

Table des matières

1	Introduction à Matlab	5
1	Introduction	5
1.1	Présentation de Matlab	6
1.2	Saisie de commandes sous matlab	6
1.3	Les caractères spéciaux notables sont les suivants :	8
1.4	Opérateurs mathématiques	8
1.5	Opérateurs relationnels	8
1.6	Opérateurs logiques	9
1.7	Fonctions mathématiques	9
1.8	Travaux pratiques	10
1.9	Calcul sur les nombres complexes	11
1.10	Affichage d'un nombre complexe	12
1.11	Forme trigonométrique	12
1.12	Forme exponentielle	12
1.13	Fonctions à variables complexes	12
2	Exercices	14
3	Systemes d'équations linéaires	14
3.1	Determinants	14
3.2	Résolution d'un système d'équations linéaires	15
3.3	Exercice	16
3.4	Exemple de résolution d'équations linéaires	17
2	Les vecteurs et les matrices	19
0.5	les Vecteurs ou tableau à une dimension	19
0.6	Fonctions propres aux vecteurs	21
0.7	Opérations éléments par éléments	23
1	Calcul sur les polynômes	23
1.1	Racines d'un polynôme	24
1.2	Détermination des coefficients d'un polynôme	24
1.3	Produit de polynômes	25
1.4	Fractions rationnelles : Décomposition en éléments simples	25
2	Matrices	26
2.1	Créer des Matrices	26
2.2	Créer des Matrices à partir des fonctions	27
2.3	Opérations sur les matrices	28
2.4	matrice inverse	28
2.5	Construction de matrices à partir de vecteurs	30
2.6	Fonctions usuelles	32
2.7	Matrices	33

3	équations linéaires	35
0.8	Résolution d'un système d'équations linéaires	35
0.9	Inversion de matrice	35
0.10	Exemple de résolution d'équations linéaires	36
0.11	Règle de Cramer	36
0.12	Exercices	37
4	Travaux Pratiques	41
1	Exercices avec solutions	41

Le présent document a pour but de fournir les bases nécessaires à l'utilisation du logiciel MATLAB, dans le cadre des cours, des exercices et du laboratoire. L'objet de ce fascicule est d'initier le lecteur aux concepts de base de MATLAB pour lui permettre d'acquérir rapidement une autonomie d'utilisation du logiciel. Ce document ne se veut pas exhaustif mais propose simplement une introduction à MATLAB. Pour de plus amples informations sur certains points précis il sera nécessaire de se reporter aux guides MATLAB.

A.Amirou

Chapitre 1

Introduction à Matlab

Sommaire

A. Amirou

1	Introduction	5
1.1	Présentation de Matlab	6
1.2	Saisie de commandes sous matlab	6
1.3	Les caractères spéciaux notables sont les suivants :	8
1.4	Opérateurs mathématiques	8
1.5	Opérateurs relationnels	8
1.6	Opérateurs logiques	9
1.7	Fonctions mathématiques	9
1.8	Travaux pratiques	10
1.9	Calcul sur les nombres complexes	11
1.10	Affichage d'un nombre complexe	12
1.11	Forme trigonométrique	12
1.12	Forme exponentielle	12
1.13	Fonctions à variables complexes	12
2	Exercices	14
3	Systemes d'équations linéaires	14
3.1	Determinants	14
3.2	Résolution d'un système d'équations linéaires	15
3.3	Exercice	16
3.4	Exemple de résolution d'équations linéaires	17

1 Introduction

Matlab signifie Matrix laboratory. C'est un logiciel de calcul numérique, dont les objets privilégiés sont les matrices (réelles, complexes), les vecteurs (réels, complexes) et les scalaires. La simplicité d'utilisation de MATLAB en fait un outil pédagogique indispensable à la compréhension de cours.

Tous les objets définis dans MATLAB le sont donc au moyen de vecteurs et de matrices/tableaux de nombres. Un ensemble important d'opérateurs et de fonctions MATLAB de base facilitent leur manipulation et des opérations comme par exemple le produit et l'inversion matricielles (`inv`), la transposition (`'`) ou encore le calcul des valeurs propres (`eig`) font partie de la bibliothèque standard. D'autres fonctions servant à la création et à la manipulation de matrices et de tableaux (`diag`, `rand`, `ones`, `zeros`, `linspace`) sont également disponibles en nombre. L'environnement MATLAB se présente sous la forme d'un espace de travail (Workspace), où un interpréteur de commandes exécute des opérations et fonctions MATLAB.

L'interface graphique de MATLAB est sans conteste l'un des points forts du logiciel et facilite le tracé de courbes et l'obtention de graphiques 2D ou 3D de grande qualité (plot, stairs, stem, hist, mesh, surf, plot3).

Le "langage" MATLAB contient un minimum de structures de programmation (structure itérative, structure conditionnelle, sous-routine) mais reste très rudimentaire. L'avantage est qu'il est très simple et très rapide à programmer, offrant une grande tolérance.

Ce système est doté d'un interpréteur de commandes dont la syntaxe est hybride du C et du PASCAL. Ceci permet de taper des commandes les unes à la suite des autres, puis de les exécuter par return. Il est également possible de créer des suites d'instructions dans un fichier et d'exécuter ces instructions en bloc en tapant le nom du fichier au niveau de la ligne de commande. Au logiciel de base s'ajoutent, selon la configuration choisie, les fonctions provenant d'une série de boîtes à outils (toolbox) dédiés à des domaines techniques spécifiques, comme :

- le traitement de signal (signal processing toolbox),
- la régulation automatique (control system toolbox),
- l'identification (system identification toolbox),
- les réseaux de neurones (neural networks toolbox),
- la logique floue (fuzzy logic toolbox),
- le calcul symbolique (symbolic math toolbox), et bien d'autres encore.

1.1 Présentation de Matlab

Matlab (Matrix Laboratory) est un logiciel de calcul matriciel à syntaxe 'simple' (relativement à des langages évolués comme C, C++). MATLAB est un interpréteur de commandes : les instructions sont interprétées et exécutées ligne par ligne (pas de compilation avant de les exécuter). Il existe deux modes de fonctionnement :

- mode interactif : MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'utilisateur.
- mode exécutif : MATLAB exécute ligne par ligne un fichier ".m" (programme en langage MATLAB).

1.2 Saisie de commandes sous matlab

Pour commencer cliquer sur l'icône associé à Matlab, afin de charger les fenêtres suivantes :

- fenêtre des commandes (Command Window),
- historique des commandes (Command History),
- répertoire courant (Current Directory),
- espace de travail (Workspace),

Seule la première nous intéresse pour l'instant. Pour que cette fenêtre soit la seule visible, il suffit de sélectionner Command Window dans le menu View. A ce stade le prompt (curseur clignotant) :

```
>> _
```

est visible dans la fenêtre des commandes. Il indique que Matlab attend une commande.

Dans un premier temps, on peut se contenter d'introduire ses commandes une à une au niveau de l'espace de travail où elles sont interprétées directement. Cependant, par la suite, il est beaucoup plus pratique d'écrire sa séquence de commandes complète au moyen d'un éditeur, puis de sauver le tout dans un fichier avec l'extension .M. Cette séquence pourra alors être exécutée dans MATLAB par simple introduction du nom du fichier.

Quitter Pour quitter Matlab, sélectionner Exit MATLAB dans le menu File ou bien saisir la commande exit dans la fenêtre des commandes. On peut aussi utiliser la commande quit, elle permet d'exécuter un certain nombre de tâches avant de quitter.

Aide en ligne En plus de l'aide de Windows, une aide en ligne est disponible pour chaque commande de MATLAB. Il suffit d'introduire :

```
>> help nom de commande
```

```

>> who      Affiche les variables actuellement présentes en mémoire.
>> whos     Affiche les variables actuellement présentes en mémoire ainsi qu'une série d'informations comme leur nature, leur taille en nombre de lignes et de colonnes pour les matrices, leur taille en bits, etc.
>> exist ('NomVariable')  Affiche la valeur 1 si la variable NomVariable existe, et la valeur 0 si la variable n'existe pas.
>> clear NomVariable      Efface de la mémoire la variable NomVariable.
>> clear                  Efface toutes les variables en mémoire.
>> help NomFonction       Affiche la description et la syntaxe de la fonction NomFonction.

```

format short Par défaut Matlab affiche les nombres réels avec 4 chiffres après la virgule, il existe d'autres formats d'affichage.

```
format short (format par défaut) »format short
```

```

»1/7
ans=
0.1429
» 2*pi
ans
= 6.2832

```

format long Les nombres réels sont codés en double précision avec 53 chiffres significatifs. Ces chiffres significatifs en base 2 correspondent aux 15 chiffres significatifs montrés par Matlab en base 10 avec le format " format long ". format long e affiche 16 chiffres :

```

»format long
»1/7
ans=
0.142857142857143
» format long e
» 2*pi
ans
= 6.283185307179586e+000
format rationnel
»format rat
»pi
ans=
355
113

```

Erreur relative L'erreur relative commise par Matlab ou l'épsilon machine est donc de $2^{1-53} = 2^{52}$. On peut retrouver cette erreur relative en tapant dans le work space »eps

```

ans=
2.2204e-016

```

Le plus grand et le plus petit réel positifs Le plus grand et le plus petit réel positifs en Matlab peuvent être obtenus en tapant »realmax

```

ans=
1.7977e+308
»realmin
ans=
2.2251e-308

```

Il existe également quelques valeurs particulières :

ans	Variable créée automatiquement quand le résultat d'une expression n'est pas affecté à une variable.
pi	Constante correspondant à π
inf	$+\infty$
-inf	$-\infty$
NaN	(Not-a-Number)

1.3 Les caractères spéciaux notables sont les suivants :

opérateur,

- permet de séparer deux instructions sur une même ligne
- termine une instruction avec affichage
- permet de séparer des éléments d'un vecteur ou d'une matrice

opérateur ;

- Utilisé à la suite d'une commande, ce caractère supprime l'écho à l'écran de cette commande et de son résultat.
- permet de séparer deux lignes d'une matrice
- séparateurs d'instructions

opérateur ...

