

Les Vecteurs et les Matrices sous *MATLAB*

R. CHOUDER et M. KAMEL

rafaa.chouder@univ-msila.dz
mohamed.kamel@univ-msila.dz

Mars 2021

Les Tableaux et les Vecteurs

- Un tableau est une collection d'objets de données du même type, généralement stockés séquentiellement en mémoire.
- Un tableau est une structure de données élémentaire.
- Presque tous les langages de programmation prennent en charge les tableaux.
- Matlab est un langage particulièrement spécialisé dans le support des tableaux (et par la suite des matrices).
- Un tableau est la manière la plus évidente de représenter un vecteur.
- Un vecteur 3D avec des coordonnées $[4, 1, 5, -5]$ serait représenté comme un tableau de 3 doubles disposés séquentiellement en mémoire.

4	1.5	-5
---	-----	----

Les Tableaux et les Vecteurs

- Un tableau est une collection d'objets de données du même type, généralement stockés séquentiellement en mémoire.
- Un tableau est une structure de données élémentaire.
- Presque tous les langages de programmation prennent en charge les tableaux.
- Matlab est un langage particulièrement spécialisé dans le support des tableaux (et par la suite des matrices).
- Un tableau est la manière la plus évidente de représenter un vecteur.
- Un vecteur 3D avec des coordonnées $[4, 1, 5, -5]$ serait représenté comme un tableau de 3 doubles disposés séquentiellement en mémoire.

4	1.5	-5
---	-----	----

Les Tableaux et les Vecteurs

- Un tableau est une collection d'objets de données du même type, généralement stockés séquentiellement en mémoire.
- Un tableau est une structure de données élémentaire.
- Presque tous les langages de programmation prennent en charge les tableaux.
- Matlab est un langage particulièrement spécialisé dans le support des tableaux (et par la suite des matrices).
- Un tableau est la manière la plus évidente de représenter un vecteur.
- Un vecteur 3D avec des coordonnées $[4, 1, 5, -5]$ serait représenté comme un tableau de 3 doubles disposés séquentiellement en mémoire.

4	1.5	-5
---	-----	----

Les Tableaux et les Vecteurs

- Un tableau est une collection d'objets de données du même type, généralement stockés séquentiellement en mémoire.
- Un tableau est une structure de données élémentaire.
- Presque tous les langages de programmation prennent en charge les tableaux.
- Matlab est un langage particulièrement spécialisé dans le support des tableaux (et par la suite des matrices).
- Un tableau est la manière la plus évidente de représenter un vecteur.
- Un vecteur 3D avec des coordonnées $[4, 1, 5, -5]$ serait représenté comme un tableau de 3 doubles disposés séquentiellement en mémoire.

4	1.5	-5
---	-----	----

Les Tableaux et les Vecteurs

- Un tableau est une collection d'objets de données du même type, généralement stockés séquentiellement en mémoire.
- Un tableau est une structure de données élémentaire.
- Presque tous les langages de programmation prennent en charge les tableaux.
- Matlab est un langage particulièrement spécialisé dans le support des tableaux (et par la suite des matrices).
- Un tableau est la manière la plus évidente de représenter un vecteur.
- Un vecteur 3D avec des coordonnées $[4, 1, 5, -5]$ serait représenté comme un tableau de 3 doubles disposés séquentiellement en mémoire.

4	1.5	-5
---	-----	----

Les Tableaux et les Vecteurs

- Un tableau est une collection d'objets de données du même type, généralement stockés séquentiellement en mémoire.
- Un tableau est une structure de données élémentaire.
- Presque tous les langages de programmation prennent en charge les tableaux.
- Matlab est un langage particulièrement spécialisé dans le support des tableaux (et par la suite des matrices).
- Un tableau est la manière la plus évidente de représenter un vecteur.
- Un vecteur 3D avec des coordonnées $[4, 1, 5, -5]$ serait représenté comme un tableau de 3 doubles disposés séquentiellement en mémoire.

4	1.5	-5
---	-----	----

Déclaration (Construction) des tableaux



Out. Prog.
Maths

Université de
MSILA

- Dans Matlab, nous pouvons construire un tableau et l'associer à un identifiant très facilement. Par exemple :

```
>> a = [4, 1.5, -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les virgules sont facultatives et peuvent être omises :

```
>> a = [4 1.5 -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les crochets indiquent à Matlab que le contenu représente un tableau.

Déclaration (Construction) des tableaux



Out. Prog.
Maths

Université de
MSILA

- Dans Matlab, nous pouvons construire un tableau et l'associer à un identifiant très facilement. Par exemple :

```
» a = [4, 1.5, -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les virgules sont facultatives et peuvent être omises :

```
» a = [4 1.5 -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les crochets indiquent à Matlab que le contenu représente un tableau.

Déclaration (Construction) des tableaux



Out. Prog.
Maths

Université de
MSILA

- Dans Matlab, nous pouvons construire un tableau et l'associer à un identifiant très facilement. Par exemple :

```
» a = [4, 1.5, -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les virgules sont facultatives et peuvent être omises :

```
» a = [4 1.5 -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les crochets indiquent à Matlab que le contenu représente un tableau.

Déclaration (Construction) des tableaux



Out. Prog.
Maths

Université de
MSILA

- Dans Matlab, nous pouvons construire un tableau et l'associer à un identifiant très facilement. Par exemple :

```
» a = [4, 1.5, -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les virgules sont facultatives et peuvent être omises :

```
» a = [4 1.5 -5]
```

```
a =
```

```
4.0000    1.5000   -5.0000
```

- Les crochets indiquent à Matlab que le contenu représente un tableau.

Les Opérations sur les tableaux

- Ce processus de construction d'un tableau implique un segment de mémoire alloué et associé au nom de la variable, et les éléments de mémoire étant mis aux valeurs spécifiées.
- Dans la plupart des langages de programmation, vous devrez déclarer un tableau et attribuer les valeurs une par une. Dans Matlab, c'est automatique.
- Vous pouvez ensuite effectuer de l'arithmétique sur des tableaux aussi simplement que possible avec des scalaires. Par exemple :

