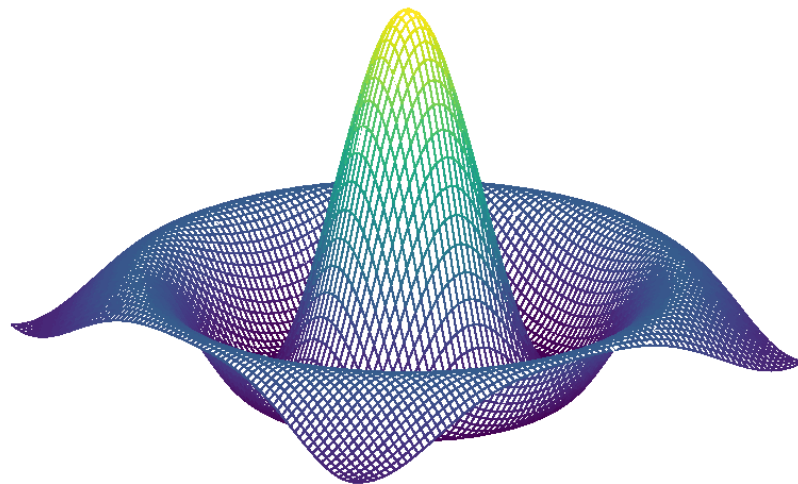

ELEMENTS D'ALGEBRE LINEAIRE AVEC OCTAVE

JAOUAD MADKOUR



Version 2

Table des matières

I	Introduction aux méthodes numériques	11
1	Nombres réels	13
1.1	Initialisation d'une variable	13
1.1.1	Principe	13
1.1.2	Règles à respecter	13
1.1.3	Sensibilité à la casse	14
1.1.4	Désactivation de l'affichage	14
1.2	Opérations arithmétiques	14
1.2.1	Addition	14
1.2.2	Soustraction	14
1.2.3	Multiplication	15
1.2.4	Division	15
1.3	Puissances, racines et valeur absolue	15
1.3.1	Puissances	15
1.3.2	Racines	15
1.3.3	Valeur absolue	16
1.4	Exponentielle et logarithmes	16
1.4.1	Exponentielle	16
1.4.2	Logarithmes	16
1.5	Trigonométrie	17
1.5.1	Sinus	17
1.5.2	Cosinus	17
1.5.3	Tangente	17
1.5.4	Cotangente	17
1.6	Constantes particulières	17
1.6.1	Constante d'Archimède ou nombre de Ludolph π	17
1.6.2	Constante de Neper ou nombre d'Euler e	17
2	Matrices et vecteurs	19
2.1	Initialisation d'une matrice et d'un vecteur	19

2.1.1	Initialisation d'une matrice	19
2.1.2	Initialisation d'un vecteur ligne	19
2.1.3	Initialisation d'un vecteur colonne	19
2.2	Redimensionnement et modification d'une matrice	20
2.2.1	Taille d'une matrice	20
2.2.2	Longueur d'une matrice ou d'un vecteur	20
2.2.3	Extraction des éléments d'une matrice	20
2.2.4	Modification des éléments d'une matrice	20
2.2.5	Extraction d'une sous-matrice	21
2.2.6	Concaténation de matrices	21
2.2.7	Empilement des colonnes d'une matrice	21
2.3	Matrice nulle, matrice de 1 et matrice identité	22
2.3.1	Matrice nulle	22
2.3.2	Matrice de 1	22
2.3.3	Matrice identité	22
2.4	Matrice de valeurs manquantes et matrice de valeurs non numériques	22
2.4.1	Matrice de valeurs manquantes	22
2.4.2	Matrice de valeurs non numériques	23
2.5	Matrice diagonale et matrice scalaire	23
2.5.1	Matrice diagonale	23
2.5.2	Matrice scalaire	23
2.6	Matrices triangulaires	24
2.6.1	Matrice triangulaire inférieure	24
2.6.2	Matrice triangulaire strictement inférieure	24
2.6.3	Matrice triangulaire supérieure	24
2.6.4	Matrice triangulaire strictement supérieure	25
2.7	Transposée et opposée d'une matrice	25
2.7.1	Transposée d'une matrice	25
2.7.2	Opposée d'une matrice	25
3	Opérations sur les matrices	27
3.1	Addition des matrices	27
3.2	Soustraction des matrices	28
3.3	Multiplication d'une matrice par un scalaire	28
3.4	Produit scalaire	29
3.5	Produit matriciel	29
3.6	Puissances d'une matrice carrée	29
3.7	Exponentielle d'une matrice carrée	30
3.8	Racine carrée d'une matrice carrée	31