Ajouté au cours d'une commande, ce caractère permet la continuation de la commande à la ligne suivante.

opérateur :

permet la création d'une suite ordonnée de valeurs. La syntaxe est la suivante :

```
>> i = [debut : pas : fin];
```

Elle génère un vecteur i dont la première composante est debut, la dernière fin et le pas entre deux composantes successives est pas. Si pas est omis il est pris par défaut égal à 1.

Cet opérateur est très important, on l'utilise très souvent. Il permet de discrétiser un intervalle avec un pas constant.

Exemple : L'instruction

```
» u = 0 :9
```

crée un vecteur u contenant les entiers de 0 à 9. Par défaut le pas est égal à 1, mais on peut spécifier une autre longueur de pas

```
» 0 :2 :9, 9 :-3 :0, 0 :pi/3 :2*pi
```

1.4 Opérateurs mathématiques

- ✓ + addition
- ✓ - soustraction
- ✓ * multiplication
- ✓ / division
- ✓ ^ puissance

1.5 Opérateurs relationnels

Les opérateurs relationnels sont :

Les opérateurs relationnels sont

- ✓ == égal
- ✓ ~= différent
- ✓ > strictement plus grand
- ✓ >= plus grand ou égal
- ✓ < strictement plus petit

✓ `<=` plus petit ou égal.

Les deux opérandes doivent être de même dimension, sauf si un des deux opérandes est un scalaire. Le résultat est une matrice de même dimension n'ayant que des 0 (faux) et des 1 (vrai), la comparaison étant effectuée élément par élément. Lorsqu'un des deux opérandes est un scalaire, la comparaison est effectuée entre chaque élément de la matrice et le scalaire.

» `R = randn(1,6), R>0`

```
R =
    1.1892   -0.0376    0.3273    0.1746   -0.1867    0.7258
ans =
     1         0         1         1         0         1
```

Exemple

```
>> R = randn(1,6), R > 0
R = 1.1892   -0.0376    0.3273    0.1746   -0.1867    0.7258
ans =     1         0         1         1         0         1
```

1.6 Opérateurs logiques

Les opérateurs logiques sont

✓ `&` *et*
 ✓ `||` *ou*
 ✓ `~` *non*

Calculons une table logique :

```
>> a = [0; 0; 1; 1]; b = [0; 1; 0; 1];
```

```
Et Ou ab = [a, b, a&b, a||b, (~a)|b]
```

```
    0  0  0  0  1
    0  1  0  1  1
    1  0  0  1  0
    1  1  1  1  1
```

Une valeur numérique est considérée comme vraie (= 1) si elle est

non nulle, sinon elle est fausse (= 0). Par exemple, `~ A` renvoie une matrice dont les éléments valent 1 là où A a des éléments nuls et 0 ailleurs.

1.7 Fonctions mathématiques

<code>sin(X)</code>	sinus
<code>asin(X)</code>	sinus inverse
<code>cos(X)</code>	cosinus
<code>acos(X)</code>	cosinus inverse
<code>tan(X)</code>	tangente
<code>atan(X)</code>	tangente inverse
<code>exp(X)</code>	exponentielle
<code>log(X)</code>	logarithme naturel (base e)
<code>log10(X)</code>	logarithme décimal (base 10)
<code>sqrt(X)</code>	racine carrée
<code>abs(X)</code>	valeur absolue

avec X : argument en radians

Quelques fonctions usuelles Ce sont les fonctions mathématiques usuelles.

```
sinh cosh tanh acosh asinh atanh
sign rem sign rem round floor ceil
```

L'argument d'entrée peut être un scalaire, un vecteur ou une matrice. L'argument de sortie est une matrice de même dimension, la fonction étant appliquée sur chacun de ses éléments.

```
>> log(0 : 5)
```

Warning : Log of zero.

```
ans = -Inf 0 0.6931 1.0986 1.3863 1.6094
```

– Pour définir un scalaire :

```
» a = 2
```

définit la variable $a = 2$ et affiche un scalaire $a = 2$

– Pour définir un nombre complexe :

```
» z = 1+2*i
```

définit le nombre complexe $z = 1 + 2i$ et affiche $z = 1.0000 + 2.0000i$

Pour connaître la valeur de a taper

```
» a
```

en actionnant la touche RETURN, on visualise la valeur de a .

Exemple : Utilisation de variables

```
» 5*3
```

```
ans = 15
```

```
» ans+4
```

```
ans = 19
```

```
» a = 2 + log(15)
```

```
a = 4.7081
```

```
» b = - 45
```

```
b = -45
```

```
» a * b
```

```
ans = -211.8623
```

```
» c = a - sqrt(abs(b))
```

```
c = -2.0002
```

1.8 Travaux pratiques

Matlab peut être vu comme une calculatrice extrêmement puissante. Les opérations simples peuvent être tapées directement, et l'on obtient le résultat en appuyant sur la touche "Entrée".

1. . Essayez de faire quelques opérations dans l'interpréteur :

```
>> 10 + 5
```

```
>> 10 * 5
```

```
>> 10^5
```

```
>> 5^2.5
```

```
>> 5^-3
```

```
>> 6/2
```

```
>> 3\6
```

```
>> 3 * (323 + 14.5 - 4/6)/3.5
```

```
>> a = 2
```

```
>> A = 3
```

```
>> 2 * a
```

Nombre complexe

```
»z1=3 + 4i
```

```
z1=
```

```
3.0000+4.0000i
```

```
»z2=4 + 3i
```

```
z2=
```

```
4.0000+3.0000i
```

Le nombre conjugué s'obtient par :

```
»z1_bar=conj(z1)
```

```
z1_bar=
```

```
3.0000-4.0000i
```

Format d'affichage >> *pi*

```
»format long
```

```
>> pi
»format rat
>> pi
»format compact
»x=4; y=x^2+2*x+53
y=
77
»format short
>> pi
Afin d'affecter une variable, on utilise le signe =. Ainsi la ligne :
>> var1 = 3
```

2. Familiarisez-vous avec les variables :

```
>> var1 = 52
>> var1
>> var2 = 32;
>> var2
>> var1 * var2
>> Var1 = 12
>> var1
>> 11 = var1
```

3. A quoi sert le point virgule à la fin d'une ligne de commande ? Que remarque-t'on à propos de la gestion des majuscules/minuscules dans les noms de variables ?

Réponse : Le point virgule sert à ne pas afficher dans l'interpréteur la valeur calculée/affectée.

4. Pour effacer une variable, on peut se servir de la command clear varname, si on ne donne pas d'argument à clear, alors toutes les variables sont effacées. Essayez par vous-même :

```
>> clear var1
>> var1
>> clear
```

5. Il existe une floppée de fonctions mathématiques comme par exemple sin, log, sqrt et bien d'autres encore. Par défaut, il y a également des variables qui sont prédéfinies comme pi ou i (nombre imaginaire pur) par exemple.

Testez les fonctions : sin(pi/2),sqrt,...

Vous ne connaîtrez probablement jamais toutes les commandes de Matlab , mais ce n'est pas un problème, car vous pourrez retrouver toutes les informations nécessaires facilement en vous servant de l'aide. Si vous vous rappelez d'une commande mais pas de comment on l'utilise, alors la commande **help commande** vous sera utile. >> help log

```
>> help mod
```

1.9 Calcul sur les nombres complexes

i	imaginaire pur
j	imaginaire pur
conj(X)	conjugué du nombre complexe X
real(X)	partie réelle
imag(X)	partie imaginaire
abs(X)	Module
angle(X)	argument (en radians)

Exemple

```
» i^2
```

```

ans=
-1
et
» j^2
ans=
-1
Nombre complexe
»z1=3 + 4i
z1=
3.0000+4.0000i

```

1.10 Affichage d'un nombre complexe

```

La partie réelle : »a=real(z1)
a=
3
La partie imaginaire : »b=image(z1)
b=
4
et son argument :
»theta=atan(b/a)
theta=
0.9273
»theta=angle(z1)
theta=
0.9273
le module :
»rho=abs(z1)
rho=
5

```

1.11 Forme trigonométrique

```

»rho*(cos(theta) + i*sin(theta))
ans=
3.0000+4.0000i

```

1.12 Forme exponentielle

```

»rho*exp(i*theta)
ans=
3.0000+4.0000i

```

1.13 Fonctions à variables complexes

```

»sin(z1)
ans=
3.8537 - 27.0168i
»cos(z1)
ans=

```

```
-27.0349 - 3.8512i
»sin(z1)^2 + cos(z1)^2
ans=
1
»z2=4 + 3i
z2=
4.0000+3.0000i
addition : »z1+z2
```

```
ans=
7.0000+7.0000i
multiplication
```

```
»z1*z2
```

```
ans=
0+25.0000i
```

```
division
```

```
»z1/z2
```

```
ans=
0.9600+0.2800i
```

```
puissance
```

```
»z1^2
```

```
ans=
-7.0000+24.0000i
```