```
>> b = a*2  
b =  
      8      3     -10
```

Les Opérations sur les tableaux



Out. Prog.
Maths

Université de
MSILA

- Ce processus de construction d'un tableau implique un segment de mémoire alloué et associé au nom de la variable, et les éléments de mémoire étant mis aux valeurs spécifiées.
- Dans la plupart des langages de programmation, vous devrez déclarer un tableau et attribuer les valeurs une par une. Dans Matlab, c'est automatique.
- Vous pouvez ensuite effectuer de l'arithmétique sur des tableaux aussi simplement que possible avec des scalaires. Par exemple :

```
>> b = a*2  
b =  
8    3   -10
```

Les Opérations sur les tableaux



Out. Prog.
Maths

Université de
MSILA

- Ce processus de construction d'un tableau implique un segment de mémoire alloué et associé au nom de la variable, et les éléments de mémoire étant mis aux valeurs spécifiées.
- Dans la plupart des langages de programmation, vous devrez déclarer un tableau et attribuer les valeurs une par une. Dans Matlab, c'est automatique.
- Vous pouvez ensuite effectuer de l'arithmétique sur des tableaux aussi simplement que possible avec des scalaires. Par exemple :

```
>> b = a*2  
b =  
8    3   -10
```

Les Opérations sur les tableaux



Out. Prog.
Maths

Université de
MSILA

- Ce processus de construction d'un tableau implique un segment de mémoire alloué et associé au nom de la variable, et les éléments de mémoire étant mis aux valeurs spécifiées.
- Dans la plupart des langages de programmation, vous devrez déclarer un tableau et attribuer les valeurs une par une. Dans Matlab, c'est automatique.
- Vous pouvez ensuite effectuer de l'arithmétique sur des tableaux aussi simplement que possible avec des scalaires. Par exemple :

```
>> b = a*2  
b =  
      8      3     -10
```

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes. Par exemple :

```
>> c = b(1)
      c =
           2
```

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

```
>> b(3) = 6
      b =
           2     5     6
```

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

```
>> b(4)
      ??? Index exceeds matrix dimensions.
```

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Utilisation des éléments de tableau

- Vous pouvez extraire des valeurs individuelles d'un tableau en spécifiant l'index dans le tableau à l'aide de parenthèses rondes.

Par exemple :

$$\gg c = b(1)$$

$$c =$$

2

- Vous pouvez également attribuer de nouvelles valeurs à des éléments individuels d'un tableau. Par exemple :

$$\gg b(3) = 6$$

$$b =$$

2 5 6

- Dans Matlab, l'indice du premier élément d'un tableau est toujours 1. Cela diffère des langages tels que C ou Java où l'index du premier élément est toujours 0.
- Matlab garde une trace de la taille des tableaux et s'assure que vous n'essayez pas d'aller au-delà de leurs limites. Par exemple :

$$\gg b(4)$$

??? Index exceeds matrix dimensions.

Les Matrices



Out. Prog.
Maths

Université de
MSILA

- Un tableau est une collection d'objets de données du même type.
- Les objets de données du tableau peuvent eux-mêmes être des tableaux.
- Une matrice est généralement représentée par un tableau de tableaux ou un tableau 2D. Matlab prend en charge les matrices de la même manière qu'il prend en charge les vecteurs.
- Matlab utilise l'opérateur point-virgule (;) pour distinguer les différentes lignes d'une matrice. Par exemple :

```
>> a = [1 2 3; 4 5 6]
```

```
a =
```

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

Les Matrices



Out. Prog.
Maths

Université de
MSILA

- Un tableau est une collection d'objets de données du même type.
- Les objets de données du tableau peuvent eux-mêmes être des tableaux.
- Une matrice est généralement représentée par un tableau de tableaux ou un tableau 2D. Matlab prend en charge les matrices de la même manière qu'il prend en charge les vecteurs.
- Matlab utilise l'opérateur point-virgule (;) pour distinguer les différentes lignes d'une matrice. Par exemple :

```
>> a = [1 2 3; 4 5 6]
```

```
a =
```

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

Les Matrices



Out. Prog.
Maths

Université de
MSILA

- Un tableau est une collection d'objets de données du même type.
- Les objets de données du tableau peuvent eux-mêmes être des tableaux.
- Une matrice est généralement représentée par un tableau de tableaux ou un tableau 2D. Matlab prend en charge les matrices de la même manière qu'il prend en charge les vecteurs.
- Matlab utilise l'opérateur point-virgule (;) pour distinguer les différentes lignes d'une matrice. Par exemple :

```
>> a = [1 2 3; 4 5 6]
```

```
a =
```

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

Les Matrices



Out. Prog.
Maths

Université de
MSILA

- Un tableau est une collection d'objets de données du même type.
- Les objets de données du tableau peuvent eux-mêmes être des tableaux.
- Une matrice est généralement représentée par un tableau de tableaux ou un tableau 2D. Matlab prend en charge les matrices de la même manière qu'il prend en charge les vecteurs.
- Matlab utilise l'opérateur point-virgule (;) pour distinguer les différentes lignes d'une matrice. Par exemple :

```
>> a = [1 2 3; 4 5 6]
```

```
a =
```

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

Les Matrices



Out. Prog.
Maths

Université de
MSILA

- Un tableau est une collection d'objets de données du même type.
- Les objets de données du tableau peuvent eux-mêmes être des tableaux.
- Une matrice est généralement représentée par un tableau de tableaux ou un tableau 2D. Matlab prend en charge les matrices de la même manière qu'il prend en charge les vecteurs.
- Matlab utilise l'opérateur point-virgule (;) pour distinguer les différentes lignes d'une matrice. Par exemple :

```
>> a = [1 2 3; 4 5 6]
```

```
a =
```

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

Tout dans Matlab est une matrice



Out. Prog.
Maths

Université de
MSILA

- Matlab permet également de saisir des lignes sur différentes lignes.
- Une fois qu'un tableau est démarré par un crochet (**[]**), Matlab suppose qu'une nouvelle ligne signifie une nouvelle ligne de la matrice. Par exemple :

```
>> a = [1 2 3  
4 5 6]; % Une matrice composée de deux lignes.
```
- En ce qui concerne Matlab, tout est une matrice !
- Un vecteur est une matrice $1 \times N$ (ou $N \times 1$); un scalaire est une matrice 1×1 .

Tout dans Matlab est une matrice



Out. Prog.
Maths

Université de
MSILA

- Matlab permet également de saisir des lignes sur différentes lignes.
- Une fois qu'un tableau est démarré par un crochet (**[]**), Matlab suppose qu'une nouvelle ligne signifie une nouvelle ligne de la matrice. Par exemple :

