L'ECOLE DE L'INNOVATION ET DE L'EXPERTISE INFORMATIQUE

B1 – Mathématiques : Calcul vectoriel

Amine ILMANE

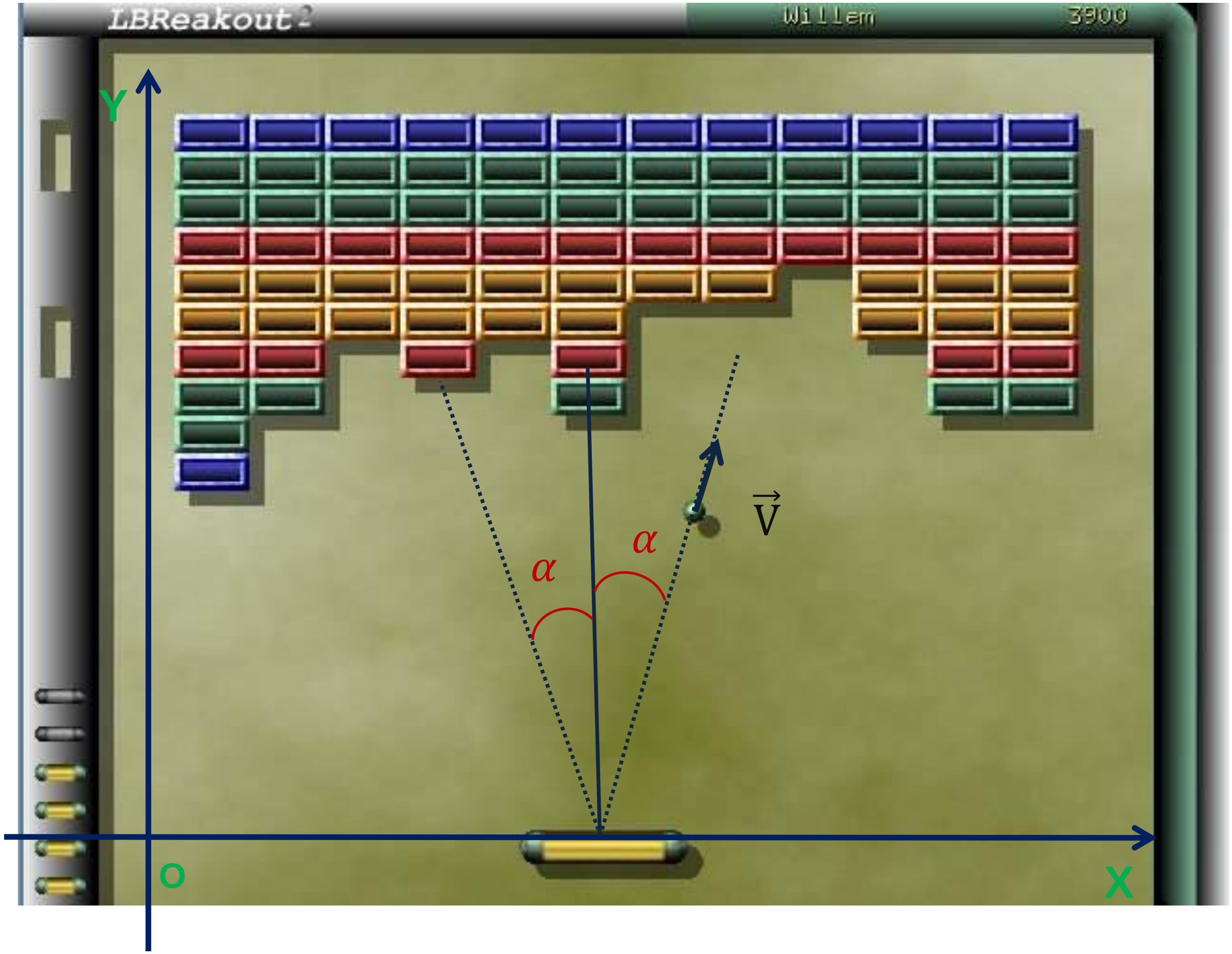
101pong

Vectors and Video Games

Ralph Baer en 1972 (Atari)

101pong

Vectors and Video Games



Projet 101 : Pong

The goal of this project is to work on a 3D version of this game (or of the *Breakout* game…). Only one paddle will be considered, located in the (Oxy) plane (which is defined by the equation $z = 0$).

