University Mohamed Boudiaf of M'sila

Faculty of Technology

Department of Electrical Engineering

Date : December 24, 2020

Level : Licence 2

TP4 : Vecteurs et matrices

1 Introduction :

Comme son nom l'indique `Matlab` est spécialement conçu pour manipuler des matrices. `Matlab` reconnait et manipule les variables matricielles suivantes (pour plus de détails, utiliser le `help`) : matrices constantes à coefficients réels ou complexes (`help matrices`), matrices creuses (`help sparse`). Les fonctions `cell` et `struct` permettent de définir et manipuler des objets plus complexes (`help cell`).

La manière la plus simple d'entrer une matrice est d'utiliser une ligne explicite d'éléments. Dans la liste, les éléments sont séparés par des blancs ou des virgules, et des point virgules (;) sont utilisés pour indiquer la fin de ligne. La liste est encadrée par des crochets [ ]. Par exemple, l'instruction

» A = [1 2 3;4 5 6;7 8 9]

fournit comme résultat

A =

1 2 3

4 5 6

7 8 9

La variable A est donc une matrice de dimension 3,3. Les éléments d'une matrice peuvent être formés de n'importe quelle expression `Matlab`. Par exemple, l'instruction

» x = [-1.3 sqrt(3) (1+2+3)*4/5]

fournit

x = -1.3000 1.7321 4.8000

Une matrice avec une seule ligne ou une seule colonne est un vecteur , et une matrice 1,1 est un scalaire. Les éléments d'une matrice peuvent être référencés par leurs indices placés entre parenthèses. La commande `size(A)` fournit le nombre de lignes et le nombre de colonnes de A. `size(A)` est elle-même une matrice de taille 1,2, mémorisée si nécessaire par `[m n] = size(A)`.

Les "deux points"

On peut utiliser le deux points de différentes manières dans Matlab (voir `help colon`). Il sert fondamentalement à construire un vecteur dont les valeurs des éléments sont incrémentées

séquentiellement. Tapez par exemple

» x = 3:9

pour obtenir

x =

3 4 5 6 7 8 9

L'incrément est de 1 par défaut. Pour avoir un autre incrément, tapez des commandes telles que

» x = 1:0.5:4

» x = 6:-1:0

La plupart des fonctions `Matlab` acceptent des entrées vectorielles et produisent des sorties vectorielles. La commande

» y = sqrt(1:10)

construit un vecteur d'entiers de 1 à 10 et prend la racine carrée de chacun d'entre eux.

Essayez-la.

Autre subtilité : quel est l'effet de chacune de ces deux commandes et pourquoi?

» 1+1:5

» 1+(1:5)

Ce dernier exemple montre qu'il faut s'efforcer de taper les commandes de manière non ambigüe.

1.1 Manipulations:

Concaténations de matrices ou de vecteurs.

Les dimensions d'une matrice ou d'un vecteur peuvent être augmentées en introduisant de nouveaux éléments empruntés à une autre matrice ou un autre vecteur. Soit par exemple le vecteur x = [1 3 5].

La commande

» x = [x 6 8 10]

fournit le résultat suivant

x =

1 3 5 6 8 10

De même la commande

» y = [x;1:6]

donne

y =

1 3 5 6 8 10

1 2 3 4 5 6

Extraction d'une sous-matrice.

On peut aussi utiliser les deux points pour extraire une sous-matrice d'une matrice A. $A(:, j)$ extrait la $j$ème colonne de A. On considère successivement toutes les lignes de A et on choisit le $j$ème élément de chaque ligne.

$A(i, :)$ extrait la $i$ème ligne de A.

$A(:)$ reforme le matrice A en un seul vecteur colonne en concaténant toutes les colonnes de A.

$A(j : k)$ extrait les éléments j à k de A et les stocke dans un vecteur ligne

$A(:, j : k)$ extrait la sous-matrice de A formée des colonnes j à k.

$A(j : k, :)$ extrait la sous-matrice de A formée des lignes j à k.

$A(j : k, q : r)$ extrait la sous-matrice de A formée des éléments situés dans les lignes j à k et dans les colonnes q à r.

Ces définitions peuvent s'étendre à des pas d'incrémentation des lignes et des colonnes différents de 1.

Transposition

Le caractère spécial (') sert à désigner la transposée d'un matrice.

La commande A = [1 2 3;4 5 6;7 8 9]' produit la matrice

A =

1 4 7

2 5 8

3 6 9

Les lignes de A' sont les colonnes de A, et vice versa. Si A est une matrice complexe, A' est sa transposée conjuguée, ou transposée hermitienne. Pour obtenir une transposée non conjuguée, il faut employer les deux caractères point-prime (.').

1.2 Opérateurs:

Opérations arithmétiques

Additions +, soustractions −, multiplications ∗ , puissances ^ s'utilisent avec la syntaxe matricielle habituelle.

Attention aux dimensions : on ne peut ajouter ou soustraire que des matrices de même taille.