```
>> a = [1 2 3  
       4 5 6]; % Une matrice composée de deux lignes.
```

- En ce qui concerne Matlab, tout est une matrice !
- Un vecteur est une matrice $1 \times N$ (ou $N \times 1$); un scalaire est une matrice 1×1 .

Tout dans Matlab est une matrice



Out. Prog.
Maths

Université de
MSILA

- Matlab permet également de saisir des lignes sur différentes lignes.
- Une fois qu'un tableau est démarré par un crochet (**[]**), Matlab suppose qu'une nouvelle ligne signifie une nouvelle ligne de la matrice. Par exemple :

```
» a = [1 2 3  
      4 5 6]; % Une matrice composée de deux lignes.
```
- En ce qui concerne Matlab, tout est une matrice !
- Un vecteur est une matrice $1 \times N$ (ou $N \times 1$); un scalaire est une matrice 1×1 .

Tout dans Matlab est une matrice



Out. Prog.
Maths

Université de
MSILA

- Matlab permet également de saisir des lignes sur différentes lignes.
- Une fois qu'un tableau est démarré par un crochet (**[]**), Matlab suppose qu'une nouvelle ligne signifie une nouvelle ligne de la matrice. Par exemple :

```
» a = [1 2 3  
      4 5 6]; % Une matrice composée de deux lignes.
```
- En ce qui concerne Matlab, tout est une matrice !
- Un vecteur est une matrice $1 \times N$ (ou $N \times 1$); un scalaire est une matrice 1×1 .

Tout dans Matlab est une matrice



Out. Prog.
Maths

Université de
MSILA

- Matlab permet également de saisir des lignes sur différentes lignes.
- Une fois qu'un tableau est démarré par un crochet (**[]**), Matlab suppose qu'une nouvelle ligne signifie une nouvelle ligne de la matrice. Par exemple :

```
» a = [1 2 3  
      4 5 6]; % Une matrice composée de deux lignes.
```
- En ce qui concerne Matlab, tout est une matrice !
- Un vecteur est une matrice $1 \times N$ (ou $N \times 1$); un scalaire est une matrice 1×1 .

Les Opérateurs matriciels et vectoriels

- Les opérateurs mathématiques standard peuvent être appliqués aux vecteurs et aux matrices. Matlab gère automatiquement tous les détails.

- Par exemple, supposons que nous ayons deux matrices définies comme :

```
>> a = [1 2; 3 4];
```

```
>> b = [5 6; 7 8];
```

- L'opérateur de transposition change les lignes et les colonnes d'une matrice.
- L'opérateur de transposition est désigné par le symbole d'apostrophe unique (').

```
>> a'
```

```
ans =
```

```
1    3
```

```
2    4
```

- La transposition a une priorité plus élevée que la multiplication.

Les Opérateurs matriciels et vectoriels

- Les opérateurs mathématiques standard peuvent être appliqués aux vecteurs et aux matrices. Matlab gère automatiquement tous les détails.

- Par exemple, supposons que nous ayons deux matrices définies comme :

```
>> a = [1 2; 3 4];
```

```
>> b = [5 6; 7 8];
```

- L'opérateur de transposition change les lignes et les colonnes d'une matrice.

- L'opérateur de transposition est désigné par le symbole d'apostrophe unique (').

```
>> a'
```

```
ans =
```

```
1    3
```

```
2    4
```

- La transposition a une priorité plus élevée que la multiplication.

Les Opérateurs matriciels et vectoriels

- Les opérateurs mathématiques standard peuvent être appliqués aux vecteurs et aux matrices. Matlab gère automatiquement tous les détails.
- Par exemple, supposons que nous ayons deux matrices définies comme :

```
>> a = [1 2; 3 4];
```

```
>> b = [5 6; 7 8];
```

- L'opérateur de transposition change les lignes et les colonnes d'une matrice.
- L'opérateur de transposition est désigné par le symbole d'apostrophe (').

```
>> a'
```

```
ans =
```

```
1    3
```

```
2    4
```

- La transposition a une priorité plus élevée que la multiplication.

Les Opérateurs matriciels et vectoriels

- Les opérateurs mathématiques standard peuvent être appliqués aux vecteurs et aux matrices. Matlab gère automatiquement tous les détails.
- Par exemple, supposons que nous ayons deux matrices définies comme :

```
>> a = [1 2; 3 4];
```

```
>> b = [5 6; 7 8];
```

- L'opérateur de transposition change les lignes et les colonnes d'une matrice.
- L'opérateur de transposition est désigné par le symbole d'apostrophe (').

```
>> a'
```

```
ans =
```

```
1    3
```

```
2    4
```

- La transposition a une priorité plus élevée que la multiplication.

Les Opérateurs matriciels et vectoriels

- Les opérateurs mathématiques standard peuvent être appliqués aux vecteurs et aux matrices. Matlab gère automatiquement tous les détails.
- Par exemple, supposons que nous ayons deux matrices définies comme :

```
>> a = [1 2; 3 4];
```

```
>> b = [5 6; 7 8];
```

- L'opérateur de transposition change les lignes et les colonnes d'une matrice.
- L'opérateur de transposition est désigné par le symbole d'apostrophe unique (').

