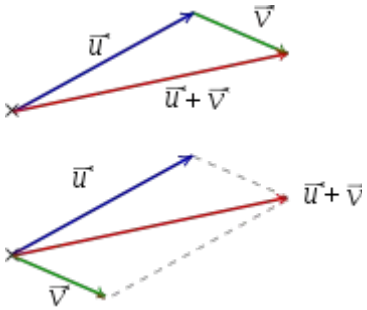
TP 3

LES VECTEURS :

En mathématiques, un vecteur est un élément d'un espace vectoriel, ce qui permet d'effectuer des opérations d'addition et de multiplication par un scalaire.

Un n-uplet (est une collection ordonnée de *n* objets, appelés « *composantes* » ou « *éléments* » ou « *termes* » du *n-uplet*.) peut constituer un exemple de vecteur, à condition qu'il appartienne à un ensemble muni des opérations adéquates. On représente fréquemment les vecteurs comme de simples n-uplets ou, graphiquement, dans le cas particulier des espaces à 1, 2 ou 3 dimensions, par des flèches.



Création d'un vecteur ligne :

En algèbre linéaire, un vecteur ligne (ou une matrice ligne) est une matrice possédant une ligne et *p* colonnes. Elle s'écrit donc :

$$\begin{bmatrix} x_1 & x_2 & \dots & x_p \end{bmatrix}$$

La création de vecteur ***a*** se fait en allant de 1 à 7 avec un pas de 2 :

Taper :

>> a=[1 3 5 7]

a =

1 3 5 7

>> b=[1,3,5,7]

b =

1 3 5 7

L'opérateur « : » d'énumération

L'opérateur d'énumération s'utilise avec la syntaxe suivante

début : incrément : fin

où l'incrément n'a pas à être précisé quand il vaut 1.

>> c=[1 :2 :7]

c =

1 3 5 7

>> c2= 1 :2 :7

c2 =

1 3 5 7

>> ind = 6:-1:1

ind =

6 5 4 3 2 1

>>temps=0:0.1:0.5

temps =

0 0.1000 0.2000 0.3000 0.4000 0.5000

>>> temps(1:3)=-2

temps =

-2.00000 -2.00000 -2.00000 0.30000 0.40000 0.50000

>>> temps(1:3)=[0.1 0.16 0.22]

temps =

0.10000 0.16000 0.22000 0.30000 0.40000 0.50000

Exercice 1

1/ créer un vecteur d1 (allant de 1 à 7) avec un pas de 1

2/ Effectuer la commande : >> d2=[1 :7]

Que peut on dire.

Création d'un vecteur colonne :

Un vecteur colonne, ou matrice colonne, est une matrice comportant *n* lignes et 1 colonne.

La transposée d'un vecteur colonne est un vecteur ligne.

Pour créer un vecteur colonne, on a la même écriture d'un vecteur ligne mais on utilise le point-virgule (au lieu d'une virgule ou du deux-points).

>> v=[2 ; 4 ; 6 ; 8]

v=

2

4

6

8

On peut aussi créer un vecteur ligne et le transposer, on ajoute pour cela un apostrophe ' .

>> w=[2 :2 :8]'

w=

2

4

6

8

Calculer les dimensions d'un vecteur :

>> s = size(v)

s =

4 1

>> s1 = size(a)

s1 =

1 4

Calculer la longueur d'un vecteur :

La plus grande dimension d'un vecteur constitue sa longueur :

>> l = length(v)

l = 4

>> l1 = length(a)

l1 = 4

Accéder à un élément :

On accède à un élément d'un vecteur en indiquant son indice entre parenthèses.

>> a4 = a(4), b3 = b(3)

a4=

7

b3=

5

On peut donc modifier un élément (une composante) d'un vecteur (ligne ou colonne) :

>> a(4)=15

a=

1 3 5 15

Opération sur les vecteurs :

Produit scalaire : il faut respecter les règles mathématiques habituelles.