<i>TABLE DES MATIÈRES</i>	5
3.9 Opérations élément-par-élément	31
3.9.1 Multiplication élément-par-élément	31
3.9.2 Puissances élément-par-élément	31
3.9.3 Division élément-par-élément	32
4 Caractéristiques des matrices carrées	33
4.1 Trace	33
4.2 Déterminant	33
4.3 Inverse	34
4.4 Valeurs et vecteurs propres	34
5 Systèmes linéaires	37
6 Importer et sauvegarder un fichier de données	39
6.1 Importer un fichier Excel	39
6.2 Importer un fichier Texte	39
7 Graphiques	41
7.1 Séquence de valeurs linéairement espacées	41
7.2 Meshgrid	42
7.3 Graphiques 2D	42
7.4 Graphiques 3D	42
II Initiation à la programmation avec Octave	43
8 Opérateurs booléens	45
9 Boucles	47
9.1 Boucle « For »	47
9.2 Boucle « While »	47
10 Tests logiques	49
10.1 Test logique simple « If »	49
10.2 Tests logiques imbriqués « If – Elseif »	49
11 Scripts et fonctions	51
11.1 Scripts	51
11.2 Fonctions	51

Listings

11.1	Commandes utiles	53
11.2	Nombres réels	53
11.3	Matrices et vecteurs	54
11.4	Opérations sur les matrices	56
11.5	Caractéristiques des matrices carrées	57
11.6	Systèmes linéaires	58
11.7	Importer des fichiers de données	58
11.8	Graphiques	59
11.9	Boucles	60
11.10	Tests logiques	60
11.11	Scripts et fonctions	61

Introduction

Pourquoi cette formation ?

1. Acquérir un savoir-faire ;
2. Vérifier les résultats des exercices ;
3. L'économie de la connaissance est l'avenir.

Pourquoi Octave ?

1. Octave est gratuit ;
2. Octave est la meilleure alternative de MatLab qui coûte cher ;
3. Avantages et inconvénients de MatLab et de Octave.

Première partie

Introduction aux méthodes numériques

Chapitre 1

Nombres réels

1.1 Initialisation d'une variable

1.1.1 Principe

```
x = 5
```

1.1.2 Règles à respecter

Utiliser uniquement des nombres, des lettres majuscules ou minuscules sans les accents, le tiret du 8.

Ne pas commencer par un nombre

```
2x = 5
```

```
parse error :  
  
  syntax error  
  
>>> 2x = 5  
      ^
```

```
x2 = 5
```

```
x2 = 5
```

```
x2x = 5
```

```
x2x = 5
```

Ne pas laisser d'espace

```
x y = 5
```

```
>> x y = 5
```

```
error: 'x' undefined near line 1 column 1
```

```
xy = 5
```

```
xy = 5
```

1.1.3 Sensibilité à la casse

```
x = 5
```

```
>> X
```

```
error: 'X' undefined near line 1 column 1
```

1.1.4 Désactivation de l'affichage

```
a = -1  
b = 3 ;
```

```
a = -1
```

1.2 Opérations arithmétiques

1.2.1 Addition

```
s = a + b
```

```
s = 2
```

1.2.2 Soustraction

```
d = a - b
```

```
d = -4
```

1.2.3 Multiplication

```
m = a * b
```

```
m = -3
```

1.2.4 Division

```
r = a / b
```

```
r = -0.33333
```

```
t = a / 0
```

```
warning: division by zero
warning: called from
SCRIPT1 at line 21 column 3
t = -Inf
```

1.3 Puissances, racines et valeur absolue

1.3.1 Puissances

```
a2 = a^2
a3 = a^3
ab = a^b
```

```
a2 = 1
a3 = -1
ab = -1
```

1.3.2 Racines

```
rcb = sqrt( b )
rqb = b^(1/3)      % Racine cubique du nombre b
```

```
rcb = 1.7321
rqb = 1.4422
```

```
rca = sqrt( a )      % Racine carrée du nombre a
```

```
rca = 0 + 1i
```

```
rqa = a^(1/3)      % Racine cubique du nombre a
```

```
rqa = 0.50000 + 0.86603i
```

1.3.3 Valeur absolue

```
va = abs( a )     % Valeur absolue du nombre a
```

```
va = 1
```

1.4 Exponentielle et logarithmes

1.4.1 Exponentielle

```
ea = e^( a )     % Exponentielle du nombre a
ea = exp( a )    % Exponentielle du nombre a
```

```
ea = 0.36788
ea = 0.36788
```

1.4.2 Logarithmes

```
logb = log( b )   % Logarithme naturel du nombre b
log2b = log2( b ) % Logarithme base 2 du nombre b
log10b = log10( b ) % Logarithme base 10 du nombre b
```

```
logb = 1.0986
log2b = 1.5850
log10b = 0.47712
```

```
loga = log( a )   % Logarithme naturel du nombre a
log2a = log2( a ) % Logarithme base 2 du nombre a
log10a = log10( a ) % Logarithme base 10 du nombre a
```

```
loga = 0.00000 + 3.14159i
log2a = 0.00000 + 4.53236i
log10a = 0.00000 + 1.36438i
```