```
>> a'
```

```
ans =
```

```
1    3
```

```
2    4
```

- La transposition a une priorité plus élevée que la multiplication.

Addition et soustraction



Out. Prog.
Maths

Université de
MSILA

- L'addition et la soustraction des matrices sont identiques à l'algèbre linéaire. Par exemple :

$$\gg c = a + b$$

$$c =$$

$$\begin{array}{cc} 6 & 8 \\ 10 & 12 \end{array}$$

$$\gg c = a + 2$$

$$c =$$

$$\begin{array}{cc} 3 & 4 \\ 5 & 6 \end{array}$$

- Pour que l'addition ou la soustraction des matrices fonctionne, les dimensions des deux matrices doivent correspondre.

Addition et soustraction



Out. Prog.
Maths

Université de
MSILA

- L'addition et la soustraction des matrices sont identiques à l'algèbre linéaire. Par exemple :

$$\gg c = a + b$$

$$c =$$

$$\begin{array}{cc} 6 & 8 \\ 10 & 12 \end{array}$$

$$\gg c = a + 2$$

$$c =$$

$$\begin{array}{cc} 3 & 4 \\ 5 & 6 \end{array}$$

- Pour que l'addition ou la soustraction des matrices fonctionne, les dimensions des deux matrices doivent correspondre.

Multiplication Matricielle



Out. Prog.
Maths

Université de
MSILA

- La multiplication matricielle est définie comme dans l'algèbre linéaire standard. Par exemple :

$$\gg a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}; b = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix};$$

$$\gg c = a * b$$

c =

$$\begin{bmatrix} 19 & 22 \end{bmatrix}$$

$$\begin{bmatrix} 43 & 50 \end{bmatrix}$$

$$\gg d = a * 3$$

d =

$$\begin{bmatrix} 3 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 9 & 12 \end{bmatrix}$$

- Pour que la multiplication matricielle fonctionne, le nombre de colonnes de la première matrice doit correspondre au nombre de lignes de la deuxième matrice.

Multiplication Matricielle

- La multiplication matricielle est définie comme dans l'algèbre linéaire standard. Par exemple :

$$\gg a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}; b = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix};$$

$$\gg c = a * b$$

$$c =$$

$$\begin{matrix} 19 & 22 \end{matrix}$$

$$\begin{matrix} 43 & 50 \end{matrix}$$

$$\gg d = a * 3$$

$$d =$$

$$\begin{matrix} 3 & 6 \end{matrix}$$

$$\begin{matrix} 9 & 12 \end{matrix}$$

- Pour que la multiplication matricielle fonctionne, le nombre de colonnes de la première matrice doit correspondre au nombre de lignes de la deuxième matrice.

Multiplication Matricielle



Out. Prog.
Maths

Université de
MSILA

- La multiplication matricielle est définie comme dans l'algèbre linéaire standard. Par exemple :

$$\gg a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}; b = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix};$$

$$\gg c = a * b$$

$$c =$$

$$\begin{matrix} 19 & 22 \end{matrix}$$

$$\begin{matrix} 43 & 50 \end{matrix}$$

$$\gg d = a * 3$$

$$d =$$

$$\begin{matrix} 3 & 6 \end{matrix}$$

$$\begin{matrix} 9 & 12 \end{matrix}$$

- Pour que la multiplication matricielle fonctionne, le nombre de colonnes de la première matrice doit correspondre au nombre de lignes de la deuxième matrice.

Multiplication ponctuelle



Out. Prog.
Maths

Université de
MSILA

- Un certain nombre d'opérateurs spéciaux sont associés aux matrices et aux vecteurs, dont beaucoup sont uniques à Matlab.
- L'opérateur `.*` effectue une multiplication point par point sur chaque paire correspondante d'éléments de deux matrices (parfois appelée «multiplication des tableaux»). Par exemple :

```
>> c = a.*b
```

```
c =
```

```
5    12  
21   32
```

Multiplication ponctuelle



Out. Prog.
Maths

Université de
MSILA

- Un certain nombre d'opérateurs spéciaux sont associés aux matrices et aux vecteurs, dont beaucoup sont uniques à Matlab.
- L'opérateur `.*` effectue une multiplication point par point sur chaque paire correspondante d'éléments de deux matrices (parfois appelée «multiplication des tableaux»). Par exemple :

```
>> c = a.*b
```

```
c =
```

```
5    12  
21   32
```

Multiplication ponctuelle



Out. Prog.
Maths

Université de
MSILA

- Un certain nombre d'opérateurs spéciaux sont associés aux matrices et aux vecteurs, dont beaucoup sont uniques à Matlab.
- L'opérateur `.*` effectue une multiplication point par point sur chaque paire correspondante d'éléments de deux matrices (parfois appelée «multiplication des tableaux»). Par exemple :