Terminal

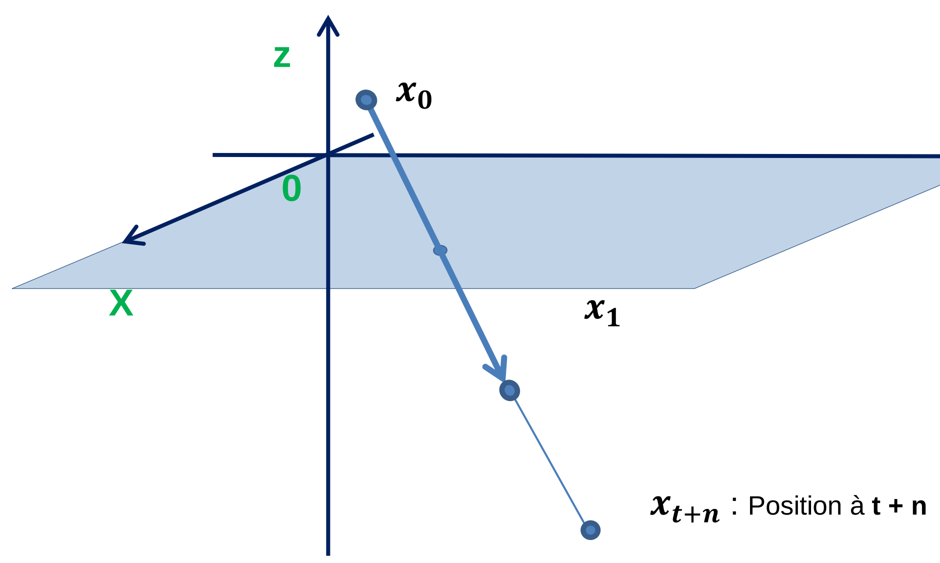
```
~/B-MAT-100> ./101pong -h
USAGE
      ./101pong x0 y0 z0 x1 y1 z1 n

DESCRIPTION
      x0    ball abscissa at time t - 1
      y0    ball ordinate at time t - 1
      z0    ball altitude at time t - 1
      x1    ball abscissa at time t
      y1    ball ordinate at time t
      z1    ball altitude at time t
      n     time shift (greater than or equal to zero, integer)
```

Mind the float numbers precision!

Projet 101 : Pong

```
Terminal
~/B-MAT-100> ./101pong 1 3 5 7 9 -2 4
The velocity vector of the ball is:
(6.00, 6.00, -7.00)
At time t + 4, ball coordinates will be:
(31.00, 33.00, -30.00)
The ball won't reach the bat.
```

Le plan (**Oxy**) : est ce que la balle touchera le plan ?