>> p=a*a

??? Error using ==> *

Inner matrix dimensions must agree.

```
>> p1=a*a'

p1 = 84


>> p=v*v'

p =

4     8    12    16

8    16    24    32

12    24    36    48

16    32    48    64
```

On peut concaténer des vecteurs facilement :

```
>> c3 = [a a]

c3 =

1 3 5 7 1 3 5 7


>> c4 = [v v]

c4 =

2     2

4     4

6     6

8     8
```

Opérations élément par élément :

Les opérations suivantes se font élément par élément (Il faut que les vecteurs ont le même dimension) : l'addition (+), la soustraction (-), la division (/), la multiplication (*) et l'exposant (^).

Il est possible et même recommandé de faire des calculs vectorisés, c’est-à-dire en traitant tous les éléments d’un vecteur en même temps. Pour faire des calculs en une seule instruction sur chaque élément d’un vecteur, on met un point avant l’opérateur :

v.^n : élève chaque élément de v à la puissance n

v.^(1/2) ou sqrt(v) : calcule la racine carrée de chaque élément de v

>> sqrt(a)

ans =

1.0000 1.7321 2.2361 3.0000

Exemple :

>> v.^2

>> 1./v

Exercice 2 :

Taper les instructions suivantes et commenter les résultats :

>> a=[1 3 5 9] ;b=[0 2 4 6] ;c=a.b

>> e=[1 3 5 9] ;f=[0 2 4 6] ;g=a.*b

Fonctions mathématiques particulières, manipulation de données

On peut obtenir différents types de vecteurs en utilisant directement des fonctions Matlab. Par exemple, la fonction zero produit un vecteur de zéros, ones produit un vecteur de 1. Il faut préciser les dimensions lors de l'appel à la fonction : par exemple un vecteur ligne à 3 éléments a pour paramètres (1,3), pour indiquer 1 ligne et 3 colonnes.

>> zeros(1,3)

ans =

0 0 0

>> ones(1,8)

ans =

1 1 1 1 1 1 1 1

Les fonctions mathématiques élémentaires s'appliquent directement aux vecteurs, en opérant sur chacun des éléments, comme par exemple les fonctions exp, log, tan ou sqrt utilisées précédemment.

Soit x un vecteur, on peut lui appliquer les fonctions suivantes :

- **max(x)** : Plus grand élément.
- **min(x)** : Plus petit élément.
- **sum(x)** : Somme des éléments.
- **prod(x)** : Produit des éléments.
- **mean(x)** : Moyenne des éléments.(mean=Mean)
- **sort(x)** : Classement des éléments par ordre croissant.
- **log10(v)** : Logarithme base 10

Remarque :ATTENTION A LA MAJUSCULE

Le générateur de nombres pseudo-aléatoires de Matlab permet d'obtenir en une seule instruction un vecteur dont les éléments sont tirés au hasard. Pour générer des valeurs pseudo-aléatoires de loi uniforme sur [0, 1] (valeurs tirées au hasard uniformément dans l'intervalle [0, 1]) on utilise rand. On peut également générer des valeurs pseudo-aléatoires de loi gaussienne de moyenne nulle et de variance 1 avec randn.

```
>> u=rand(1,5)
u =
0.9501    0.2311    0.6068    0.4860    0.8913
>> n=randn(4,1)
n =
-0.4326
-1.6656
0.1253
0.2877
```

On refait les deux instructions ci-dessus plusieurs fois d'affilée. On constate que les valeurs changent à chaque fois, c'est-à-dire à chaque tirage.

Exercice 3 :

Vérifier alors les commandes suivantes :

```
>> x=[16 8 -9 1] ;
>> m=min(x)
>> M=max(x)
>> P=prod(x)
>>S = sum(x)
>>Moy=mean(x)
>> tri=sort(x)
```

Déterminer le résultat de la commande size(x).