```
>> c = a.*b
```

```
c =
```

```
5    12  
21   32
```

Division et Exponentiation ponctuelle



Out. Prog.
Maths

Université de
MSILA

- L'opérateur ./ effectue une division point par point sur chaque paire correspondante d'éléments de deux matrices. Par exemple :

$$\gg c = a ./ b$$

c =

0.2000	0.3333
0.4286	0.5000

- L'opérateur .^ effectue une exponentiation point par point sur chaque paire correspondante d'éléments de deux matrices. Par exemple :

$$\gg c = a.^b$$

c =

1	64
2187	65536

Division et Exponentiation ponctuelle



Out. Prog.
Maths

Université de
MSILA

- L'opérateur `./` effectue une division point par point sur chaque paire correspondante d'éléments de deux matrices. Par exemple :

$$\gg c = a ./ b$$

c =

0.2000	0.3333
0.4286	0.5000

- L'opérateur `.^` effectue une exponentiation point par point sur chaque paire correspondante d'éléments de deux matrices. Par exemple :

$$\gg c = a .^ b$$

c =

1	64
2187	65536

Division Matricielle

- La division matricielle implique la résolution des inverses matriciels. Matlab gère cela automatiquement !
- Notez que parce-que la multiplication matricielle n'est pas commutative, nous avons besoin du concept de division à gauche et à droite.
- La division à droite est post-multiplication par l'inverse d'une matrice :

$$\gg c = a / b \% c = a * b^{-1}$$

c =

3 -2

2 -1

- La division à gauche est pre-multiplication par l'inverse d'une matrice :

$$\gg c = a \backslash b \% c = a^{-1} * b$$

c =

-3 -4

4 5

Division Matricielle

- La division matricielle implique la résolution des inverses matriciels. Matlab gère cela automatiquement !
- Notez que parce-que la multiplication matricielle n'est pas commutative, nous avons besoin du concept de division à gauche et à droite.
- La division à droite est post-multiplication par l'inverse d'une matrice :

$$\gg c = a / b \% c = a * b^{-1}$$

c =

3 -2

2 -1

- La division à gauche est pre-multiplication par l'inverse d'une matrice :

$$\gg c = a \backslash b \% c = a^{-1} * b$$

c =

-3 -4

4 5

Division Matricielle

- La division matricielle implique la résolution des inverses matriciels. Matlab gère cela automatiquement !
- Notez que parce-que la multiplication matricielle n'est pas commutative, nous avons besoin du concept de division à gauche et à droite.
- La division à droite est post-multiplication par l'inverse d'une matrice :

$$\gg c = a / b \% c = a * b^{-1}$$

c =

3 -2

2 -1

- La division à gauche est pre-multiplication par l'inverse d'une matrice :

$$\gg c = a \backslash b \% c = a^{-1} * b$$

c =

-3 -4

4 5

Division Matricielle

- La division matricielle implique la résolution des inverses matriciels. Matlab gère cela automatiquement !
- Notez que parce-que la multiplication matricielle n'est pas commutative, nous avons besoin du concept de division à gauche et à droite.
- La division à droite est post-multiplication par l'inverse d'une matrice :

$$\gg c = a / b \% c = a * b^{-1}$$

c =

$$\begin{matrix} 3 & -2 \\ 2 & -1 \end{matrix}$$

$$\begin{matrix} 3 & -2 \\ 2 & -1 \end{matrix}$$

- La division à gauche est pre-multiplication par l'inverse d'une matrice :

$$\gg c = a \backslash b \% c = a^{-1} * b$$

c =

$$\begin{matrix} -3 & -4 \\ 4 & 5 \end{matrix}$$

$$\begin{matrix} -3 & -4 \\ 4 & 5 \end{matrix}$$

Constructeurs de tableaux - L'opérateur deux-points

- Il est facile de construire de petits tableaux en spécifiant explicitement tous les éléments, mais ce n'est pas pratique pour les grands tableaux.
- Matlab fournit l'opérateur **deux-points** (`:`) pour construire des séquences de valeurs.
- L'opérateur **deux-points** (`:`) produit un tableau équivalent aux éléments d'une séquence arithmétique.
- Les séquences arithmétiques sont définies en fonction de la première valeur de la série, de l'incrément entre les valeurs successives et de la dernière valeur de la série.
- La syntaxe pour créer un tableau à l'aide de l'opérateur **deux-points** (`:`) est la suivante :

tableau = premier : incrément : dernier

Constructeurs de tableaux - L'opérateur deux-points



Out. Prog.
Maths

Université de
MSILA

- Il est facile de construire de petits tableaux en spécifiant explicitement tous les éléments, mais ce n'est pas pratique pour les grands tableaux.
- Matlab fournit l'opérateur **deux-points (:)** pour construire des séquences de valeurs.
- L'opérateur **deux-points (:)** produit un tableau équivalent aux éléments d'une séquence arithmétique.
- Les séquences arithmétiques sont définies en fonction de la première valeur de la série, de l'incrément entre les valeurs successives et de la dernière valeur de la série.
- La syntaxe pour créer un tableau à l'aide de l'opérateur **deux-points (:)** est la suivante :
tableau = premier : incrément : dernier

Constructeurs de tableaux - L'opérateur deux-points



Out. Prog.
Maths

Université de
MSILA

- Il est facile de construire de petits tableaux en spécifiant explicitement tous les éléments, mais ce n'est pas pratique pour les grands tableaux.
- Matlab fournit l'opérateur **deux-points (:)** pour construire des séquences de valeurs.
- L'opérateur **deux-points (:)** produit un tableau équivalent aux éléments d'une séquence arithmétique.
- Les séquences arithmétiques sont définies en fonction de la première valeur de la série, de l'incrément entre les valeurs successives et de la dernière valeur de la série.
- La syntaxe pour créer un tableau à l'aide de l'opérateur **deux-points (:)** est la suivante :
tableau = premier : incrément : dernier

Constructeurs de tableaux - L'opérateur deux-points



Out. Prog.
Maths

Université de
MSILA

- Il est facile de construire de petits tableaux en spécifiant explicitement tous les éléments, mais ce n'est pas pratique pour les grands tableaux.
- Matlab fournit l'opérateur **deux-points (:)** pour construire des séquences de valeurs.
- L'opérateur **deux-points (:)** produit un tableau équivalent aux éléments d'une séquence arithmétique.
- Les séquences arithmétiques sont définies en fonction de la première valeur de la série, de l'incrément entre les valeurs successives et de la dernière valeur de la série.
- La syntaxe pour créer un tableau à l'aide de l'opérateur **deux-points (:)** est la suivante :
tableau = premier : incrément : dernier

Constructeurs de tableaux - L'opérateur deux-points



Out. Prog.
Maths

Université de
MSILA

- Il est facile de construire de petits tableaux en spécifiant explicitement tous les éléments, mais ce n'est pas pratique pour les grands tableaux.
- Matlab fournit l'opérateur **deux-points (:)** pour construire des séquences de valeurs.
- L'opérateur **deux-points (:)** produit un tableau équivalent aux éléments d'une séquence arithmétique.
- Les séquences arithmétiques sont définies en fonction de la première valeur de la série, de l'incrément entre les valeurs successives et de la dernière valeur de la série.
- La syntaxe pour créer un tableau à l'aide de l'opérateur **deux-points (:)** est la suivante :
$$\text{tableau} = \text{premier} : \text{incrément} : \text{dernier}$$

L'opérateur deux-points (:)



Out. Prog.
Maths

Université de
MSILA

- Par exemple :

```
>>> x = 3 : 2 : 11
```

```
x =
```

```
    3     5     7     9    11
```

- Si l'incrément est 1, il peut être omis. Par exemple :

```
>>> x = 1 : 5
```

```
x =
```

```
    1     2     3     4     5
```

- L'opérateur deux-points est extrêmement utile, non seulement pour la création de tableaux, mais aussi pour le contrôle des boucles.

L'opérateur deux-points (:)



Out. Prog.
Maths

Université de
MSILA

- Par exemple :

```
>>> x = 3 : 2 : 11
```

```
x =
```

```
3    5    7    9   11
```

- Si l'incrément est 1, il peut être omis. Par exemple :

```
>>> x = 1 : 5
```

```
x =
```

```
1    2    3    4    5
```

- L'opérateur deux-points est extrêmement utile, non seulement pour la création de tableaux, mais aussi pour le contrôle des boucles.

L'opérateur deux-points (:)



Out. Prog.
Maths

Université de
MSILA

- Par exemple :

```
>>> x = 3 : 2 : 11
```

```
x =
```

```
3    5    7    9   11
```

- Si l'incrément est 1, il peut être omis. Par exemple :

```
>>> x = 1 : 5
```

```
x =
```

```
1    2    3    4    5
```

- L'opérateur deux-points est extrêmement utile, non seulement pour la création de tableaux, mais aussi pour le contrôle des boucles.

Sous-Tableaux



Out. Prog.
Maths

Université de
MSILA

- En plus de sélectionner des éléments individuels à partir de tableaux, Matlab permet la sélection de sections d'un tableau.
Par exemple :

```
>> a = [10 :-1 : 1]
```

```
a =
```

```
10    9    8    7    6    5    4    3    2    1
```

```
>> a(2 :5)
```

```
ans =
```

```
9    8    7    6
```

- On peut également attribuer des valeurs aux sous-tableaux.