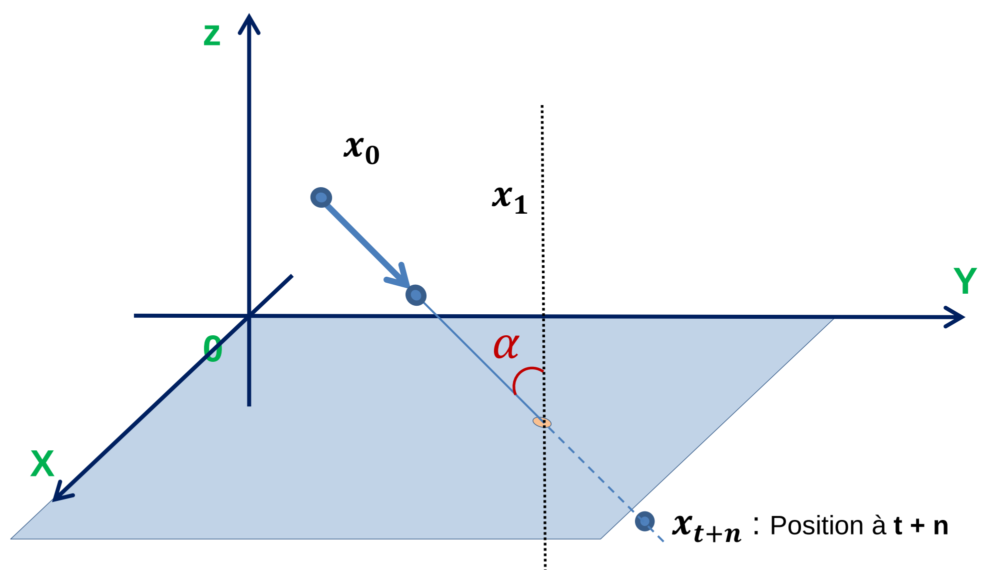


Projet 101 : Pong

Terminal

```
~/B-MAT-100> ./101pong 1.1 3 5 -7 9 2 4
The velocity vector of the ball is:
(-8.10, 6.00, -3.00)
At time t + 4, ball coordinates will be:
(-39.40, 33.00, -10.00)
The incidence angle is:
16.57 degrees
```

The incidence angle should be between 0 and 90 degrees.



N.B. l'angle d'incidence se calcule entre le vecteur normale au plan (Oxy) et le vecteur ….. ??

Projet 101 : Pong

Généralisation : **utilisation de l'écriture vectorielle**

$$\vec{\mathbf{V}} = \frac{1}{(\boldsymbol{t}_{final} - \boldsymbol{t}_{initial})}\overrightarrow{\boldsymbol{x_0 x_1}} = \frac{1}{\Delta \boldsymbol{t}}\begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix}$$

$$\vec{\mathbf{V}} = \begin{pmatrix}V_x\\ V_y\\ V_z\end{pmatrix} = \begin{cases} V_x = \dfrac{\boldsymbol{x}_{final} - \boldsymbol{x}_{initial}}{\boldsymbol{t}_{final} - \boldsymbol{t}_{initial}} \\ V_y = \dfrac{\boldsymbol{y}_{final} - \boldsymbol{y}_{initial}}{\boldsymbol{t}_{final} - \boldsymbol{t}_{initial}} \\ V_z = \dfrac{\boldsymbol{z}_{final} - \boldsymbol{z}_{initial}}{\boldsymbol{t}_{final} - \boldsymbol{t}_{initial}} \end{cases} = \begin{cases} V_x = \dfrac{1}{\Delta \boldsymbol{t}}\ \Delta x \\ V_y = \dfrac{1}{\Delta \boldsymbol{t}}\ \Delta y \\ V_z = \dfrac{1}{\Delta \boldsymbol{t}}\ \Delta z \end{cases} = \frac{1}{\Delta \boldsymbol{t}}\begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix} = \frac{1}{\Delta \boldsymbol{t}}\overrightarrow{\boldsymbol{x_0 x_1}}$$

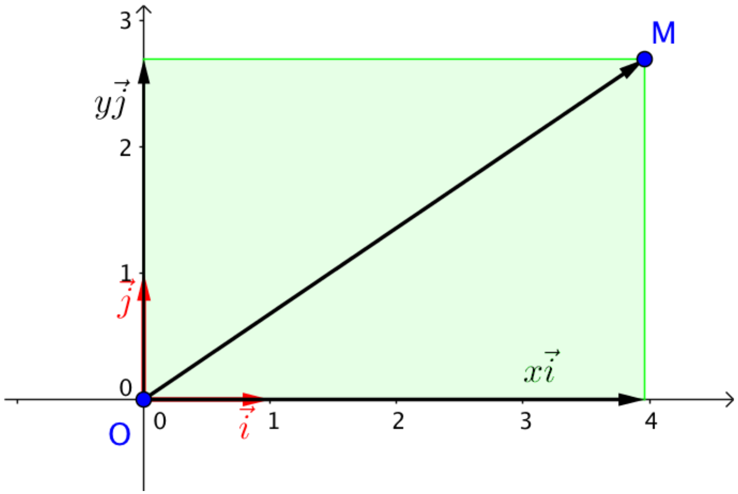
Terminal

```
~/B-MAT-100> ./101pong 1 3 5 7 9 -2 4
The velocity vector of the ball is:
(6.00, 6.00, -7.00)
At time t + 4, ball coordinates will be:
(31.00, 33.00, -30.00)
The ball won't reach the bat.
```