Le nombre conjugué s'obtient par :

```
»z1_bar=conj(z1)
```

```
z1_bar=
```

```
3.0000-4.0000i
```

Le module d'un nombre complexe peut se calculer comme la racine carré du produit du nombre complexe par son conjugué »rho=sqrt(z1*z1_bar)

```
rho=
```

```
5
```

le module s'obtient directement

```
»rho=abs(z1)
```

```
rho=
```

```
5
```

Exemples :

```
» (4 - 2.5i)*(-2 + i)/(1 + i)
```

```
ans = 1.7500 + 7.2500i
```

```
» a = 1 + i
```

```
a = 1.0000 + 1.0000i
```

```
» b = -2 + 3.5j
```

```
b = -2.0000 + 3.5000i
```

```
» a + b
```

```
ans = -1.0000 + 4.5000i
```

```
» a * b
```

```
ans = -5.5000 + 1.5000i
```

```
» a / b
```

```
ans = 0.0923 - 0.3385i
```

```
» conj(a)
```

```
ans = 1.0000 - 1.0000i
```

```
» a * conj(a)
```

```
ans = 2
```

```
» real(a)
```

```
ans = 1
```

```
» imag(conj(a))
```

```

ans = -1
» abs(a)
ans = 1.4142
» angle(a)
ans = 0.7854 sqrt : fonction racine carrée
» c = 2 - sqrt(3)*i
c = 2.0000 - 1.7321i
» abs(c)
ans = 2.6458
» angle(c)
ans = -0.7137 Argument en degrés :
» angle(c)*180/pi
ans = -40.8934

```

A.Amirou

2 Exercices

Considérons le nombre complexe suivant : $z_1 = 1 + 2i$ et $z_2 = 3 - 5j$
Donner le code Matlab qui effectue les opérations arithmétiques et affiche le résultat de :

1. l'addition de z_1 et z_2
2. la division de z_1 sur z_2
3. la soustraction z_1 sur z_2
4. et la multiplication de z_1 par z_2
5. Calcule de z_2 à la puissance 2

Considérons le nombre complexe suivant : $z = [1+j \ 2-3j; \ 4+2j \ 5-4j]$
Donner le code Matlab et afficher le résultat qui calcule :

1. la partie réelle de z
2. la partie imaginaire de z
3. le conjugué de z
4. et le module de z
5. produit de z par son conjugué
6. produit scalaire de z par son conjugué

3 Systemes d'équations linéaires

3.1 Determinants

Considérons le système d'équations linéaires suivant

$$x_1 + 2x_2 + 3x_3 = 4$$

$$2x_1 + 3x_2 + 4x_3 = 5$$

$$4x_1 + 2x_2 + 5x_3 = 1$$

Si nous définissons la **matrice de coefficients A**

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 2 & 5 \end{bmatrix}$$

et le vecteur **B**

$$B = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix}$$

nous pouvons écrire le système sous forme matricielle $A X = B$

$$\gg A = [1, 2, 3; 2, 3, 4; 4, 2, 5]$$

$$\gg B = [4; 5; 1]$$

Calcul du déterminant

$$\gg \det(A)$$

$$\text{ans} = -5$$

Règle de Cramer

D1 s'obtient à partir de A, en attribuant le vecteur B à la première colonne

$$\gg D1 = A; D1(:,1) = B$$

$$D1 = \begin{pmatrix} 4 & 2 & 3 \\ 5 & 3 & 4 \\ 1 & 2 & 5 \end{pmatrix}$$

$$\gg D2 = A; D2(:,2) = B$$

$$D2 = \begin{pmatrix} 1 & 4 & 3 \\ 2 & 5 & 4 \\ 4 & 1 & 5 \end{pmatrix}$$

$$\gg D3 = A; D3(:,3) = B$$

$$D3 = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 3 & 5 \\ 4 & 2 & 1 \end{pmatrix}$$

La solution par la règle de Cramer donne

$$\gg X = [\det(D1); \det(D2); \det(D3)] / \det(A)$$

$$X = \begin{pmatrix} -1.40000 \\ 1.80000 \\ 0.60000 \end{pmatrix}$$

AX

$$\gg A * X$$

$$\text{ans} = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix}$$

On retrouve le vecteur B

A.Amirou

3.2 Résolution d'un système d'équations linéaires

nous proposons le système suivant, comme exemple de résolution linéaire

$$\begin{aligned} 2x_1 + 3x_2 &= 8 \\ x_1 - 2x_2 &= -3 \end{aligned}$$

ce système peut être mis sous forme matricielle $A X = B$

fichier systlin.m

solution $x = \text{inv}(A) * B$ ou $X = A \setminus B$

$$A = [2 \ 3; 1 \ -2];$$

$$B = [8 \ -3]';$$

$$x = \text{inv}(A) * B;$$

disp('solutions :')

disp(['x1 =', num2str(x(1)), ' et x2 =', num2str(x(2))])

\gg systlin

solutions :

$$x_1 = 1 \text{ et } x_2 = 2$$

3.3 Exercice

Soit à résoudre un système de 3 équations à 3 inconnues x_1 , x_2 et x_3 :

$$3x_1 + 2x_2 + x_3 = 4 - x_1 + 5x_2 + 2x_3 = -14x_1 - 2x_2 + 3x_3 = 3$$

On saisit les différents coefficients dans une matrice 3x3

```
»A=[3 2 1; -1 5 2; 4 -2 3]
```

On complete avec un vecteur colonne 3x1

```
»B=[4;-1;3]
```

La solution est donné sous la forme d'un vecteur colonne

```
»A\B
```

```
ans=
```

```
1.3279
```

```
0.2951
```

```
-0.5738
```

Autre méthode

```
»inv(A)*B
```

```
ans=
```

```
1.3279
```

```
0.2951
```

```
-0.5738
```

```
»X(1)
```

```
ans =
```

```
1.3279
```

```
»X(2)
```

```
ans =
```

```
0.2951
```

```
»X(3)
```

```
ans =
```

```
-0.5738
```

Verification

```
»3X(1)+2X(2)+X(3)
```

```
ans=
```

```
4
```

```
»-X(1)+5X(2)+2X(3)
```

```
ans=
```

```
-1
```

```
»4X(1)-2X(2)+3X(3)
```

```
=
```

```
3
```

A.Amirou

Inversion de matrice La matrice A est une matrice carrée de déterminant différent de zero.

Une telle matrice a une matrice inverse qui se calcule par

```
»A_inv=inv(A)
```

```
-1.4000 0.8000 0.2000
```

```
A_inv= -1.2000 1.4000 -0.4000
```

```
1.6000 -1.2000 0.2000
```

Dans ce cas, il est possible de résoudre le système d'équation en multipliant à gauche, le vecteur B par la matrice inverse de A. »X=A_inv*B

```
-1.40000
```

```
X= 1.80000
```

```
0.6000
```


3.4 Exemple de résolution d'équations linéaires

Nous pouvons résoudre à l'aide de l'opérateur " \ "

» A = [2 -1 ; 1 1]

» B = [2 5]

» X = A \ B

X =

2.3333

2.6667

Le rang de la matrice

» rank(A), rank([A B])

ans =

2

ans =

2

Exercice Soit le système d'équations paramétriques :

$$-x_1 + 2x_2 + x_3 = b_1$$

$$-x_1 + 2x_2 + 2x_3 = b_2$$

$$x_1 - 2x_2 + x_3 = b_3$$

On cherche à

exprimer x_1, x_2 et x_3 en fonction de b_1, b_2 et b_3

» A = [-1 2 1 ; -1 1 2 ; 1 -2 1];

A.Amirou

Chapitre 2

Les vecteurs et les matrices

A. Amirou

0.5 les Vecteurs ou tableau à une dimension

L'élément de base est une matrice à élément complexe. Ainsi tout nombre réel est considéré comme une matrice à une ligne et une colonne dont le seul élément est mla partie imaginaire nulle. Un vecteur n'est autre qu'une matrice à une ligne ou à une colonne.

0.5.1 création d'un vecteur

d éfinitions

Les éléments d'un vecteur sont entrés l'un après l'autre, séparés par des blancs ou des virgules, l'ensemble étant encadré par des crochets.

L'exemple suivant est un vecteur ligne à 5 éléments

```
» x=[1 4 5 6 7]
```

```
x=
```

```
1 4 5 6 7
```

Création d'un vecteur v : » v=[1.2 2 3 5 6 cos(0)]

affiche

```
v = 1.2000 2.0000 3.0000 5.0000 6.0000 1.0000
```

v est alors un vecteur ligne de 6 colonnes.

Le vecteur x est considéré comme une matrice à 1 ligne et 5 colonnes.

```
» size(x)
```

```
ans=
```

```
1 5
```

la commande size renvoie 2 valeurs, qui permet de récupérer les dimensions du vecteur

```
» [m n]= size(x)
```

```
m=
```

```
1
```

```
n=
```

```
5
```

La plus grande dimension d'un vecteur constitue sa longueur

```
» longueur_x=length(x)
```

```
longueur_x=
```

```
5
```

On peut concaténer plusieurs vecteurs pour en faire un seul.

```
» x1=[1 3 5]; x2=[2 4 6]
```

```
x=[x1 x2]
```

```
1 3 5 2 4 6
```

Pour supprimer un ou plusieurs éléments d'un vecteur, on les remplace par des ensembles vides.

```

    » x(3 :5)=[]
x=
1 3 6
Pour récupérer un élément ou une partie d'un vecteur
» x(2)
ans=
3
» x(2 :3)
ans=
3 6

```

Exemple Soit le vecteur v

```
» v=[1.2 2 3 5 6 cos(0)]
```

L'accès aux composantes d'un vecteur s'effectue directement par des commandes du genre : $v(3)$ ce qui donne à l'écran :

```
ans
= 3
```

MATLAB copie d'office le résultat dans la variable système ans (answer)

Remarque Dans MATLAB, les indices des vecteurs et matrices doivent être des entiers positifs. L'indice zéro n'est donc pas plus admis que les indices négatifs.

