire le

% point virgule. Ici, il faut noter que c'est le pas qui a été fixé et non pas le nombre de points.

% Si l'on avait fixé le nombre de points, on aurait utilisé la commande *linspace*. En fixant, le

% nombre de pas, le dernier point ne fait pas partie nécessairement du tableau, il indique tout

% simplement une borne à ne pas dépasser par le dernier terme du tableau crée.

2.

% Tout le monde sait que la dérivée exacte de la fonction $x \mapsto \sin(x)$ est la

% fonction $x \mapsto \cos(x)$. D'où

>> d=cos(x);

% en évitant l'affichage des 31 termes qui composent ce tableau.

% Le tableau d1 peut être crée de la façon suivante :

>> h=0.1 ;

**% Remarquer que l'on fait l'opération en bloc. Tous les taux d'accroissement sont calculés
% d'un seul coup. Avec un langage comme le Fortran par exemple, il aurait fallu utiliser la
% notion de boucle pour définir d1.**

3.

% L'erreur maximum est donnée par la norme uniforme de d-d1. soit

```
>> erreur=norm(d-d1,Inf)
```

```
erreur =
```

```
0.049951037293196
```

% Pour étudier l'influence du pas de discrétisation h, on peut utiliser la séquence suivante :

```
>> h=0.01; x=0:h:pi; d=cos(x); d1=(sin(x+h)-sin(x))/h;
```

```
erreur=norm(d-d1,Inf)
```

```
erreur =
```

```
0.004999970020186
```

%

```
>> h=0.001; x=0:h:pi; d=cos(x); d1=(sin(x+h)-sin(x))/h; erreur=norm(d-d1,Inf)
```

```
erreur =
```

```
4.999999325069332e-004
```

%

```
>> h=1.e-6; x=0:h:pi; d=cos(x); d1=(sin(x+h)-sin(x))/h; erreur=norm(d-d1,Inf)
```

```
erreur =
```

```
5.001098949791965e-007
```

% On voit que l'erreur diminue de manière proportionnelle à h . On dit que d1 est une

% approximation de premier ordre.

% En essayant h=1.e-10, la mémoire de mon ordinateur a été saturée. J'ai reçu le message

% suivant :

```
>> h=1.e-8; x=0:h:pi; d=cos(x); d1=(sin(x+h)-sin(x))/h; erreur=norm(d-d1,Inf)
```

??? Out of memory. Type HELP MEMORY for your options.

% En utilisant Matlab, il faut faire attention à la dimension des tableaux manipulés.

atlab met un peu plus de temps pour afficher le résultat sur

% l'écran