Chapitre 3 : Fonctions sur les Matrices sous Matlab

Université Alger1, Dept MI

2° année Maths, Semestre 3, 2016

Matière : Outils de Programmation 2

Contact : fodil.laib@hotmail.com

Rang et Déterminant d'une Matrice

Le rang d'une matrice est le nombre de colonnes linéairement indépendantes :

```
>> x = [1 2 ; 5 3 ];
>> rank(x)
ans =
     2
```

Si rang = nombre de colonne alors la matrice est régulière, son déterminant est non nul

```
>> det(x)
ans =
    -7
```

Et elle admet une matrice inverse :

```
>> inv(x)
ans =
   -0.4286    0.2857
    0.7143   -0.1429
```

Résolution d'un système d'équations linéaires $Ax = b$

Si det(A) est non nul, la solution est

$$x = A^{-1}b$$

Soit à résoudre le système

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

```
>> A = [2 3 ; 1 -2 ];
>> b = [8 ; -3 ]  ;
>> det(A)
ans =
   -7
```

Puisque det(A)≠0 alors inv(A) existe, et la solution sera

```
>> x = inv(A) * b
x =
     1
     2
```

Les Polynômes

Le polynôme $P(x) = x^2 - 6x + 9$ est déclaré sous Matlab comme suit :

```
>> P = [1 -6 9]
P =
     1    -6     9
```

On détermine ses racines comme suit :

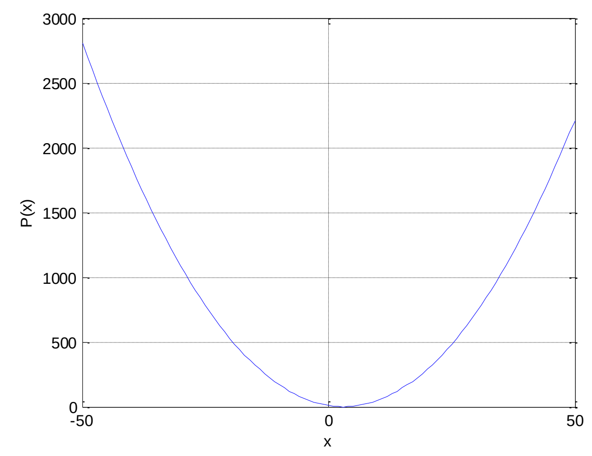
```
>> roots(P)
ans =
   3.0000 + 0.0000i
   3.0000 - 0.0000i
```

Pour l'évaluer au point x = 1 :

```
>> polyval(P,1)
ans =
     4
```

Pour dessiner la courbe du polynôme :

```
>> x = -50 : 50;
>> y = polyval(P,x);
>> plot(x,y), grid, xlabel('x'), ylabel('P(x)')
```



Multiplication et Division de Polynômes

Soient 2 polynômes $P_1(x) = x + 2$ et $P_2(x) = x^2 - 2x + 1$

```
>> P1 = [1 2];
>> P2 = [1 -2 1];
```

Leur produit est calculé par :

```
>> conv(P1,P2)
ans =
    1    0   -3    2
```

Soit le polynôme obtenu est $P_3(x) = x^3 - 3x + 2$

La division de polynôme est réalisé par :

```
>> [Q,R] = deconv(P3,P1)
Q =
    1   -2    1
R =
    0    0    0    0
```

où Q est le résultat de la division et R est le reste de la division.

La Fonction *sum*

Si A contient une seule ligne, alors *sum(A)* calcule la somme des elements de A :

```
>> A = 1:3
A =
    1    2    3
>> sum(A)
ans =
    6
```

Si A contient plusieurs lignes ($A = [a_{ij}] \in R^{m \times n}$), *sum(A)* calcule la somme de chaque colonne :

```
>> B = [A ; 4:6]
B =
    1    2    3
    4    5    6
>> sum(B)
ans =
    5    7    9
```

La Fonction *cumsum*

Si A contient une seule ligne ($A = [a_j] \in R^n$), alors

S=cumsum(A) avec

$S = [s_j] \in R^n$

et

$s_1 = a_1$,

$s_j = s_{j-1} + a_j$ où $j = 2..n$

```
>> cumsum(A)
ans =
     1     3     6
```

Si A contient plusieurs lignes, on applique *cumsum* sur chaque colonne :

```
>> C = [B ; 7 :9 ]
C =
     1     2     3
     4     5     6
     7     8     9
>> cumsum(C)
ans =
     1     2     3
     5     7     9
    12    15    18
```

Les Fonctions *prod* et *cumprod*

Même principe que les fonctions *sum* et *cumsum* respectivement en remplaçant la somme par le produit :

```
>> prod(C)
ans =
   28    80   162

>> cumprod(C)
ans =
    1     2     3
    4    10    18
   28    80   162
```

Exercice : Calculer le factoriel 5! :

```
>> prod(1:5)
ans =
   120
```

La Fonction *mean*

Si A a une seule ligne, *mean(A)* calcule la moyenne des éléments de A :

```
>> mean(A)
ans =
   2
```

Si A a plusieurs lignes, *mean(A)* calcule la moyenne de chaque colonne de A :

```
>> mean(C)
ans =
     4     5     6
```

Fonction *std()* : calcul de l'ecart type, même principe que *mean()*

La Fonctions *min*

Si A a une seule ligne, *min(A)* détermine le minimum des éléments de A :

```
>> min(A)
ans =
    1
```

Si A a plusieurs lignes, *min(A)* détermine le minimum de chaque colonne de A :

```
>> min(C)
ans =
    1    2    3
```

Fonction *max* : même principe que *min*

La Fonction *diff*

Si X a une seule ligne, la fonction *diff(X)* calcule la différence entre 2 éléments successifs de X :

```
>> D = 1 : 4
D =
    1    2    3    4

>> D = D .^ 2
D =
    1    4    9    16

>> diff(D)
ans =
   3    5    7
```

Si X a plusieurs lignes, *diff(X)* s'execute sur chaque colonne de X:

```
>> C .^ 2
ans =
    1     4     9
   16    25    36
   49    64    81

>> diff(ans)
ans =
   15    21    27
   33    39    45
```

La Fonction *linspace*

La fonction *linspace(a,b,n)* permet de générer un vecteur ligne de n points équidistants dont le premier est a et le dernier est b :

```
>> t = linspace(0,1,10)
t =
  Columns 1 through 8
       0    0.1111    0.2222    0.3333    0.4444    0.5556    0.6667    0.7778
  Columns 9 through 10
    0.8889    1.0000
```

Si le parametre *n* n'est pas spécifié, *linspace* génère par defaut 100 valeurs :

```
>> t1 = linspace(0,1);
>> length(t1)
ans =
   100
```