Pour accéder à un élément de vecteur :

```
» v(6)=(v(6)-0.5)*10
v = 1.2000 2.0000 3.0000 5.0000 6.0000 5.0000
```

Pour remplacer

$v(1)$ par $v(6)$ et $v(2)$ par $v(5)$:

1ère possibilité :

```
» v(1)=v(6);
» v(2)=v(5)
```

affiche
v = 5.0000 6.0000 3.0000 5.0000 6.0000 5.0000

2ème possibilité :

```
» v(1 :2)=v(6 :-1 :5) modifie le contenu de v et affiche
v = 5.0000 6.0000 3.0000 5.0000 6.0000 5.0000
```

3ème possibilité :

```
» v1=[1 2]; v2=[6 5];
» v(v1)=v(v2)
```

crée le vecteur v et affiche

```
v = 5.0000 6.0000 3.0000 5.0000 6.0000 5.0000
```

On peut éviter les crochets si les composantes du vecteur varient d'un pas constant. Le vecteur alpha suivant varie de $-\pi$ à π par pas de $\frac{\pi}{100}$.

```
» alpha=-pi :pi/100 :pi;
```

Si l'espacement est de 1, on met le signe " : " entre le premier et le dernier élément.

```
» x=-1 :5
x=
-1 0 1 2 3 4 5
```

0.5.2 Vecteurs lignes et vecteurs colonnes

vecteur-ligne

Un vecteur-ligne est introduit de la façon suivante :

```
» v = [ 5 , 2 , 1 3 , 4 ]
```

Le vecteur v s'affiche alors à l'écran :

```
v=
5 2 1 3 4
```

A.Amirou

» A = [2 3 5 7 11 13]

Le vecteur A s'affiche alors à l'écran :

v=

2 3 5 7 11 13

Si l'introduction est terminée par un point-virgule, on évite l'affichage du vecteur.

vecteur-colonne

Un vecteur-colonne peut être introduit en remplaçant les virgules par des points-virgules ou des retours de chariot.

un vecteur colonne se construit comme un vecteur ligne, en séparant les éléments par des points-virgules à la place des espaces ou des virgules.

Exemple

vecteur colonne

» A=[2;3;5;7;11;13]

convertir vecteur-colonne vecteur-ligne

Il est possible de convertir un vecteur colonne en un vecteur ligne et inversement, par transposition. Cette opération est réalisée par l'apostrophe, "'".

vecteur ligne

» At=A'

At=

2 3 5 7 11 13

0.6 Fonctions propres aux vecteurs

Des fonctions qui s'appliquent directement aux vecteurs

»x=[2 8 4 3 5];

somme des éléments du vecteur »sum(x)

ans=

22

Différence des éléments du vecteur

»diff(x)

ans=

6 -4 -1 2

La différence se fait de droite vers la gauche et on obtient un élément en moins.

Produit des éléments du vecteur

»produit=prod(x)

produit=

960

Cette fonction peut servir pour un calcul de la factorielle d'un nombre entier naturel.

»Fact_5=prod(1 :5)

Fact_5=

120

A.Amirou

0.6.1 Opérations entre vecteurs

L'addition et la soustraction de vecteur de même dimension se font élément par élément.

Soient les 2 vecteurs

»x1=[4 6 2]; x2=[3 -2 4];

La somme est la différence x1-x2 donne des vecteurs de mêmes dimensions

»x1+x2

ans=

7 4 6

»x1-x2

```
ans=
1 8 -2
»x1*x2
??? Error using *
Inner matrix dimensions must agree.
```

0.6.2 Opérations entre vecteurs et scalaires

L'ajout et la soustraction d'un scalaire à un vecteur »x1=[1 2 -3]; x2=[6 4 5];
 Considerons le vecteur x1 auquel on ajoute puis on retranche la valeur 5.

```
»plus_x1=x1+5
plus_x1=
6 7 2
»moins_x2=x1-5
moins_x1=
-4 -3 -8
```

Multiplier un scalaire par un vecteur revient à le faire pour chacun de ses éléments.

```
»2*x1
ans=
2 4 -6
```

On peut enchaîner plusieurs opérations de ce type dans la même ligne de commandes. »2*x1-x2

```
ans=
-4 0 -11
```

Il est de même pour la division par un scalaire. Tous les éléments du vecteur sont multipliés ou divisés par ce scalaire.

```
»double_x1=x1*2
double_x1=
2 4 -6
»quart_x=x2/4
quart_x2=
1.5000 1.0000 1.2500
```

A. Amirou

0.6.3 Produit scalaire

Le produit scalaire entre deux vecteurs se fait en multipliant un vecteur ligne par un vecteur colonne.

Un vecteur ligne $A = [a_1 a_2 \dots a_n]$

et un vecteur colonne

»B=[b₁; b₂; ...; b_n]

peuvent être multipliés en utilisant '*'. le resultat sera le suivant

$A*B = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$

pour obtenir un vecteur colonne à partir d'un vecteur ligne, il suffit de le transposer

»x1=[4 6 2];

»x2=x1'

x2=

4

6 Le produit scalaire entre x1 et x2 revient à faire le produit de x1 par x2.

2

»x1*x2

ans=

32

Le produit de x2(vecteur colonne) par x1(vecteur ligne) donne une matrice.

```

»x2*x1
ans=
 8 12 4
16 24 8
 0 0 0

```

0.7 Opérations éléments par éléments

Matlab utilise un type d'opérations particulières appelées "array operations". Ces opérations concernent $*$, \wedge , $/$ et \backslash . Lorsque un de ces opérateurs est précédé d'un point, l'opération est effectuée sur chaque élément de la matrice.

```

» [1 :10].^2
ans =
1 4 9 16 25 36 49 64 81 100
» [1 :4]./[2 :5]
ans =
0.5000 0.6667 0.7500 0.8000
» [1 2 3; 4 5 6; 7 8 9] .* eye(3)
ans = 1 0 0 0 5 0 0 0 9

```

Ces opérateurs sont très utiles pour tracer des graphiques. Lorsqu'on effectue des calculs avec des nombres complexes, la transposition non complexe conjuguée est obtenue avec l'opérateur $'$.

```

» Z1 = [1; i]', Z2 = [1; i].'
```

```

Z1 =
1.0000 0 - 1.0000i
Z2 =
1.0000 0 + 1.0000i

```

Exercice Calculer M, P et Q. $M = [1,2] * [3;4]$, $P = [1;2] * [3,4]$, $Q = [1,2] * [3,4]$

Solution

```
M =
```

```
11
```

```
P =
```

```
 3 4
 6 8 ??? Error using
```

```
==> * Inner matrix dimensions must agree.
```

A.Amirou

1 Calcul sur les polynômes

Matlab représente les polynômes sous forme de vecteur lignes dont les composantes sont ordonnées par ordre de puissance décroissante. Un polynôme de degré n est représenté par un vecteur de taille (n+1).

Exemple

$$f(x) = 2x^5 - x^3 + 5x^2 + 8x + 2$$

est représenté par le vecteur ligne

```
»f=[ 2 0 -1 5 8 2 ]
```

```
f=
```

```
2 0 -1 5 8 2
```

$$p(x) = 3x^2 - 5x^2 + 2$$

On commence par définir un " vecteur " qui contient les coefficients du polynôme : » $p = [3 -5 2]$

```
p =
```

```
3 -5 2
```

1.1 Racines d'un polynôme

Les racines d'un polynôme sont données par la fonction **roots**.

Exemple 1 : $3x^2 - 5x + 2 = 0$

On commence par définir un " vecteur " qui contient les coefficients du polynôme : » $p = [3 -5 2]$

$p = 3 -5 2$

» roots(p)

ans =

1.0000

0.6667

Exemple 2 : $x^2 - 4x + 4 = 0$

» $p = [1 -4 4]$

$p = 1 -4 4$

» roots(p)

ans =

2

2

Exemple 3 : $x^2 + 3x + 8 = 0$

» $p = [1 3 8]$

$p = 1 3 8$

» roots(p)

ans =

-1.5000 + 2.3979i

-1.5000 - 2.3979i

Exemple 4 : $x^5 + 2x^4 - 2x^3 + 4x^2 + 3x + 5 = 0$

» $p = [1 2 -2 4 3 5]$

$p = 1 2 -2 4 3 5$

» roots(p)

ans =

-3.0417

0.9704 + 1.0983i

0.9704 - 1.0983i

-0.4495 + 0.7505i

-0.4495 - 0.7505i

Exemple 5 : polynôme à coefficients complexes : $(1 + i)x^2 + (2 - 5i)x^2 + 3.5 = 0$

» $p = [1+i 2-5i 3.5]$

» roots(p)

ans =

1.7116 + 4.0248i

-0.2116 - 0.5248i

A.Amirou

1.2 Détermination des coefficients d'un polynôme

Inversement, on peut construire un polynôme à partir de ses racines en utilisant la fonction **poly**

Exemple 1 :

» $a = [2 1]$;

» poly(a)

ans =

1

-3

2

(c'est-à-dire : $x^2 - 3x + 2 = 0$)

1.3 Produit de polynômes

Exemple $(x-2)(x-1) = ?$

» p1=[1 -2];
 » p2=[1 -1];
 » conv(p1 , p2)
 ans =
 1 -3 2

Autrement dit : $(x-2)(x-1) = x^2 - 3x + 2 = 0$

Exercice 1 Donner le produit de polynomes ?