1.5 Trigonométrie

1.5.1 Sinus

```
sinb = sin( b )      % Sinus du nombre b ( b en radian )
```

```
sinb = 0.14112
```

1.5.2 Cosinus

```
cosb = cos( b )     % Consinus du nombre b ( b en radian )
```

```
cosb = -0.98999
```

1.5.3 Tangente

```
tanb = tan( b )     % Tangente du nombre b ( b en radian )
```

```
tanb = -0.14255
```

1.5.4 Cotangente

```
cotb = cot( b )     % Cotangente du nombre b ( b en radian )
```

```
cotb = -7.0153
```

1.6 Constantes particulières

1.6.1 Constante d'Archimède ou nombre de Ludolph π

```
p = pi              % Constante d'Archimède ou nombre de Ludolph Pi
```

```
p = 3.1416
```

1.6.2 Constante de Neper ou nombre d'Euler e

```
q = e              % Constante de Neper ou nombre d'Euler e
```

```
q = 2.7183
```


Chapitre 2

Matrices et vecteurs

2.1 Initialisation d'une matrice et d'un vecteur

2.1.1 Initialisation d'une matrice

```
A = [ 1 2 3
      4 5 6
      7 8 9 ] % Initialisation de la matrice A
A = [ 1 , 2 , 3 ; 4 , 5 , 6 ; 7 , 8 , 9 ] % Initialisation de la matrice A
```

```
A =

     1     2     3
     4     5     6
     7     8     9
```

2.1.2 Initialisation d'un vecteur ligne

```
L = [ 1 2 3 ]
L = [ 1 , 2 , 3 ]
```

```
L =

     1     2     3
```

2.1.3 Initialisation d'un vecteur colonne

```
C = [ 1
      2
      3 ] % Initialisation du vecteur colonne C
C = [ 1 ;
```

```

    2 ;
    3 ]
C = [ 1 ; 2 ; 3 ]
% Initialisation du vecteur colonne C
% Initialisation du vecteur colonne C

```

```

C =

1
2
3

```

2.2 Redimensionnement et modification d'une matrice

2.2.1 Taille d'une matrice

```

[nl , nc] = size( A )
% Nombre de lignes et de colonnes

```

```

nl = 3
nc = 3

```

2.2.2 Longueur d'une matrice ou d'un vecteur

```

ll = length( L )
lc = length( C )
% Longueur du vecteur ligne L
% Longueur du vecteur colonne C

```

```

ll = 3
lc = 3

```

2.2.3 Extraction des éléments d'une matrice

```

a13 = A( 1 , 3 )
% Atteindre les éléments d'une matrice

```

```

a13 = 3

```

2.2.4 Modification des éléments d'une matrice

```

A( 1 , 3 ) = 99
% Modifier les éléments d'une matrice

```

```

A =

1    2    99
4    5     6
7    8     9

```

2.2.5 Extraction d'une sous-matrice

```
A1 = A( [2,3] , [1,2] )      % Sous-matrice de la matrice A
A2 = A( 2 , [1,2] )        % Sous-matrice de la matrice A
A3 = A( : , [1,2] )        % Sous-matrice de la matrice A
```

```
A1 =
```

```
 4  5
 7  8
```

```
A2 =
```

```
 4  5
```

```
A3 =
```

```
 1  2
 4  5
 7  8
```

2.2.6 Concaténation de matrices

```
AL = [ A ; L ]             % Concaténation verticale de A et L
AC = [ A , C ]             % Concaténation horizontale de A et C
```

```
AL =
```

```
 1  2  3
 4  5  6
 7  8  9
 1  2  3
```

```
AC =
```

```
 1  2  3  1
 4  5  6  2
 7  8  9  3
```

2.2.7 Empilement des colonnes d'une matrice

`vec(X)` ou `x(:)`

2.3 Matrice nulle, matrice de 1 et matrice identité

2.3.1 Matrice nulle

```
O = zeros( 2 , 3 ) % Matrice nulle de taille 2 x 3
```

```
O =
0 0 0
0 0 0
```

2.3.2 Matrice de 1

```
U = ones( 4 , 3 ) % Matrice de 1 de taille 4 x 3
```

```
U =
1 1 1
1 1 1
1 1 1
1 1 1
```

2.3.3 Matrice identité

```
I = eye( 4 ) % Matrice identité d'ordre 4
```

```
I =
Diagonal Matrix
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

2.4 Matrice de valeurs manquantes et matrice de valeurs non numériques

2.4.1 Matrice de valeurs manquantes

```
V = NA( 5 , 6 ) % NA: Not Available (Non disponible)
```

```
V =
NA    NA    NA    NA    NA    NA
NA    NA    NA    NA    NA    NA
NA    NA    NA    NA    NA    NA
NA    NA    NA    NA    NA    NA
NA    NA    NA    NA    NA    NA
```

2.4.2 Matrice de valeurs non numériques

```
W = NaN( 4 , 6 ) % NaN: Not a Number (ce n'est pas un nombre)
```

```
W =
NaN    NaN    NaN    NaN    NaN    NaN
NaN    NaN    NaN    NaN    NaN    NaN
NaN    NaN    NaN    NaN    NaN    NaN
NaN    NaN    NaN    NaN    NaN    NaN
```

2.5 Matrice diagonale et matrice scalaire

2.5.1 Matrice diagonale