```
>> a(1 :3) = [-5 -2 -4]
```

```
a =
```

```
-5    -2    -4    7    6    5    4    3    2    1
```

- La taille du tableau attribué doit correspondre à la taille du tableau sélectionné.

Sous-Tableaux



Out. Prog.
Maths

Université de
MSILA

- En plus de sélectionner des éléments individuels à partir de tableaux, Matlab permet la sélection de sections d'un tableau.
Par exemple :

```
>> a = [10 :-1 : 1]
```

```
a =
```

```
10    9    8    7    6    5    4    3    2    1
```

```
>> a(2 :5)
```

```
ans =
```

```
9    8    7    6
```

- On peut également attribuer des valeurs aux sous-tableaux.

```
>> a(1 :3) = [-5 -2 -4]
```

```
a =
```

```
-5    -2    -4    7    6    5    4    3    2    1
```

- La taille du tableau attribué doit correspondre à la taille du tableau sélectionné.

Sous-Tableaux



Out. Prog.
Maths

Université de
MSILA

- En plus de sélectionner des éléments individuels à partir de tableaux, Matlab permet la sélection de sections d'un tableau.
Par exemple :

```
>> a = [10 :-1 : 1]
```

```
a =
```

```
10    9    8    7    6    5    4    3    2    1
```

```
>> a(2 :5)
```

```
ans =
```

```
9    8    7    6
```

- On peut également attribuer des valeurs aux sous-tableaux.

```
>> a(1 :3) = [-5 -2 -4]
```

```
a =
```

```
-5    -2    -4    7    6    5    4    3    2    1
```

- La taille du tableau attribué doit correspondre à la taille du tableau sélectionné.

Sous-Tableaux



Out. Prog.
Maths

Université de
MSILA

- En plus de sélectionner des éléments individuels à partir de tableaux, Matlab permet la sélection de sections d'un tableau.
Par exemple :

```
>> a = [10 :-1 : 1]
```

```
a =
```

```
10    9    8    7    6    5    4    3    2    1
```

```
>> a(2 :5)
```

```
ans =
```

```
9    8    7    6
```

- On peut également attribuer des valeurs aux sous-tableaux.

```
>> a(1 :3) = [-5 -2 -4]
```

```
a =
```

```
-5    -2    -4    7    6    5    4    3    2    1
```

- La taille du tableau attribué doit correspondre à la taille du tableau sélectionné.

Extraction de données à partir de tableaux 2D



Out. Prog.
Maths

Université de
MSILA

Les opérations de sélection de matrices sont étendues aux tableaux 2D de manière naturelle.

```
>>> a = [1 2 3;4 5 6;7 8 9]; % Define a 2D array
```

```
>>> a(3, 2); % Get the value at row 3, column 2.
```

```
ans =
```

```
8
```

```
>>> a(2, 1 :3); % Get row 2, columns 1 to 3.
```

```
ans =
```

```
4     5     6
```

```
>>> a(2 :3, 1 :2); % Get the values from rows 2-3 in columns 1-2.
```

```
ans =
```

```
4     5
```

```
7     8
```

Extraction de données à partir de tableaux 2D



Out. Prog.
Maths

Université de
MSILA

Les opérations de sélection de matrices sont étendues aux tableaux 2D de manière naturelle.

```
>>> a = [1 2 3;4 5 6;7 8 9]; % Define a 2D array
```

```
>>> a(3, 2); % Get the value at row 3, column 2.
```

```
ans =
```

```
8
```

```
>>> a(2, 1 :3); % Get row 2, columns 1 to 3.
```

```
ans =
```

```
4     5     6
```

```
>>> a(2 :3, 1 :2); % Get the values from rows 2-3 in columns 1-2.
```

```
ans =
```

```
4     5
```

```
7     8
```

Extraction de données à partir de tableaux 2D



Out. Prog.
Maths

Université de
MSILA

Les opérations de sélection de matrices sont étendues aux tableaux 2D de manière naturelle.

```
>>> a = [1 2 3;4 5 6;7 8 9]; % Define a 2D array
```

```
>>> a(3, 2); % Get the value at row 3, column 2.
```

```
ans =
```

```
8
```

```
>>> a(2, 1 :3); % Get row 2, columns 1 to 3.
```

```
ans =
```

```
4 5 6
```

```
>>> a(2 :3, 1 :2); % Get the values from rows 2-3 in columns 1-2.
```

```
ans =
```

```
4 5
```

```
7 8
```

Extraction de données à partir de tableaux 2D



Out. Prog.
Maths

Université de
MSILA

Les opérations de sélection de matrices sont étendues aux tableaux 2D de manière naturelle.

```
>>> a = [1 2 3;4 5 6;7 8 9]; % Define a 2D array
```

```
>>> a(3, 2); % Get the value at row 3, column 2.
```

```
ans =
```

```
8
```

```
>>> a(2, 1 :3); % Get row 2, columns 1 to 3.
```

```
ans =
```

```
4    5    6
```

```
>>> a(2 :3, 1 :2); % Get the values from rows 2-3 in columns 1-2.
```

```
ans =
```

```
4    5
```

```
7    8
```

Extraction de données à partir de tableaux 2D



Out. Prog.
Maths

Université de
MSILA

Les opérations de sélection de matrices sont étendues aux tableaux 2D de manière naturelle.

```
>>> a = [1 2 3;4 5 6;7 8 9]; % Define a 2D array
```

```
>>> a(3, 2); % Get the value at row 3, column 2.
```

```
ans =
```

```
8
```

```
>>> a(2, 1 :3); % Get row 2, columns 1 to 3.
```

```
ans =
```

```
4    5    6
```

```
>>> a(2 :3, 1 :2); % Get the values from rows 2-3 in columns 1-2.
```

```
ans =
```

```
4    5
```

```
7    8
```

Extraction de données à partir de tableaux 2D



Out. Prog.
Maths

Université de
MSILA

Les opérations de sélection de matrices sont étendues aux tableaux 2D de manière naturelle.