$(3x^2 - 5x + 2)(x^2 + 3x + 8) = ?$

Résultat

» p1=[3 -5 2];
 » p2=[1 3 8];
 » conv(p1 , p2)
 ans = 3 4 11 -34 16

Autrement écrit : $(3x^2 - 5x + 2)(x^2 + 3x + 8) = 3x^4 + 4x^3 + 11x^2 - 34x + 16$

Exercice 2 :

A partir des racines calculer le polynome ?

Verifier les racines.

Résultat

» a = [2+i 2-3i 5]
 » poly(a)
 ans =

1.0000 - 9.0000 + 2.0000i 27.0000 - 14.0000i - 35.0000 + 20.0000i

Vérification :

» p = ans

p =

1.0000 - 9.0000 + 2.0000i 27.0000 - 14.0000i - 35.0000 + 20.0000i

» roots(p)

ans =

2.0000 - 3.0000i 5.0000 - 0.0000i 2.0000 + 1.0000i

A.Amirou

1.4 Fractions rationnelles : Décomposition en éléments simples

La commande $[r, p, k] = \text{residue}(n, d)$

décompose la transformée en éléments simples du premier ordre.

$$\frac{N(p)}{D(p)} = \frac{r_1}{p-p_1} + \frac{r_2}{p-p_2} + \frac{r_k}{p-p_k}$$

Exemple

$$\frac{6}{x^4 + 6x^3 + 11x^2 + 6} = ?$$

polynôme du numérateur

» n=[6];

polynôme du dénominateur :

» d=[161160];

» [r , p , k] = residue (n , d)

r = -1.0000 3.0000 -3.0000 1.0000

p = -3.0000 -2.0000 -1.0000 0

k = []

Finalemnt :

$$\frac{6}{x^4 + 6x^3 + 11x^2 + 6} = \frac{-1}{x+3} + \frac{3}{x+2} + \frac{-3}{x+1} + \frac{1}{x}$$

calcul de f(x)**exemple**

$$y = f(x) = 3x^2 - 5x + 2$$

» p = [3 -5 2]

p = 3 -5 2

Calcul de f(x = 1) :

» polyval(p , 1)

ans =

0

Calcul de f(x = 2) :

» polyval(p , 2)

ans =

4

A.Amirou

2 Matrices

La plupart des opérations et des fonctions agissent directement sur une matrice toute entière ou bien colonne par colonne. Un tableau permet de stocker plusieurs valeurs à la fois en pouvant accéder à chacune de manière positionnelle.

2.1 Créer des Matrices

Pour créer une matrice il suffit d'ouvrir un [, énumérer les éléments ligne par ligne, puis fermer le].

- Sur une même ligne les éléments sont séparés par une virgule ou un espace, deux lignes successives sont séparées par un point virgule ou un retour chariot. Par exemple, Pour déclarer ce tableau, il suffit de taper :

```
>> A = [1 2 3 4 ; 5 6 7 8 ; 9 10 11 12 ; 13 14 15 16]
```

crée une matrice 4 × 4 appelée A.

- L'instruction suivante crée la même matrice, mais le résultat n'est pas affiché car un point-virgule termine l'instruction :

```
» A = [1, 2, 3, 4
```

```
5, 6, 7, 8
```

```
9 10 11 12; 13, 14, 15, 16
```

```
];
```

- On peut aussi créer la même matrice A avec l'instruction suivante : » A = [1 :4; 5 :8; 9 :12; 13 :16]

Nous avons utilisé ici l'opérateur deux-points.

La réponse de Matlab est

A=	1	2	3	4
	5	6	7	8
	9	10	11	12
	13	14	15	16

- o On peut accéder indépendamment à chaque valeur du tableau. Si on veut accéder à la valeur dans la troisième ligne et dans la deuxième colonne, on tape A(3,2).
- o On peut également faire des tableaux à une "dimension" appelés vecteurs :
 - » b=[1 2 3 4]

Pour saisir une matrice à partir de scalaires, à partir de vecteurs ou à partir d'autres matrices :

- >> A = [1 2 3 4 ; 5 6 7 8 ; 9 10 11 12 ; 13 14 15 16]

- » A=[v1 ;v2 ;v3 ;v4]; si v1, v2 , v3 et v4 sont des vecteurs lignes

– » $A=[v1\ v2\ v3\ v4]$; si $v1, v2, v3$ et $v4$ sont des vecteurs colonnes

Pour accéder aux éléments d'une matrice, on utilise la notation standard de l'algèbre linéaire : $A(i,j)$ fait référence à l'élément de la ligne i et de la colonne j de A . Dans l'exemple précédent, l'instruction $A(2,3)$ retourne la valeur 7. Seul un entier positif non nul est accepté comme indice de matrice ou de vecteur. On peut extraire plusieurs éléments simultanément.

Exemple :

» $A([1,3],2)$, $A([1,3],[2,1,4])$

On utilise l'opérateur deux-points pour extraire

- une ligne, L'instruction $A(2, :)$ extrait la deuxième ligne,
- une colonne, $A(:,1)$ la première colonne
- ou une sousmatrice, $A(2 :4,1 :3)$ retourne le bloc

```
ans =
     5     6     7
     9    10    11
    13    14    15
```

A.Amirou

Remarque

L'instruction $A(:)$ crée un vecteur colonne contenant les éléments de A énumérés colonne par colonne. Notons que l'instruction $A(7)$ retourne la valeur 10, car les éléments d'une matrice sont numérotés selon ce même ordre d'énumération. Ainsi, l'élément $A(i,j)$ est numéroté $i+m*(j-1)$, où m est le nombre de lignes de A .

1	m+1	2m+1	...
2	m+2	2m+2	...
.	.	.	.
m	m+m	2m+m	...

Cette numérotation permet d'extraire des éléments avec un tableau d'indices. Par exemple, l'instruction

» $A([1\ 2\ 3; 4\ 5\ 6])$

retourne

```
ans =
     1     2     9
    13     2     6
```

Le nombre de lignes et de colonnes dans une matrice s'obtient par la fonction `size`. » `size(A)`

ans=

```
2 3
```

Le rang de la matrice A

» `rank(A)`

ans=

```
2
```

Modifier les éléments d'une matrice On peut modifier les éléments d'une matrice en leur affectant de nouvelles valeurs :

» $A(4,4) = 0$, $A(1, :) = A(1, :)*10$, $A = [A; [1\ 1\ 1\ 1]]$

Pour concaténer deux matrices :

» $M=[A\ A]$

```
A=
     1     2     3     4     1     2     3     4
     5     6     7     8     5     6     7     8
     9    10    11    12     9    10    11    12
    13    14    15    16    13    14    15    16
```

Aide L'instruction `help` permet d'obtenir de l'aide. Par exemple,

» `help diag`

renvoie l'aide en ligne de la fonction `diag`.

2.2 Créer des Matrices à partir des fonctions

. Certaines fonctions permettent de créer des matrices particulières :

- eye(n) crée une matrice identité de dimension n ;
- ones et zeros créent des matrices de 1 et de 0 ;
- rand et randn des matrices dont les éléments sont choisis au hasard selon une loi uniforme sur]0, 1[et une loi normale centrée réduite ;
- magic(n) crée un carré magique de dimension $n(n > 3)$; diag permet d'extraire des diagonales ou bien de créer des matrices diagonales ;
- triu et tril permettent d'extraire les parties triangulaire supérieure et triangulaire inférieure.

2.3 Opérations sur les matrices

2.3.1 Opérations algébriques

+	addition
-	soustraction
*	produit
'	transposition (complexe conjuguée)
^	puissance
inv(X)	inverse d'une matrice
/	division à droite : A / B est équivalent à $A * \text{inv}(B)$
\	division à gauche : $A \setminus B$ est équivalent à $\text{inv}(A) * B$

Exemple

Saisie d'une matrice carrée de taille 3 x 3 :

» A = [2 4 5 ; 1 5 7 ; -3 3 1]

A =
 2 4 5
 1 5 7 » A(2 , 3)
 -3 3 1

ans

= 7

» A(2 , 3) = 6

A =
 2 4 5
 1 5 6
 -3 3 1

» A'

ans =

2 1 -3
 4 5 3
 5 6 1

A.Amirou

2.4 matrice inverse

» inv(A)

ans =

1.0833 -0.9167 0.0833
 1.5833 -1.4167 0.5833 » D = A * inv(A)
 -1.5000 1.5000 -0.5000