```
D = diag( [1 , 2 , 3 , 6 , 7 , 8 ] ) % Matrice diagonale
```

```
D =
Diagonal Matrix
1  0  0  0  0  0
0  2  0  0  0  0
0  0  3  0  0  0
0  0  0  6  0  0
0  0  0  0  7  0
0  0  0  0  0  8
```

2.5.2 Matrice scalaire

```
S = diag( [3 , 3 , 3 , 3 , 3 , 3 ] ) % Matrice scalaire
```

S =

Diagonal Matrix

```

3  0  0  0  0  0
0  3  0  0  0  0
0  0  3  0  0  0
0  0  0  3  0  0
0  0  0  0  3  0
0  0  0  0  0  3

```

2.6 Matrices triangulaires

2.6.1 Matrice triangulaire inférieure

```
TIE = tril( A ) % Matrice triangulaire inférieure
```

TIE =

```

1  0  0
4  5  0
7  8  9

```

2.6.2 Matrice triangulaire strictement inférieure

```
TSI = tril( A , -1 ) % Matrice triangulaire strictement inférieure
```

TSI =

```

0  0  0
4  0  0
7  8  0

```

2.6.3 Matrice triangulaire supérieure

```
TSE = triu( A ) % Matrice triangulaire supérieure
```

TSE =

```

1  2  3
0  5  6
0  0  9

```


2.6.4 Matrice triangulaire strictement supérieure

```
TSS = triu( A , 1 ) % Matrice triangulaire strictement supérieure
```

```
TSS =
```

```
0 2 3
0 0 6
0 0 0
```

2.7 Transposée et opposée d'une matrice

2.7.1 Transposée d'une matrice

```
TA = A' % Transposée de la matrice A
```

```
TA =
```

```
1 4 7
2 5 8
99 6 9
```

2.7.2 Opposée d'une matrice

```
OA = - A % Opposée de la matrice A
```

```
OA =
```

```
-1 -2 -99
-4 -5 -6
-7 -8 -9
```

ccc

Chapitre 3

Opérations sur les matrices

ccc

```
A = [ 1 2 ; 5 6 ; 7 8 ]           % Matrice 3 x 2
B = [ 1 , 2 , 3 ; 4 , 5 , 6 ; 7 , 8 , 9 ] % Matrice 3 x 3
C = [ 1 , 4 , 7 ; 2 , 5 , 6 ; 3 , 8 , 9 ] % Matrice 3 x 3
```

A =

```
1  2
5  6
7  8
```

B =

```
1  2  3
4  5  6
7  8  9
```

C =

```
1  4  7
2  5  6
3  8  9
```

3.1 Addition des matrices

```
S1 = A + B           % Addition des matrices A et B
S2 = B + C           % Addition des matrices B et C
```

```
>> S1 = A + B
```

```
error: operator +: nonconformant arguments (op1 is 3x2, op2 is 3x3)
```

```
>> S2 = B + C

S2 =

     2     6    10
     6    10    12
    10    16    18
```

3.2 Soustraction des matrices

```
D1 = A - B           % Soustraction de la matrice B de A
D2 = B - C           % Soustraction de la matrice C de B
```

```
>> D1 = A - B

error: operator -: nonconformant arguments (op1 is 3x2, op2 is 3x3)

>> D2 = B - C

D2 =

     0    -2    -4
     2     0     0
     4     0     0
```

3.3 Multiplication d'une matrice par un scalaire

```
AA = 2 * A           % Multiplication de la matrice A par 2
CC = C * 3           % Multiplication de la matrice C par 3
```

```
AA =

     2     4
    10    12
    14    16

CC =

     3    12    21
     6    15    18
     9    24    27
```

3.4 Produit scalaire

```
x = [ 1 5 7 9 ] ;           % Vecteur de taille 4
y = [ 2 7 3 1 ] ;           % Vecteur de taille 4
z = dot( x , y )           % Produit scalaire de x et y
```

```
z = 67
```

3.5 Produit matriciel

```
P1 = B * A                 % Produit matriciel de B et A
P2 = B * C                 % Produit matriciel de B et C
P3 = C * B                 % Produit matriciel de C et B
```

```
P1 =

    32    38
    71    86
   110   134
```

```
P2 =

    14    38    46
    32    89   112
    50   140   178
```

```
P3 =

    66    78    90
    64    77    90
    98   118   138
```

```
P4 = A * B                 % Produit matriciel de A et B
```

```
>> P4 = A * B
```

```
error: operator *: nonconformant arguments (op1 is 3x2, op2 is 3x3)
```

3.6 Puissances d'une matrice carrée

```
B4 = B^4                 % Puissance 4 de la matrice B
C5 = C^5                 % Puissance 5 de la matrice C
```

```
B4 =
```

```
7560    9288    11016
17118   21033   24948
26676   32778   38880
```

```
C5 =
```

```
122976   331256   399424
126438   340581   410666
193576   521428   628728
```

```
A2 = A^2
```

```
% Puissance 2 de la matrice A
```

```
error: for x^A, A must be a square matrix. Use .^ for elementwise power.
```

3.7 Exponentielle d'une matrice carrée

