

## La fonction length

La fonction length renvoie le nombre de coefficient de la matrice. Ici C appartient à  $\mathcal{M}_{5,4}(\mathbb{R})$  donc possède 20 coefficients.

```
-->length(C)
```

```
ans =
```

```
20.
```

## La fonction size

La fonction size renvoie une matrice à une ligne et deux colonnes. Le premier coefficient est le nombre de lignes de la matrice à laquelle on affecte la fonction size et le deuxième coefficient est le nombre de colonnes. Comme ici C appartient à  $\mathcal{M}_{5,4}(\mathbb{R})$ , on a alors 5. 4.

```
-->size(C)
```

```
ans =
```

```
5. 4.
```

size(C,'c') renvoie le nombre de colonnes et size(C,'r') renvoie le nombre de lignes.

```
->size(C,'r')
```

```
ans =
```

```
5.
```

```
-->size(C,'c')
```

```
ans =
```

```
4.
```

## La fonction find

find(D>5) renvoie un vecteur ligne contenant **les numéros des coefficients de la matrice D** qui sont strictement supérieurs à 5

-->E=find(D>5)

E =

3. 4.

-->F=find(D>0)

F =

1. 2. 3. 4.

-->G=find(D<0)

G =

[]

## Les fonctions length et find ensemble

-->L=length(find(D>5))

L =

2.

2 est le nombre de coefficients de D qui sont strictement supérieurs à 5.

-->M=D(E)

M =

6.

7.

M est un vecteur-colonne qui contient

**les valeurs des coefficients de la matrice D** qui sont strictement supérieurs à 5

-->N=D(F)

N =

1.

2.

6.

7.

-->O=D(G)

O =

[]

## Les fonctions min, max et mean

La fonction min donne le coefficient minimum d'une matrice.

La fonction max donne le coefficient maximum d'une matrice.

La fonction mean donne la moyenne de tous les coefficients d'une matrice.

--> A=[1,5,7;8,3,4;2,9,6]		-->min(A)
A =		ans =
1. 5. 7.		1.
8. 3. 4.		
2. 9. 6.		-->mean(A)
		ans =
-->max(A)		5.
ans =		
9.		

## Les fonctions sum et prod

La fonction sum donne la somme de tous les coefficients d'une matrice.

La fonction prod donne le produit de tous les coefficients d'une matrice.

-->sum(A)		-->prod(A)
ans =		ans =
45.		362880.

## Les fonctions cumsum et cumprod (2<sup>ème</sup> année)

cumsum(A) donne la somme cumulée des coefficients de la matrice A. L'ordre est le même que celui pour compter le numéro des coefficients d'une matrice.

cumprod(A) est le produit cumulé des coefficients de la matrice A.

-->cumsum(A)		-->cumprod(A)
ans =		ans =
1. 16. 35.		1. 80. 15120.
9. 19. 39.		8. 240. 60480.
11. 28. 45.		16. 2160. 362880.

## Les fonctions liées à l'algèbre

La fonction rank donne le rang d'une matrice.

La fonction spec donne les valeurs propres d'une matrice (2<sup>ème</sup> année)

```
-->spec(A)
```

```
ans =
```

```
16.116844
```

```
- 1.116844
```

```
- 1.304D-15
```

```
-->rank(A)
```

```
ans =
```

```
2.
```

```
d=n-rank(A-lambda*eye(n,n))
```

```
donne la dimension de  $\text{Ker}(A - \lambda I)$ 
```

### III] If then else et fonction

#### 1) Apprendre à utiliser fonction

```
function [y, r, z]=h(x, p)
```

```
    y=5*x
```

```
    r=3*x+2*p
```

```
    z=p+3
```

```
endfunction
```

h est le nom de la fonction

Les antécédents sont x et p

Les images sont y, r et z. Notons que dans Scilab une fonction peut avoir plusieurs images, ce qui n'est pas le cas dans votre cours de mathématiques (c'est ce qui différencie une fonction d'une relation qui ne serait pas une fonction).

Après que la fonction ait été exécutée, elle est enregistrée et est utilisable. On peut l'appeler comme dans l'exemple suivant. Notons que l'on est pas obligé d'utiliser y, r et z en variables de sorties.

```
-->[y,r,z]=h(1,2)
```

```
z =
```

```
5.
```

```
r =
```

```
7.
```

```
y =
```

```
5.
```

```
-->[t,u,v]=h(5,3)
```

```
v =
```

```
6.
```

```
u =
```

```
21.
```

```
t =
```

```
25.
```

## 2) La fonction if then else.

### Apprendre à utiliser la fonction if then else

Si la propriété est vérifiée alors on applique ce qui est suivi de then, sinon on applique ce qui est suivi de else.

Soit la fonction  $f$  définie par :

$$f(x) = \begin{cases} -x \ln(x) & \text{si } x > 0 \\ -x \ln(-x + 1) & \text{si } x \leq 0 \end{cases}$$

Si  $x \leq 0$  alors  $f(x)$  vaut  $-x \ln(-x + 1)$ , sinon  $f(x)$  vaut  $-x \ln(x)$

On a alors la fonction :

```
function y=f(x)
    if x<=0 then y=-x*log(-x+1), else y=-x*log(x),end
endfunction
```

### Valeur absolue sans la fonction abs

Si on ne veut rien modifier dans le cas où la propriété est fausse, on n'écrit pas else.

```
y=input('Entrez une valeur de x : ');
if y<0 then y=-y;
end
disp(y, 'La valeur absolue de x est :')
```

Et en exécutant le fichier dans la console :

Entrez une valeur de x : -4

La valeur absolue de x est :  
4.

Execution done.

Entrez une valeur de x : 8

La valeur absolue de x est :  
8.

Execution done.

## La fonction racine

La fonction racine a un problème, elle n'est définie que sur  $\mathbb{R}_+$ . On doit donc poser un cas où le nombre écrit n'est pas positif ou nul.

```
function [a]=calculracine(x)
if(x>=0) then disp('La racine de x est : ')
    a=sqrt(x)
else a="Le nombre x n'a pas de racine"
end
endfunction
```

Et donc en appelant dans la console la fonction calculracine

```
-->calculracine(9)
```

La racine de x est :

a =

3.

```
-->a=calculracine(4)
```

La racine de x est :

a =

2.

```
-->a=calculracine(-4)
```

a =

Le nombre x n'a pas de racine