D =

1.0000 0.0000 0.0000
 0.0000 1.0000 0.0000 » rank(A)
 0.0000 0.0000 1.0000

ans =

3

» det(A)

ans =

-12

```

» eye(7)
ans =
  1 0 0 0 0 0
  0 1 0 0 0 0
  0 0 1 0 0 0
  0 0 0 1 0 0 » B = [ 1 1 0; 1 0 1; 0 1 1 ]
  0 0 0 0 1 0
  0 0 0 0 0 1
B =
  1 1 0
  1 0 1
  0 1 1
» A + B
ans =
  3 5 5
  2 5 7
 -3 4 2
» 2 + A
ans =
  4 6 7
  3 7 8
 -1 5 3
» 2 * A
ans =
  4 8 10
  2 10 12
 -6 6 2
» A * B
ans =
  6 7 9
  6 7 11
  0 -2 4
» B * A
ans =
  3 9 11
 -1 7 6
 -2 8 7
» A*A*A
ans =
 -88 304 262
 -98 314 268
 -18 18 10
» A^3
ans =
 -88 304 262
 -98 314 268
 -18 18 10
Saisie d'une matrice à coefficients complexes de taille 2 x 3 :
» C = [ 1 + i 0 0; 1 - i i 2]
C =
  1.0000 + 1.0000i    0    0
  1.0000 - 1.0000i    0 + 1.0000i    2.0000
» C * A
ans =

```

A.Amirou

```
2.0000 + 2.0000i   4.0000 + 4.0000i   5.0000 + 5.0000i
-4.0000 - 1.0000i  10.0000 + 1.0000i   7.0000 + 1.0000
```

Quand les opérandes sont une matrice et un scalaire (par exemple $A+1$, $A-1$, $A*2$ ou $A/2$), l'opération est effectuée entre le scalaire et chaque élément de la matrice.

```
» twos = ones(2) + 1
```

```
twos =
```

```
2 2
```

```
2 2
```

```
exemple »A=[1 2 3; 4 5 6; 7 8 9]
```

```
» [m n]=size(A);ones(m,n)
```

```
ans=
```

```
1 1 1
```

```
1 1 1
```

```
1 1 1
```

Il est possible de simplifier par

```
» ones(size(A))
```

A.Amirou

Exercice

```
Créer la matrice suivante : A=
      6  43  2  11  87
     12  6  34  0  5
     34  18  7  41  9
```

1. Créer un vecteur colonne de 5 lignes nommé va qui contient les éléments de la seconde ligne de A
2. Créer un vecteur colonne de trois éléments nommé vb qui contient les éléments de la quatrième colonne de A
3. Créer un vecteur colonne de 10 éléments nommé vc qui contient les éléments de la première et de la seconde ligne de A
4. Créer un vecteur colonne de 6 éléments nommé vd qui contient les éléments de la première et de la cinquième colonne de A
5. Créer un vecteur colonne de 3 éléments nommé vg qui contient la somme des éléments de la première colonne plus les éléments de la quatrième colonne de A

2.4.1 La division

Les opérateurs de division permettent de calculer une solution d'une équation linéaire. Lorsque A est inversible, $A \backslash b$ calcule la solution de l'équation $Ax=b$. C'est la même chose que $\text{inv}(A)*b$, la solution étant calculée sans inversion explicite de la matrice. Matlab utilise une méthode de Gauss avec pivotage partiel. Si la matrice est singulière, Matlab retourne un message d'avertissement.

```
Par exemple,
» [1 3;2 6]\[1;1]
```

```
Warning : Matrix is singular to working precision. ans =
```

```
Inf
```

```
Inf
```

L'opérateur de division à droite est défini par $b/A = (A' \backslash b)'$. Si A est inversible, c'est la même chose que $b*\text{inv}(A)$.

2.5 Construction de matrices à partir de vecteurs

```
»price=[ 3.00 1.99 10.99 9.15 1.29]
»jan=[ 3 2 1 5 2]
»feb=[ 2 3 1 3 3]
»march=[ 1 0 3 3 3]
matrice quatity »quantity=[ jan' feb' march']
```

Cette commande permet d'obtenir une matrice. Notons que ce calcul ne peut s'obtenir avec les vecteurs lignes jan, feb et march. Il faut au préalable les transposer.

```
quantity= 3 2 1
          2 3 0
          1 1 3
          5 3 3
          2 3 3
```

La dimension de la matrice quantity » [m n]=size(quantity)

```
m =
```

```
5
```

```
n =
```

```
3
```

La plus grande la dimension de la matrice quantity » length(quantity)

```
ans=
```

```
5
```

On dispose de plusieurs mode d'affichage

```
»disp(quantity)
```

```
printmat (quantity, 'matrice quantity' , 'i1 i2 i3 i4 i5','c1 c2 c3')
```

```
quantity=
      c1  c2  c3
i1  3  2  1
i2  2  3  0
i3  1  1  3
i4  5  3  3
i5  2  3  3
```

2.5.1 Produit d'un vecteur par vecteur

```
»totaljan    =price*jean'      % = 72.30
»totalfeb    =price*feb'      % = 54.28
»totalmarch  =price*march'    % =67.29
```

2.5.2 Produit d'un vecteur par une matrice

```
»expense=price*quantity      % resultat : [72.30 54.28 67.29]
```

2.5.3 Opérateurs logiques

```
Les opérateurs logiques sont  &   et
                               |   ou
                               ~   non.
```

Calculons une table logique :

```
» a = [0;0;1;1]; b = [0;1;0;1]; EtOuDonc = [a,b,a&b,a|b,(~a)|b]
```

```
EtOuDonc =
 0  0  0  0  1
 0  1  0  1  1
 1  0  0  1  0
 1  1  1  1  1
```

Une valeur numérique est considérée comme vraie (= 1) si elle est non nulle, sinon elle est fausse (= 0). Par exemple, ~A renvoie une matrice dont les éléments valent 1 là où A a des éléments nuls et 0 ailleurs.

Exercice Quel est le résultat de l'instruction suivante ?

```
» M = [1 2 3; 4 5 6; 7 8 9]; M = diag(diag(M))
```

2.6 Fonctions usuelles

2.6.1 Fonctions scalaires

Ce sont les fonctions mathématiques usuelles.

```

sin   cos   tan   acos   asin   atan
sinh  cosh  tanh  acosh  asinh  atanh
exp   log   log10  abs   sqrt   sign
rem   round floor  ceil   fix
real  imag  conj   angle

```

L'argument d'entrée peut être un scalaire, un vecteur ou une matrice. L'argument de sortie est une matrice de même dimension, la fonction étant appliquée sur chacun de ses éléments.

2.6.2 Fonctions vectorielles

Lorsque l'argument d'entrée de ces fonctions est un vecteur, le résultat est un scalaire.

```

min   max   sum   prod  sort
median mean  std   any   all

```

```

» [min(rand(1,10000)), max(rand(1,10000))]
ans =
0.0001 0.9999

```

Lorsqu'une telle fonction est appliquée à une matrice, elle agit colonne par colonne et retourne un vecteur ligne.

```

» sum(magic(5))
ans =
65 65 65 65 65

```

Les fonctions any et all sont très utiles. Si x est un vecteur, any(x) renvoie la valeur logique 1 si au moins une composante du vecteur est non nulle, sinon elle renvoie la valeur logique 0; all(x) renvoie la valeur logique 1 si toutes les composantes de x sont non nulles, sinon elle renvoie 0.

Considérons la matrix rectangulaire X suivante

```

»X=[1 2 3; 4 5 6] X=
1 2 3
4 5 6

```

- Moyenne des élément de chaque colonne de X


```

      »moyenne=mean(X)
      moyenne =
      2.5000  3.5000  4.5000
      
```
- Ecart type élément de chaque colonne de X


```

      »ecart_type=std(X)
      ecart_type=
      2.1213  2.1213  2.1213
      
```
- valeur minimale et maximale de chaque colonne de X


```

      »mn=min(X)
      mn=
      1 2 3
      »mx=max(X)
      mx=
      4 5 6
      
```
- Somme des élément de chaque colonne de X


```

      »som=sum(X)
      som=
      5 7 9
      
```

A. Amirou

- Produit des élément de chaque colonne de X
 »prod(X)
 ans=
 4 10 18
- Transformation de tous les éléments en un seul vecteur X(:) colonne
 »X(:)'
 ans=
 1 2 3 4 5 6
- Moyenne des éléments de la matrice X
 »moy_X=mean(X(:))
 moy_X=
 3.5000
- On peut realiser la même opération sur tous les éléments de la matrice en doublant valeur minimum
 » min(min(X))
 ans=
 1
 valeur maximum
 » max(max(X))
 ans=
 6

2.6.3 Fonctions matricielles

Ce sont des fonctions usuelles de l'analyse numérique matricielle. L'argument d'entrée est une matrice.

cond	conditionnement
det	déterminant
norm	norme 1, 2, Frobenius, ∞
rank	rang
trace	trace
chol	factorisation de Cholesky
inv	inverse
lu	factorisation LU
qr	factorisation QR
eig	valeurs propres et vecteurs propres
poly	polynôme caractéristique
svd	décomposition en valeurs singulières
expm	exponentielle de matrice
sqrtm	racine carrée de matrice

A.Amirou

Dans l'exemple ci dessous, on calcule les vecteurs propres (V) et les valeurs propres (diagonale de D) de la matrice $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ » [V,D] = eigs(ones(2,2))

$$V = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

$$D = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

2.7 Matrices

Matlab est particulièrement adapté lorsque l'on veut faire des applications numériques sur des matrices. Voyons quelques méthodes permettant de les manipuler.

Définissez une matrice $M = [1 \ 2; 3 \ 4]$ puis essayez les opérations suivantes dans l'interpréteur :

- » $2 * M + 3$
- » $M + M$
- » $\text{sqrt}(M)$
- » $M * M$
- » $M .* M$
- » $\text{ones}(4)$
- » $\text{ones}(3, 5)$

Quelle est la différence entre les opérateurs $*$ et $.*$? Que fait la fonction ones ?