```
EB = expm( B )
```

```
% Exponentielle de la matrice B
```

```
EC = expm( C )
```

```
% Exponentielle de la matrice C
```

```
EB =
```

```
1.1189e+006   1.3748e+006   1.6307e+006
2.5339e+006   3.1134e+006   3.6929e+006
3.9489e+006   4.8520e+006   5.7552e+006
```

```
EC =
```

```
1.1407e+006   3.0728e+006   3.7051e+006
1.1728e+006   3.1593e+006   3.8094e+006
1.7956e+006   4.8368e+006   5.8321e+006
```

```
EA = expm( A )
```

```
% Exponentielle de la matrice A
```

```
>> EA = expm( A )
```

```
error: expm: A must be a square matrix
error: called from
expm at line 82 column 5
```

3.8 Racine carrée d'une matrice carrée

```
SB = sqrtm( B )           % Racine carrée de la matrice B
SC = sqrtm( C )           % Racine carrée de la matrice C
```

```
SB =

    0.44976 + 0.76228i    0.55262 + 0.20680i    0.65549 - 0.34869i
    1.01852 + 0.08415i    1.25147 + 0.02283i    1.48442 - 0.03849i
    1.58729 - 0.59398i    1.95032 - 0.16114i    2.31335 + 0.27170i

SC =

    0.45219 + 0.86258i    1.21804 + 0.50813i    1.46869 - 0.87988i
    0.46492 - 0.13052i    1.25233 + 0.25541i    1.51003 - 0.08390i
    0.71179 - 0.15733i    1.91731 - 0.36826i    2.31185 + 0.34049i
```

```
SA = sqrtm( A )           % Racine carrée de la matrice A
```

```
>> SA = sqrtm( A )

error: sqrtm: A must be a square matrix
```

3.9 Opérations élément-par-élément

3.9.1 Multiplication élément-par-élément

```
MBC = B .* C             % Multiplication élément-par-élément de B et C
```

```
MBC =

     1     8    21
     8    25    36
    21    64    81
```

```
MBA = B .* A             % Multiplication élément-par-élément de B et A
```

```
error: product: nonconformant arguments (op1 is 3x3, op2 is 3x2)
```

3.9.2 Puissances élément-par-élément

```
AA2 = A .^ 2             % Puissances 2 élément-par-élément de A
BB3 = B .^ 3             % Puissances 3 élément-par-élément de B
```

```
AA2 =
```

```
1    4
25   36
49   64
```

```
BB3 =
```

```
1    8    27
64   125  216
343  512  729
```

3.9.3 Division élément-par-élément

```
DBC = B ./ C           % Division élément-par-élément de B par C
```

```
DBC =
```

```
1.00000    0.50000    0.42857
2.00000    1.00000    1.00000
2.33333    1.00000    1.00000
```

```
DBA = B ./ A           % Division élément-par-élément de B par A
```

```
error: quotient: nonconformant arguments (op1 is 3x3, op2 is 3x2)
```


Chapitre 4

Caractéristiques des matrices carrées

```
ccc
```

4.1 Trace

```
ccc
```

```
trB = trace( B )           % Trace de la matrice B
trC = trace( C )           % Trace de la matrice C
```

```
trB = 15
trC = 15
```

```
trA = trace( A )           % Trace de la matrice A
```

```
trA = 7
```

Ce dernier résultat est inattendu ! On ne peut calculer la trace que pour des matrices carrées.

4.2 Déterminant

```
detB = det( B )           % Déterminant de la matrice B
detC = det( C )           % Déterminant de la matrice C
```

```
detB = -9.5171e-016
detC = 4.0000
```

```
detA = det( A )           % Déterminant de la matrice A
```

```
error: det: A must be a square matrix
```

4.3 Inverse

```
invB = inv( B )           % Inverse de la matrice B
invC = inv( C )           % Inverse de la matrice C
```

```
invB =

    3.1522e+015   -6.3044e+015    3.1522e+015
   -6.3044e+015    1.2609e+016   -6.3044e+015
    3.1522e+015   -6.3044e+015    3.1522e+015
```

```
invC =

   -7.5000e-001    5.0000e+000   -2.7500e+000
    1.9429e-016   -3.0000e+000    2.0000e+000
    2.5000e-001    1.0000e+000   -7.5000e-001
```

```
invA = inv( A )           % Inverse de la matrice A
```

```
error: inverse: A must be a square matrix
```

4.4 Valeurs et vecteurs propres

```
[vepB , vapB] = eig( B )           % Valeurs et vecteurs propres de B
[vepC , vapC] = eig( C )           % Valeurs et vecteurs propres de C
```

```
vepB =

   -0.231971   -0.785830    0.408248
   -0.525322   -0.086751   -0.816497
   -0.818673    0.612328    0.408248
```

```
vapB =
```

Diagonal Matrix

```
    1.6117e+001    0                0
    0                -1.1168e+000    0
    0                0                -8.8230e-016
```

```
vepC =
```

```
   -0.469590   -0.972494    0.900502
   -0.482808    0.173103   -0.427873
```