```
>>> a = [1 2 3;4 5 6;7 8 9]; % Define a 2D array
```

```
>>> a(3, 2); % Get the value at row 3, column 2.
```

```
ans =
```

```
8
```

```
>>> a(2, 1 :3); % Get row 2, columns 1 to 3.
```

```
ans =
```

```
4     5     6
```

```
>>> a(2 :3, 1 :2); % Get the values from rows 2-3 in columns 1-2.
```

```
ans =
```

```
4     5
```

```
7     8
```

Extraction de toutes les lignes ou colonnes

Out. Prog.
Maths

Université de
MSILA

Si vous souhaitez extraire toutes les lignes (ou toutes les colonnes) d'une matrice, vous pouvez utiliser l'opérateur deux-points vides pour spécifier la ligne ou la colonne.

```
>>> a(:, 2) % Get the values from every row in column 2
```

```
ans =
```

```
2
```

```
5
```

```
8
```

```
>>> a(2 :3, :) % Get the values from rows 2-3 in all columns
```

```
ans =
```

```
4    5    6
```

```
7    8    9
```

Extraction de toutes les lignes ou colonnes



Out. Prog.
Maths

Université de
MSILA

Si vous souhaitez extraire toutes les lignes (ou toutes les colonnes) d'une matrice, vous pouvez utiliser l'opérateur deux-points vides pour spécifier la ligne ou la colonne.

```
>>> a(:, 2) % Get the values from every row in column 2
```

```
ans =
```

```
2
```

```
5
```

```
8
```

```
>>> a(2 :3, :) % Get the values from rows 2-3 in all columns
```

```
ans =
```

```
4    5    6
```

```
7    8    9
```

Concaténation des matrices



Out. Prog.
Maths

Université de
MSILA

Matlab a une syntaxe très pratique pour concaténer des matrices, il suffit de coller les matrices côte à côte, ou les unes sur les autres, dans un ensemble de crochets englobants.

```
>> a = [1 2 3];
```

```
>> b = [a 7 8]; % Concatenate 7 and 8 onto the end of a.
```

```
b =
```

```
    1    2    3    7    8
```

```
>> a = [a a(1 :2); b]
```

```
a =
```

```
    1    2    3    1    2
```

```
    1    2    3    7    8
```

Concaténation des matrices



Out. Prog.
Maths

Université de
MSILA

Matlab a une syntaxe très pratique pour concaténer des matrices, il suffit de coller les matrices côte à côte, ou les unes sur les autres, dans un ensemble de crochets englobants.