Réponse : L'opérateur $*$ désigne le produit de deux matrices (comme vous l'avez vu en maths), tandis que $.*$ désigne le résultat du produit terme à terme de deux matrices de mêmes dimensions : La fonction ones crée une matrice ayant les dimensions indiquées ne contenant que des 1.

Comment peut-on créer facilement une matrice 54×42 ne contenant que des 7?

Réponse : » $M = 7 * \text{ones}(54, 42)$;

Si l'on veut créer une matrice $(a_{i,j})$ de dimensions 8×9 définie par $a_{i,j} = i+j$, on peut utiliser un script et des boucles for . Cependant dans ce cas une seule boucle ne suffit pas, et il va falloir en utiliser deux " imbriquées " : une première boucle qui parcourt les lignes de la matrice (les différentes valeurs de i) et une seconde boucle qui parcourt les colonnes (les valeurs de j). **Exercice** Créez un tableau tab contenant les entiers pairs de 0 à 42, puis écrivez un script (une fonction) qui parcourt le tableau et remplace chacune des valeurs par son carré.

Remarque : La fonction size renvoie les dimensions d'une matrice. Ici tab est un tableau d'une ligne et 22 colonnes, mais seule la seconde dimension (nombre de colonnes) nous intéresse, d'où $\text{size}(\text{tab}, 2)$ qui demande la taille de tab dans la deuxième dimension. où bien on utilise $\text{length}(\text{tab})$.

A.Amirou

Chapitre 3

équations linéaires

Sommaire

0.8	Résolution d'un système d'équations linéaires	35
0.9	Inversion de matrice	35
0.10	Exemple de résolution d'équations linéaires	36
0.11	Règle de Cramer	36
0.12	Exercices	37

0.8 Résolution d'un système d'équations linéaires

Tout système linéaire peut être représenté sous forme matricielle. La résolution d'un tel système fait appel à la notion d'inverse d'une matrice. Nous proposons le système suivant, comme exemple de résolution linéaire

Exemple 0.1.
$$\begin{aligned} 2x_1 + 3x_2 &= 8 \\ x_1 - 2x_2 &= -3 \end{aligned}$$

Ce système peut être écrit sous une forme matricielle : $AX = B$, avec :

$$A = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad B = \begin{bmatrix} 8 \\ -3 \end{bmatrix} \quad \text{» } A=[2 \ 3; 1 \ -2]; \ B=[8 \ -3]; \ \text{Resolution :}$$

Méthode1 $X = A \setminus B$

Méthode2 $x = \text{inv}(A) * B$

Méthode3 Méthode de Gramer

» `disp('solutions :'); disp(['x1 =', num2str(x(1)), ' et x2 =', num2str(x(2))])`

solutions :

x1=1 et x2=2

A.Amirou

0.9 Inversion de matrice

La matrice A est une matrice carrée de déterminant différent de zero. Une telle matrice a une matrice invers qui se calcule par

» `A_inv=inv(A)`

Dans ce cas, il est possible de résoudre le système d'équation en multipliant à gauche, le vecteur B par la matrice inverse de A.

» `X=A_inv*B`

0.10 Exemple de résolution d'équations linéaires

Nous pouvons résoudre à l'aide de l'opérateur " \ "

Exemple 0.2. $A = \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}$

$B = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$
 » $A = [2 \ -1 ; 1 \ 1]$

» $B = [2 \ 5]$

» $X = A \setminus B$

$X =$
 2.3333
 2.6667

Le rang de la matrice

» $\text{rank}(A), \text{rank}([A \ B])$

ans =

2

ans =

2

Exemple 0.3. Soit le système d'équations paramétriques :

$$\begin{aligned} -x_1 + 2x_2 + x_3 &= b_1 \\ -x_1 + 2x_2 + 2x_3 &= b_2 \\ x_1 - 2x_2 + x_3 &= b_3 \end{aligned} \quad \text{On cherche}$$

à exprimer x_1, x_2 et x_3 en fonction de b_1, b_2 et b_3

» $A = [-1 \ 2 \ 1 ; -1 \ 1 \ 2 ; 1 \ -2 \ 1]$;

» $\text{inv}(A)$

ans =

2.5000 -2.0000 1.5000
 1.5000 -1.0000 0.5000
 0.5000 0 0.5000

» format rational

» $\text{inv}(A)$

ans =

5/2 -2 3/2
 3/2 -1 1/2
 1/2 0 1/2

Finalement

$$\begin{aligned} x_1 &= \frac{5}{2}b_1 - 2b_2 + \frac{3}{2}b_3 \\ x_2 &= \frac{3}{2}b_1 - b_2 + \frac{1}{2}b_3 \\ x_3 &= \frac{1}{2}b_1 + 0 + \frac{1}{2}b_3 \end{aligned}$$

A.Amirou

0.11 Règle de Cramer

Théorème 0.1. Règle de Cramer Soit A une matrice $n \times n$ inversible. Pour tout $b \in R^n$, les composantes de l'unique solution du système d'équations $Ax = b$ sont données par

$$x_i = \frac{|A_i(b)|}{|A|}, \quad i = 1, 2, \dots, n$$

$$A = (a_1, a_2, \dots, a_{i-1}, \underline{a_i}, a_{i+1}, \dots, a_n)$$

$$A_i(b) = (a_1, a_2, \dots, a_{i-1}, \underline{\mathbf{b}}, a_{i+1}, \dots, a_n)$$

Pour résoudre le système d'équations linéaires $AX = B$

$$\text{avec : } A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 2 & 5 \end{pmatrix} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad B = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix}$$

Il faut calculer d'abord les matrices D1, D2, D3 avec $D_i = A$ ayant la colonne (i) remplacée par B

- » $D1 = A$; $D1(:, 1) = B$
- » $D2 = A$; $D2(:, 2) = B$
- » $D3 = A$; $D3(:, 3) = B$

Ensuite, il faut calculer les déterminants des matrices A, D1, D2 et D3 : » $\det(A)$

- » $\det(D1)$
- » $\det(D2)$
- » $\det(D3)$

La solution par règle de Cramer donne $X(i) = \det(D_i) / \det(A)$

- » $X = [\det(D1) ; \det(D2) ; \det(D3)] / \det(A)$

Pour vérifier la solution, on devrait retrouver B en faisant :

- » $A * X$

0.12 Exercices

Exercice 0.1. (résolu) Considérons le système d'équations suivant :

$$\begin{aligned} 3x + 2y - z &= 10 \\ -x + 3y + 2z &= 5 \\ x - y - z &= -1 \end{aligned}$$

Ce système peut être écrit sous une forme matricielle : $AX = B$, avec :

$$A = \begin{bmatrix} 3 & 2 & -1 \\ -1 & 3 & 2 \\ 1 & -1 & -1 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} 10 \\ 5 \\ -1 \end{bmatrix}$$

Exercice 0.2. Soit à résoudre un système de 3 équations à 3 inconnues x_1, x_2 et x_3 :

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 4 \\ x_1 + 5x_2 + 2x_3 &= -1 \\ 4x_1 - 2x_2 + 3x_3 &= 3 \end{aligned}$$

Exercice 0.3. Soit le système d'équations paramétriques :

$$\begin{aligned} -x_1 + 2x_2 + x_3 &= b_1 \\ -x_1 + 2x_2 + 2x_3 &= b_2 \\ x_1 - 2x_2 + x_3 &= b_3 \end{aligned}$$

On cherche à exprimer x_1, x_2 et x_3 en fonction de b_1, b_2 et b_3

Exercice 0.4. Résoudre le système d'équations :

$$\begin{aligned} 1x_1 + 3x_2 &= 6 \\ 2x_1 + 4x_2 &= 10 \end{aligned}$$

On cherche à exprimer x_1 et x_2

Solution 0.2 Résoudre ce système d'équations, c'est trouver X tel que $AX = B$. MATLAB permet de résoudre ce problème en écrivant les instructions suivantes :

$$A = [3 \ 2 \ -1; -1 \ 3 \ 2; 1 \ -1 \ -1];$$

$$B = [10; 5; -1];$$

$$X = \text{inv}(A)*B$$

On obtient comme résultat :

```
X =
-2.0000
5.0000
-6.0000
```

Note : on peut aussi utiliser la commande MATLAB

$$X = A \setminus B$$

Solution 0.3 On saisit les différents coefficients dans une matrice 3x3

$$\gg A = [3 \ 2 \ 1; -1 \ 5 \ 2; 4 \ -2 \ 3]$$

On complète avec un vecteur colonne 3x1

$$\gg B = [4; -1; 3]$$

La solution est donnée sous la forme d'un vecteur colonne

$$\gg A \setminus B$$

```
ans =
1.3279
0.2951
-0.5738
```

Autre méthode

$$A_inv =$$

$$\gg \text{inv}_A = \text{inv}(A)$$

$$\text{inv}_A = \begin{bmatrix} 0.3115 & -0.1311 & -0.0164 \\ 0.1803 & 0.0820 & -0.1148 \\ -0.2951 & 0.2295 & 0.2787 \end{bmatrix}$$

$$X = \begin{bmatrix} -1.40000 \\ 1.80000 \\ 0.60000 \end{bmatrix}$$

$$\gg \text{inv}(A)*B$$

```
ans =
1.3279
0.2951
-0.5738
>X(1)
ans =
1.3279
```

A.Amirou

»X(2)

ans =

0.2951

»X(3)

ans =

-0.5738

Verification

»3X(1)+2X(2)+X(3)

ans=

4

»-X(1)+5X(2)+2X(3)

ans=

-1

»4X(1)-2X(2)+3X(3)

=

3

A.Amirou

Chapitre 4

Travaux Pratiques

Matlab peut être vu comme une calculatrice extrêmement puissante. Les opérations simples peuvent être tapées directement, et l'on obtient le résultat en appuyant sur la touche "Entrée". 1. Essayez de faire quelques opérations dans l'interpréteur :

```
>> 10 + 5
>> 10 * 5
>> 105
>> 10/5
```

Afin d'affecter une variable, on utilise le signe =. Ainsi la ligne :

```
>> var1 = 3
```

2. Familiarisez-vous avec les variables :

```
>> var1 = 52
>> var1
>> var2 = 32;
>> var2
>> var1 * var2
>> Var1 = 12
>> var1
>> 11 = var1
```

A.Amirou

3. A quoi sert le point virgule à la fin d'une ligne de commande? Que remarque-t'on à propos de la gestion des majuscules/minuscules dans les noms de variables? Réponse : Le point virgule sert à ne pas afficher dans l'interpréteur la valeur calculée/affectée.