```
-0.739177    0.155856    0.077600
```

```
vapC =
```

```
Diagonal Matrix
```

```
16.13122    0          0
0           -0.83384   0
0           0         -0.29738
```

```
[vepA , vapA] = eig( A )           % Valeurs et vecteurs propres de A
```

```
error: eig: A must be a square matrix
```


Chapitre 5

Systemes linéaires

Chapitre 6

Importer et sauvegarder un fichier de données

6.1 Importer un fichier Excel

6.2 Importer un fichier Texte

Chapitre 7

Graphiques

7.1 Séquence de valeurs linéairement espacées

```
x = linspace( -2 , 2 , 10 )           % Générer 10 valeurs comprises entre -2 et 2
x = -2 : 0.1 : 2                     % Générer des valeurs entre -2 et 2 avec un pas de 0.1
```

```
x =
Columns 1 through 6:
-2      -1.5556   -1.1111   -0.66667   -0.22222   0.22222
Columns 7 through 10:
0.66667   1.1111   1.5556   2
x =
Columns 1 through 3:
-2      -1.9      -1.8
Columns 4 through 6:
-1.7     -1.6     -1.5
...
Columns 40 and 41:
1.9      2
```

7.2 Meshgrid

```
x = linspace( -2 , 2 , 5 ) ;  
y = linspace( -2 , 2 , 5 ) ;  
[ xx , yy ] = meshgrid( x , y )
```

```
xx =  
  
-2 -1 0 1 2  
-2 -1 0 1 2  
-2 -1 0 1 2  
-2 -1 0 1 2  
-2 -1 0 1 2  
  
yy =  
  
-2 -2 -2 -2 -2  
-1 -1 -1 -1 -1  
0 0 0 0 0  
1 1 1 1 1  
2 2 2 2 2
```

7.3 Graphiques 2D

```
y = cos( x );           % Evaluer la cos(x) en chacune des valeurs de x  
plot( y )               % Représenter graphiquement la fonction y = cos(x)  
print -deps figure1.eps % sauvegarder la représentation graphique au format PS
```

7.4 Graphiques 3D

Deuxième partie

Initiation à la programmation avec Octave

Chapitre 8

Opérateurs booléens

Chapitre 9

Boucles

9.1 Boucle « For »

9.2 Boucle « While »

Chapitre 10

Tests logiques

10.1 Test logique simple « If »

10.2 Tests logiques imbriqués « If – Elseif »

Chapitre 11

Scripts et fonctions

11.1 Scripts

11.2 Fonctions

Conclusion

Certificat

Listings

```
#-----  
# clc      : effacer la console  
# clear x  : supprimer la variable "x"  
# clear    : supprimer toutes les variables  
# help doc xx : afficher l'aide sur xx  
# ;        : empêcher l'affichage du résultat  
# Ctrl + c : arrêter l'exécution  
# # ou %   : mettre une ligne en commentaire  
# #{...#} ou : mettre un bloc de lignes en commentaire  
# Format short g  
#-----
```

Listing 11.1 – Commandes utiles

```
#-----  
#                               NOMBRES REELS  
#-----  
# Initialisation des variables  
# Opérations arithmétiques sur les nombres réels  
# Puissances, racines et valeur absolue d'un nombre réel  
# Exponentielle et logarithmes  
# Trigonométrie  
# Constantes particulières  
#-----  
  
% Initialisation des variables  
a = -1                % Initialisation de la variable "a"  
b = 3 ;              % Initialisation de la variable "b"  
  
% Opérations arithmétiques sur les nombres  
s = a + b            % Somme des nombres a et b  
d = a - b            % Soustraction du nombre b du nombre a
```

```

m = a * b          % Multiplication du nombre a par le nombre b
r = a / b          % Division du nombre a par le nombre b
t = a / 0          % Division du nombre a par 0

% Puissances , racines et valeur absolue d'un nombre réel
a2 = a^2           % Puissance 2 du nombre a
a3 = a^3           % Puissance 3 du nombre a
ab = a^b           % Puissance b du nombre a
rcb = sqrt( b )    % Racine carrée du nombre b
rqb = b^(1/3)      % Racine cubique du nombre b
rca = sqrt( a )    % Racine carrée du nombre a
rqa = a^(1/3)      % Racine cubique du nombre a
va = abs( a )      % Valeur absolue du nombre a

% Exponentielle et logarithmes
ea = e^( a )       % Exponentielle du nombre a
ea = exp( a )      % Exponentielle du nombre a
logb = log( b )    % Logarithme naturel du nombre b
log2b = log2( b )  % Logarithme base 2 du nombre b
log10b = log10( b ) % Logarithme base 10 du nombre b
loga = log( a )    % Logarithme naturel du nombre a
log2a = log2( a )  % Logarithme base 2 du nombre a
log10a = log10( a ) % Logarithme base 10 du nombre a

% Trigonométrie
sinb = sin( b )    % Sinus du nombre b ( b en radian )
cosb = cos( b )    % Consinus du nombre b ( b en radian )
tanb = tan( b )    % Tangente du nombre b ( b en radian )
cotb = cot( b )    % Cotangente du nombre b ( b en radian )