```
>> a = [1 2 3];
```

```
>> b = [a 7 8]; % Concatenate 7 and 8 onto the end of a.
```

```
b =
```

```
1     2     3     7     8
```

```
>> a = [a a(1 :2); b]
```

```
a =
```

```
1     2     3     1     2
```

```
1     2     3     7     8
```

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Array Right Division	$a./b$	Element-by-element division of a and b : $a(i, j)/b(i, j)$.
Array Left Division	$a.\backslash b$	Element-by-element division of a and b : $b(i, j)/a(i, j)$.
Array Multiplication	$a.* b$	Element-by-element multiplication of a and b : $a(i, j) * b(i, j)$.
Array Exponentiation	$a.^b$	Element-by-element exponentiation of a and b : $a(i, j)^b(i, j)$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Array Right Division	<code>a./b</code>	Element-by-element division of a and b : $a(i, j)/b(i, j)$.
Array Left Division	<code>a.\b</code>	Element-by-element division of a and b : $b(i, j)/a(i, j)$.
Array Multiplication	<code>a.* b</code>	Element-by-element multiplication of a and b : $a(i, j) * b(i, j)$.
Array Exponentiation	<code>a.^b</code>	Element-by-element exponentiation of a and b : $a(i, j)^b(i, j)$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Array Right Division	$a./b$	Element-by-element division of a and b : $a(i, j)/b(i, j)$.
Array Left Division	$a.\backslash b$	Element-by-element division of a and b : $b(i, j)/a(i, j)$.
Array Multiplication	$a.* b$	Element-by-element multiplication of a and b : $a(i, j) * b(i, j)$.
Array Exponentiation	$a.^b$	Element-by-element exponentiation of a and b : $a(i, j)^b(i, j)$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Array Right Division	$a./b$	Element-by-element division of a and b : $a(i, j)/b(i, j)$.
Array Left Division	$a.\backslash b$	Element-by-element division of a and b : $b(i, j)/a(i, j)$.
Array Multiplication	$a.* b$	Element-by-element multiplication of a and b : $a(i, j) * b(i, j)$.
Array Exponentiation	$a.^b$	Element-by-element exponentiation of a and b : $a(i, j)^b(i, j)$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Addition (Subtraction)	$a + b$ $a - b$	Array and Matrix addition (Subtraction) are identical.
Matrix Multiplication	$a * b$	Matrix multiplication of a and b.
Matrix Right Division	a / b	Matrix division is defined by : $a * b^{-1}$.
Matrix Left Division	$a \setminus b$	Matrix division is defined by : $a^{-1} * b$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Addition (Subtraction)	$a + b$ $a - b$	Array and Matrix addition (Subtraction) are identical.
Matrix Multiplication	$a * b$	Matrix multiplication of a and b.
Matrix Right Division	a / b	Matrix division is defined by : $a * b^{-1}$.
Matrix Left Division	$a \setminus b$	Matrix division is defined by : $a^{-1} * b$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Addition (Subtraction)	$a + b$ $a - b$	Array and Matrix addition (Subtraction) are identical.
Matrix Multiplication	$a * b$	Matrix multiplication of a and b.
Matrix Right Division	a / b	Matrix division is defined by : $a * b^{-1}$.
Matrix Left Division	$a \setminus b$	Matrix division is defined by : $a^{-1} * b$.

Opérations sur les Tableaux et les Matrices



Out. Prog.
Maths

Université de
MSILA

Operation	Matlab Form	Comments
Addition (Subtraction)	$a + b$ $a - b$	Array and Matrix addition (Subtraction) are identical.
Matrix Multiplication	$a * b$	Matrix multiplication of a and b.
Matrix Right Division	a / b	Matrix division is defined by : $a * b^{-1}$.
Matrix Left Division	$a \setminus b$	Matrix division is defined by : $a^{-1} * b$.

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>length(A)</code>	retourne la taille d'un vecteur A
<code>linspace(a,b,n)</code>	génère un vecteur ligne de n éléments espacés linéairement entre a et b
<code>sum(x)</code>	somme des éléments du vecteur x
<code>prod(x)</code>	produit des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>length(A)</code>	retourne la taille d'un vecteur A
<code>linspace(a,b,n)</code>	génère un vecteur ligne de n éléments espacés linéairement entre a et b
<code>sum(x)</code>	somme des éléments du vecteur x
<code>prod(x)</code>	produit des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>length(A)</code>	retourne la taille d'un vecteur A
<code>linspace(a,b,n)</code>	génère un vecteur ligne de n éléments espacés linéairement entre a et b
<code>sum(x)</code>	somme des éléments du vecteur x
<code>prod(x)</code>	produit des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>length(A)</code>	retourne la taille d'un vecteur A
<code>linspace(a,b,n)</code>	génère un vecteur ligne de n éléments espacés linéairement entre a et b
<code>sum(x)</code>	somme des éléments du vecteur x
<code>prod(x)</code>	produit des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
$\max(x)$	plus grand élément du vecteur x
$\min(x)$	plus petit élément du vecteur x
$\text{mean}(x)$	moyenne des éléments du vecteur x
$\text{sort}(x)$	ordonne les éléments du vecteur x par ordre croissant
$\text{fliplr}(x)$	renverse l'ordre des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
$\max(x)$	plus grand élément du vecteur x
$\min(x)$	plus petit élément du vecteur x
$\text{mean}(x)$	moyenne des éléments du vecteur x
$\text{sort}(x)$	ordonne les éléments du vecteur x par ordre croissant
$\text{fliplr}(x)$	renverse l'ordre des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
$\max(x)$	plus grand élément du vecteur x
$\min(x)$	plus petit élément du vecteur x
$\text{mean}(x)$	moyenne des éléments du vecteur x
$\text{sort}(x)$	ordonne les éléments du vecteur x par ordre croissant
$\text{fliplr}(x)$	renverse l'ordre des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
$\max(x)$	plus grand élément du vecteur x
$\min(x)$	plus petit élément du vecteur x
$\text{mean}(x)$	moyenne des éléments du vecteur x
$\text{sort}(x)$	ordonne les éléments du vecteur x par ordre croissant
$\text{fliplr}(x)$	renverse l'ordre des éléments du vecteur x

Fonctions propres aux vecteurs



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
$\max(x)$	plus grand élément du vecteur x
$\min(x)$	plus petit élément du vecteur x
$\text{mean}(x)$	moyenne des éléments du vecteur x
$\text{sort}(x)$	ordonne les éléments du vecteur x par ordre croissant
$\text{fliplr}(x)$	renverse l'ordre des éléments du vecteur x

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>eye(n)</code>	la matrice identité (carrée de taille n)
<code>ones(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 1
<code>zeros(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 0
<code>rand(m,n)</code>	une matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.
<code>magic(n)</code>	une matrice magique de dimension n .

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>eye(n)</code>	la matrice identité (carrée de taille n)
<code>ones(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 1
<code>zeros(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 0
<code>rand(m,n)</code>	une matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.
<code>magic(n)</code>	une matrice magique de dimension n .

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>eye(n)</code>	la matrice identité (carrée de taille n)
<code>ones(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 1
<code>zeros(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 0
<code>rand(m,n)</code>	une matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.
<code>magic(n)</code>	une matrice magique de dimension n .

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>eye(n)</code>	la matrice identité (carrée de taille n)
<code>ones(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 1
<code>zeros(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 0
<code>rand(m,n)</code>	une matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.
<code>magic(n)</code>	une matrice magique de dimension n .

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>eye(n)</code>	la matrice identité (carrée de taille n)
<code>ones(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 1
<code>zeros(m,n)</code>	la matrice à m lignes et n colonnes dont tous les éléments valent 0
<code>rand(m,n)</code>	une matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.
<code>magic(n)</code>	une matrice magique de dimension n .

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>diag()</code>	permet d'extraire les éléments diagonaux
<code>triu()</code> <code>tril()</code>	permettent d'extraire les parties triangulaire supérieure et triangulaire inférieure
<code>size()</code>	permet de récupérer les dimensions d'une matrice
<code>det()</code>	déterminant d'une matrice
<code>inv()</code>	inverse d'une matrice carrée

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>diag()</code>	permet d'extraire les éléments diagonaux
<code>triu()</code> <code>tril()</code>	permettent d'extraire les parties triangulaire supérieure et triangulaire inférieure
<code>size()</code>	permet de récupérer les dimensions d'une matrice
<code>det()</code>	déterminant d'une matrice
<code>inv()</code>	inverse d'une matrice carrée

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>diag()</code>	permet d'extraire les éléments diagonaux
<code>triu()</code> <code>tril()</code>	permettent d'extraire les parties triangulaire supérieure et triangulaire inférieure
<code>size()</code>	permet de récupérer les dimensions d'une matrice
<code>det()</code>	déterminant d'une matrice
<code>inv()</code>	inverse d'une matrice carrée

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>diag()</code>	permet d'extraire les éléments diagonaux
<code>triu()</code> <code>tril()</code>	permettent d'extraire les parties triangulaire supérieure et triangulaire inférieure
<code>size()</code>	permet de récupérer les dimensions d'une matrice
<code>det()</code>	déterminant d'une matrice
<code>inv()</code>	inverse d'une matrice carrée

Fonctions propres aux matrices



Out. Prog.
Maths

Université de
MSILA

Fonction	Utilisation
<code>diag()</code>	permet d'extraire les éléments diagonaux
<code>triu()</code> <code>tril()</code>	permettent d'extraire les parties triangulaire supérieure et triangulaire inférieure
<code>size()</code>	permet de récupérer les dimensions d'une matrice
<code>det()</code>	déterminant d'une matrice
<code>inv()</code>	inverse d'une matrice carrée