Il existe cependant une manière simple de soustraire le même scalaire à tous les éléments d'une

matrice. Par exemple, x = [1 2 3 4], x = x-1 produit le résultat

x =

1 2 3 4

x =

0 1 2 3

La multiplication de deux matrices n'a de sens que si leurs dimensions "internes" sont égales.

$$(A * B)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \tag{1}$$

où n est le nombre de colonnes de A et aussi le nombre de lignes de B. Essayez de taper $A = [123; 456]; B = [7; 8; 9]; A * B$. Vous devriez obtenir le résultat suivant

ans =

50

122

Fonctions de créations de matrices élémentaires

Ci-dessous, quelques exemples de fonctions qui vous permettrons dé générer facilement des matrices utiles dans la vie de tous les jours.

- `ones` matrice de 1 de la taille voulue;
- `zeros` matrice de 0 de la taille voulue;
- `eye` matrice identité de la taille voulue;
- `rand` matrice de la taille voulue dont les éléments sont le résultat d'une distribution aléatoire uniforme;
- `randn` matrice de la taille voulue dont les éléments sont le résultat d'une distribution aléatoire normale
- `magic` carré magique de taille quelconque.

Produits internes et externes : Le produit interne, ou scalaire, de deux vecteurs colonnes x et y est le scalaire défini comme le produit $x' * y$ ou, de manière équivalente $y' * x$. Par exemple, x = [1;2],y = [3;4], x'*y conduit au résultat

ans =

11

La norme hermitienne d'un vecteur est définie comme la racine carrée du produit scalaire du vecteur avec lui même. On peut aussi la calculer à l'aide de la fonction `norm`. Essayez par

exemple de calculer la norme du vecteur [1 2 3 4]. Vous devriez obtenir 5.4772.

Le produit externe, ou antiscalaire, de deux vecteurs colonnes est la matrice antiscalaire $x * y'$. De mŒme, le produit externe de deux vecteurs lignes est la matrice $x' * y$.

Tout scalaire peut être multiplié par une matrice. La multiplication se fait alors élément par élément. Ainsi, $A = [123; 456; 789]$; $A * 2$ donne le résultat suivant

ans =

2 4 6

8 10 12

14 16 18

Vérifier que 2*A donne le même résultat.

Inversions

L'inverse d'une matrice est calculée à l'aide de la fonction `inv(A)` qui n'est valide que si A est carrée. Si la matrice est singulière, ou non inversible, un message apparaît. Si vous tapez `inv(A)` avec la matrice précédente, vous devriez obtenir

Warning: Matrix is close to singular or badly scaled.

Results may be inaccurate. RCOND=2.937385e-18

ans =

1.0e+16*

0.3152 -0.6304 0.3152

-0.6304 1.2609 -0.6304

0.3152 -0.6304 0.3152

Si ce message aparaît, la matrice concernée n'est pas nécessairement singulière, mais son nombre de conditionnement `rcond(A)` est si petit que la précision du calcul de l'inverse est sujette à caution. Pour vous en convaincre, calculez les valeurs propres de A, en tapant `eig(A)`.

L'inverse d'une matrice est utilisée pour résoudre un système d'équations linéaires. Ainsi, pour résoudre le système

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & -2 & 4 \\ 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix} \tag{2}$$

vous pouvez taper A = [1 2 3;1 -2 4;0 -2 1]; b = [2;7;3]; inv(A)*b pour obtenir

ans =

1

-1

1

vérifiez que la réponse est correcte en tapant `A*ans`. Qu'observe t'on ?

Matlab offre un autre moyen de résoudre un système linéaire, basé sur l'élimination de Gauss, qui est plus rapide que le calcul d'une l'inverse. La syntaxe est A\b. Elle est valable tant que A possède le même nombre de lignes que b. Essayez.

Valeurs propres

La commande `eig(A)` affiche les valeurs propres de A . . .si A est carrée pas trop grande !

L'opérateur "**point**"

En calcul matriciel, on a souvent besoin d'effectuer une opération élément par élément. Dans matlab, ces opérations sont appelées opérations sur des réseaux ou des tableaux (array operations). Bien entendu, l'addition et la soustraction sont des opérations qui se font élément par élément. L'opération A.*B désigne la multiplication, élément par élément, des matrices A et B.

Construisez arbitrairement deux matrices 3,3, A = rand(3) et B = rand(3), et tapez successivement » A*B » A.*B Qu'observe t'on ?

Supposons que nous voulions calculer le carré de chaque élément de A. La bonne manière de spécifier ce calcul est

» A_square = A.^2

où le point indique qu'une opération doit être effectuée sur chaque élément de A. Sans ce point, A est multipliée par A suivant les rŁgles usuelles de la multiplication des matrices, ce qui conduit à un résultat totalement différent

» A^2

Essayez également

» 2.^A

Opérations matricielles

Les opérations matricielles exécutées par `Matlab` sont illustrées dans le tableau suivant:

`B = A'` La matrice `B` est égale à la matrice `A` transposée;

`E = inv(A)` La matrice `E` est égale à la matrice `A` inversée;

`C = A+B` Addition;

`D = A-B` Soustraction;

`Z = X*Y` Multiplication;

`X = A\B` Équivalent à `inv(A)*B`;

`X = B/A` Équivalent à `B*inv(A)`.