% Constantes particulières
p = pi             % Constante d'Archimède Pi
q = e              % Constante de Neper e

```

Listing 11.2 – Nombres réels

```

#=====
#                               MATRICES ET VECTEURS
#=====
# Initialisation d'une matrice et d'un vecteur
# Matrices nulles , matrices de 1 et matrices identité
# Matrices diagonales et matrices scalaires
# Matrices triangulaires
# Matrices de valeurs manquantes
# Matrices de valeurs non numériques
# Taille d'une matrice
# Atteindre et modifier les éléments d'une matrice

```

```

# Sous-matrice
# Transposée d'une matrice
# Opposée d'une matrice
#-----

% Initialisation d'une matrice et d'un vecteur
A = [ 1 2 3
      4 5 6
      7 8 9 ] % Initialisation de la matrice A
A = [ 1 , 2 , 3 ; 4 , 5 , 6 ; 7 , 8 , 9 ] % Initialisation de la matrice A
L = [ 1 2 3 ] % Initialisation du vecteur ligne L
L = [ 1 , 2 , 3 ] % Initialisation du vecteur ligne L
C = [ 1
      2
      3 ] % Initialisation du vecteur colonne C
C = [ 1 ;
      2 ;
      3 ] % Initialisation du vecteur colonne C
C = [ 1 ; 2 ; 3 ] % Initialisation du vecteur colonne C

% Matrices nulles , matrices de 1 et matrices identité
O = zeros( 2 , 3 ) % Matrice nulle de taille 2 x 3
U = ones( 4 , 3 ) % Matrice de 1 de taille 4 x 3
I = eye( 4 ) % Matrice identité d'ordre 4

% Matrices diagonales et matrices scalaires
D = diag( [1 , 2 , 3 , 6 , 7 , 8 ] ) % Matrice diagonale
S = diag( [3 , 3 , 3 , 3 , 3 , 3 ] ) % Matrice scalaire

% Matrices triangulaires
TIE = tril( A ) % Matrice triangulaire inférieure
TSI = tril( A , -1 ) % Mat. tri strictement inférieure
TSE = triu( A ) % Matrice triangulaire supérieure
TSS = triu( A , 1 ) % Mat. tri strictement supérieure

% Matrices de valeurs manquantes
V = NA( 5 , 6 ) % NA: Not Available

% Matrices de valeurs non numériques
W = NaN( 4 , 6 ) % NaN: Not a Number

% Taille d'une matrice
[nl , nc] = size( A ) % Nombre de lignes et de colonnes

% Longueur d'un vecteur
ll = length( L ) % Longueur du vecteur ligne L
lc = length( C ) % Longueur du vecteur colonne C

```



```

% Produit scalaire
x = [ 1 5 7 9 ] ;           % Vecteur de taille 4
y = [ 2 7 3 1 ] ;           % Vecteur de taille 4
z = dot( x , y )           % Produit scalaire de x et y

% Produit matriciel
P1 = B * A                 % Produit matriciel de B et A
P2 = B * C                 % Produit matriciel de B et C
P3 = C * B                 % Produit matriciel de C et B
P4 = A * B                 % Produit matriciel de A et B

% Puissances d'une matrice carrée
B4 = B^4                   % Puissance 4 de la matrice B
C5 = C^5                   % Puissance 5 de la matrice C
A2 = A^2                   % Puissance 2 de la matrice A

% Exponentielle d'une matrice carrée
EB = expm( B )             % Exponentielle de la matrice B
EC = expm( C )             % Exponentielle de la matrice C
EA = expm( A )             % Exponentielle de la matrice A

% Racine carrée d'une matrice
SB = sqrtm( B )           % Racine carrée de la matrice B
SC = sqrtm( C )           % Racine carrée de la matrice C
SA = sqrtm( A )           % Racine carrée de la matrice A

% Opérations élément-par-élément
MBC = B .* C              % Multiplication élément-par-élément de B et C
MBA = B .* A              % Multiplication élément-par-élément de B et A
AA2 = A .^ 2              % Puissances 2 élément-par-élément de A
BB3 = B .^ 3              % Puissances 3 élément-par-élément de B
DBC = B ./ C              % Division élément-par-élément de B par C
DBA = B ./ A              % Division élément-par-élément de B par A

```

Listing 11.4 – Opérations sur les matrices

```

#-----
#                               CARACTERISTIQUES DES MATRICES CARREES
#-----
# Trace
# Déterminant
# Inverse
# Valeurs et vecteurs propres
#-----

A = [ 1 2 ; 5 6 ; 7 8 ]           % Matrice 3 x 2
B = [ 1 , 2 , 3 ; 4 , 5 , 6 ; 7 , 8 , 9 ] % Matrice 3 x 3

```

```

C = [ 1 , 4 , 7 ; 2 , 5 , 6 ; 3 , 8 , 9 ] % Matrice 3 x 3

% Trace
trB = trace( B )           % Trace de la matrice B
trC = trace( C )           % Trace de la matrice C
trA = trace( A )           % Trace de la matrice A