4. Essayez par vous-même : » clear var1 » var1 » clear

1 Exercices avec solutions

Considérons le nombre complexe suivant : $z_1 = 1 + 2i$ et $z_2 = 3 - 5j$
Donner le code Matlab qui effectue les opérations arithmétiques et affiche le résultat de :

1. l'addition de z_1 et z_2

2. la division de z_1 sur z_2
3. la soustraction z_1 sur z_2
4. et la multiplication de z_1 par z_2
5. Calcule de z_2 à la puissance 2

Considérons le nombre complexe suivant : $z = [1+j \ 2-3j ; 4+2j \ 5-4j]$
Donner le code Matlab et afficher le résultat qui calcule :

1. la partie réelle de z
2. la partie imaginaire de z
3. le conjugué de z
4. et le module de z
5. produit de z par son conjugué
6. produit scalaire de z par son conjugué

Exercice

Créer la matrice suivante : $A = \begin{bmatrix} 6 & 43 & 2 & 11 & 87 \\ 12 & 6 & 34 & 0 & 5 \\ 34 & 18 & 7 & 41 & 9 \end{bmatrix}$

1. Créer un vecteur colonne de 5 lignes nommé v_a qui contient les éléments de la seconde ligne de A
2. Créer un vecteur colonne de trois éléments nommé v_b qui contient les éléments de la quatrième colonne de A
3. Créer un vecteur colonne de 10 éléments nommé v_c qui contient les éléments de la première et de la seconde ligne de A
4. Créer un vecteur colonne de 6 éléments nommé v_d qui contient les éléments de la première et de la cinquième colonne de A
5. Créer un vecteur colonne de 3 éléments nommé v_g qui contient la somme des éléments de la première colonne plus les éléments de la quatrième colonne de A

Exercice On se propose de tracer la courbe suivante $y = \cos 2x + 1/2 \sin \frac{x}{2}$
La variable x est un vecteur dont les valeurs vont de $-\pi$ à π par pas de $\frac{\pi}{100}$

1. Ecrire le code Matlab qui va tracer la courbe y en fonction de x
2. Donner un titre à cette figure " tracé de la courbe y "
3. Ajouter un label des axes x en abscisses " variable x "
4. Ajouter un label des y en ordonnées " variable y "
5. Utiliser la commande qui permet de réaliser un quadrillage afin de visualiser les valeurs en ordonnées et en abscisses

Exercice Soit la fonction suivante à deux variables $z = \frac{\sin(x^2+y^2)}{(x^2+y^2)}$
Ecrire le code Matlab qui va tracer la courbe z en fonction de x et de y .

Remarque : On génère deux matrices carrées X et Y qui définissent le domaine de calcul de z ; On utilise pour ceci une fonction `meshgrid` pour x et y variant de $-\pi$ à π par pas de $\frac{\pi}{100}$

Exercice Créez un tableau tab contenant les entiers pairs de 0 à 42, puis écrivez un script (une fonction) qui parcourt le tableau et remplace chacune des valeurs par son carré.

Exercice Soit le système d'équations paramétriques :

$$\begin{aligned} -x_1 + 2x_2 + x_3 &= b_1 \\ -x_1 + 2x_2 + 2x_3 &= b_2 \\ x_1 - 2x_2 + x_3 &= b_3 \end{aligned}$$

On cherche à exprimer x_1, x_2 et x_3 en fonction de b_1, b_2 et b_3
 »A=[-1 2 1; -1 1 2; 1 -2 1];

Solution Exercice1 Soit le nombre complexe suivant : $z_1 = 1+2i$ et $z_2 = 3-5j$

1. $z_1+z_2=4.0000 -3.000i$
2. $z_1-z_2=-2.0000+7.000i$
3. $z_1/z_2=-0.2.059+0.3235i$
4. $z_1*z_2=13.000+1.0000i$
5. Calculez z_2 à la puissance 2, $z_2^2=-16.0000-30.0000i$

Solution Exercice2 » $z=[1+j \ 2-3j; 4+2j \ 5-4j]$ $z = \begin{matrix} 1.0000 + 1.0000i & 2.0000 - 3.0000i \\ 4.0000 + 2.0000i & 5.0000 - 4.0000i \end{matrix}$

1. la partie réelle de z » $r=\text{real}(z)$ $z = \begin{matrix} 1 & 2 \\ 4 & 5 \end{matrix}$
2. la partie imaginaire de z $m=\text{imag}(z)$ $m = \begin{matrix} 1 & -3 \\ 2 & -4 \end{matrix}$
3. le conjugué de z » $\text{conj}(z)$ $\text{ans} = \begin{matrix} 1.0000 - 1.0000i & 2.0000 + 3.0000i \\ 4.0000 - 2.0000i & 5.0000 + 4.0000i \end{matrix}$
4. et le module de z » $\text{abs}(z)$ $\text{ans} = \begin{matrix} 1.4142 & 3.6056 \\ 4.4721 & 6.4031 \end{matrix}$
5. produit de z par son conjugué » $\text{conjz}=\text{conj}(z)$
 » $\text{produitselezconjz}=z.*\text{conjz}$
 » $\text{produitselezconjz}=\text{conjz}.*z$
 $\text{produitselezconjz} = \begin{matrix} 2 & 13 \\ 20 & 41 \end{matrix}$
6. produit scalaire de z par son conjugué » $\text{produitzconjz}=z*\text{conjz}$
 $\text{produitzconjz} = \begin{matrix} 4.0000 -16.0000i & 21.0000 - 2.0000i \\ 18.0 -28.0000i & 43.0000 +16.0000i \end{matrix}$

A.Amirou

Solution Exercice3 » $A=[6 \ 43 \ 2 \ 11 \ 87; 12 \ 6 \ 34 \ 0 \ 5; 34 \ 18 \ 7 \ 41 \ 9]$ $A = \begin{matrix} 6 & 43 & 2 & 11 & 87 \\ 12 & 6 & 34 & 0 & 5 \\ 34 & 18 & 7 & 41 & 9 \end{matrix}$

1. » $va=A(2, :)$ $va = 12 \ 6 \ 34 \ 0 \ 5$
 $vb = 11$
2. » $vb=A(:,4)$ 0
 41

```

3. vc=A(1 :2, :)   vc   6   43   2   11   87
                   12   6   34   0   5

                   vd =   6   87
4. » vd=A( :, [1,5])  12   5
                   34   9

                   vg =   17
5. » vg=A( :,1)+A( :,4)  12
                   5

```

Solution Exercice4 »x=pi :pi/100 :pi;
 »cos(2*x)+0.5*sin(x/2)
 »plot(x,y)
 »grid;
 »xlabel('variable x')
 » ylabel('variable y')

Solution Exercice5 » x=-pi :pi/100 :pi;
 »y=x;
 on génère deux matrices carrées X et Y qui définissent le domaine de calcul de z;
 On utilise pour ceci une fonction meshgrid
 »Z=sin(X.^2+Y.^2)./(X.^2+Y.^2);
 »mesh(Z)

Solution Exercice6 Script :MATLAB\pr1
 tab= 0 :2 :42;
 for i = 1 :size(tab,2)
 tab(i) = tab(i).^2
 end

A.Amirou

Exercice 7 Soit le système d'équations paramétriques :

$$\begin{aligned} -x_1 + 2x_2 + x_3 &= b_1 \\ -x_1 + 2x_2 + 2x_3 &= b_2 \\ x_1 - 2x_2 + x_3 &= b_3 \end{aligned}$$

On cherche à exprimer x_1, x_2 et x_3 en fonction de b_1, b_2 et b_3
 »A=[-1 2 1; -1 1 2; 1 -2 1];

Solution Exercice 7 »inv(A)
 ans=
 2.5000 -2.0000 1.5000
 1.5000 -1.0000 0.5000
 0.5000 0 0.5000
 »format rational
 »inv(A)
 ans=
 5/2 -2 3/2
 3/2 -1 1/2
 1/2 0 1/2
 Finalement

$$\begin{aligned} x_1 &= \frac{5}{2}b_1 - 2b_2 + \frac{3}{2}b_3 \\ x_2 &= \frac{3}{2}b_1 - b_2 + \frac{1}{2}b_3 \\ x_3 &= \frac{1}{2}b_1 + 0 + \frac{1}{2}b_3 \end{aligned}$$