% Déterminant
detB = det( B )            % Déterminant de la matrice B
detC = det( C )            % Déterminant de la matrice C
detA = det( A )            % Déterminant de la matrice A

% Inverse
invB = inv( B )            % Inverse de la matrice B
invC = inv( C )            % Inverse de la matrice C
invA = inv( A )            % Inverse de la matrice A

% Valeurs et vecteurs propres
[vepB , vapB] = eig( B )    % Valeurs et vecteurs propres de B
[vepC , vapC] = eig( C )    % Valeurs et vecteurs propres de C
[vepA , vapA] = eig( A )    % Valeurs et vecteurs propres de A

```

Listing 11.5 – Caractéristiques des matrices carrées

```

#=====
#                               SYSTEMES LINEAIRES
#=====
# Trace
# Déterminant
# Inverse
# Valeurs et vecteurs propres
#-----
https://www.youtube.com/watch?v=twGcAlac3ds

```

Listing 11.6 – Systèmes linéaires

```

#=====
#                               IMPORTER ET EXPORTER DES FICHIERS DE DONNEES
#=====
# Importer des fichiers Excel
# Importer des fichiers Text
# Importer des fichiers CSV
# Exporter vers des fichiers Excel
# Exporter vers des fichiers Text
# Exporter vers des fichiers CSV
#-----
% Importer des fichiers Excel

```

```
[numarr , txtarr , rawarr , limits] = xlsread( ImporTest.xls )

% Importer des fichiers Text

% Importer des fichiers CSV

% Exporter vers des fichiers Excel

x = rand( 30 , 5 )
status = xlswrite('test4.xls');

% Exporter vers des fichiers Text

% Exporter vers des fichiers CSV
```

Listing 11.7 – Importer des fichiers de données

```
##
#
# GRAPHIQUES
#
# Séquences linéairement espacées
# Graphiques 2D
# Graphiques 3D
#
% Séquences linéairement espacées
x = linspace( -2 , 2 , 5 ) % Générer 5 valeurs comprises entre -2 et 2
y = -2 : 1 : 2 % Générer des valeurs entre -2 et 2 avec un pas de 1

% Graphiques 2D
plot( x , cos( x ) ) % Représenter graphiquement la fonction cos(x)
%print -deps -color figure1.eps % sauvegarder la représentation graphique au format
PS
%print -dpng -color file.png
print -deps -color file.eps

% Graphiques 3D

x = linspace( -2 , 2 , 5 ) ;
y = linspace( -2 , 2 , 5 ) ;
[ xx , yy ] = meshgrid( x , y )
%meshc( xx , yy , 4 - ( xx.^2 + yy.^2 ) )
mesh( xx , yy , ( xx.^2 + yy.^2 ) - 4 )

mesh( xx , yy , sin( 2 .* ( xx.^2 + yy.^2 ).^(1/2) ) )

tx = ty = linspace( -8 , 8 , 80 )' ;
```

```
[ xx , yy ] = meshgrid( tx , ty ) ;
r = sqrt( xx.^2 + yy.^2 ) + eps ;
tz = sin( r ) ./ r ;
mesh( tx , ty , tz )
xlabel( "tx" )
ylabel( "ty" )
zlabel( "tz" )
%title( "3-D Sombrero plot" )
grid off
axis off
print -dpng -color sombrero.png
```

Listing 11.8 – Graphiques

```
#=====
#                               BOUCLES
#=====
# Graphiques 2D
# Graphiques 3D
#-----

% Graphiques 2D

% Graphiques 3D

https://www.youtube.com/watch?v=irGGAALK3kE
```

Listing 11.9 – Boucles

```
#=====
#                               TESTS LOGIQUES
#=====
# Graphiques 2D
# Graphiques 3D
#-----

% Graphiques 2D

% Graphiques 3D

x = linspace( -2 , 2 , 100 ) ;
y = linspace( -2 , 2 , 100 ) ;
[ xx , yy ] = meshgrid( x , y ) ;
%meshc( xx , yy , 4 - ( xx.^2 + yy.^2 ) )
mesh( xx , yy , ( xx.^2 + yy.^2 ) - 4 )

mesh( xx , yy , sin( 2 .* ( xx.^2 + yy.^2 ).^(1/2) ) )
```

```
tx = ty = linspace( -8 , 8 , 80 )' ;  
[ xx , yy ] = meshgrid( tx , ty ) ;  
r = sqrt( xx.^2 + yy.^2) + eps ;  
tz = sin( r ) ./ r ;  
mesh( tx , ty , tz )  
xlabel( "tx" )  
ylabel( "ty" )  
zlabel( "tz" )  
title( "3-D Sombrero plot" )  
grid on  
axis off
```

Listing 11.10 – Tests logiques

```
#####  
#                               SCRIPTS ET FONCTIONS                               #  
#####  
# Graphiques 2D  
# Graphiques 3D  
#  
  
% Graphiques 2D  
  
% Graphiques 3D
```

Listing 11.11 – Scripts et fonctions