N.B. Le symbole ' Δ ' représente la différence entre une valeur finale est une valeur initiale $\Delta = x_2 - x_1$

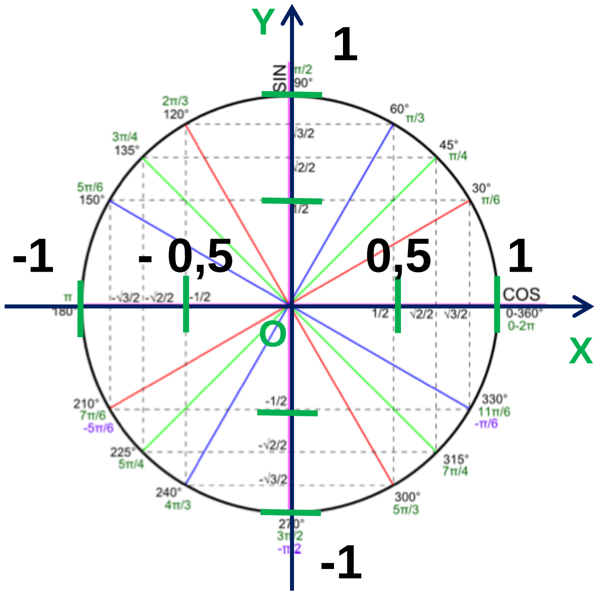
Chapitre 1 : Calcul vectoriel

- Trigonométrie
- Calcul vectoriel
  - Opérations sur les vecteurs
  - Norme et produit scalaire
  - Produit vectoriel
- Equations paramétriques



Fonctions trigonométriques

L'unité de mesure d'une angle **θ** est le radian. Un tour complet représente la valeur 2**π**.



- Un angle est déterminé par son **cosinus** et son **sinus : θ** (**cos , sin**)
- Les angles sont déterminés à 2**π** près :

  $sin(\theta + 2\pi) = sin(\theta)$ $\quad\quad$ $cos(\theta + 2\pi) = cos(\theta)$

- Valeurs possibles : $-1 \leq sin(\theta) \leq 1$ $\quad\quad$ $-1 \leq cos(\theta) \leq 1$
- Propriété importante : $cos^2(\theta) + sin^2(\theta) = 1$
- La fonction tangente : $tg(\theta) = \dfrac{sin(\theta)}{cos(\theta)}$
- Elle donne des valeurs réelles et, est définie pour tous les angles sauf $\frac{\pi}{2} + \mathrm{k}.\pi$
- Périodicité : $tg(\theta + \pi) = tg(\theta)$

Fonctions trigonométriques

Angle en radians	Angle en degrés	Cosinus	Sinus	Tangente
0	0°	1	0	0
$\pi/6$	30°	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\frac{1}{\sqrt{3}}$
$\pi/4$	45°	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	1
$\pi/3$	60°	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\sqrt{3}$
$\pi/2$	90°	0	1	$+\infty$
π	180°	-1	0	0
$3\pi/2$	270°	0	-1	$-\infty$

Les vecteurs

- Un vecteur est une quantité de l'espace définie par une direction et une longueur on le note $\overrightarrow{u}$

- Un vecteur peut également être définie par ses composantes , le nombre de composante dépend de la dimension de l'espace :

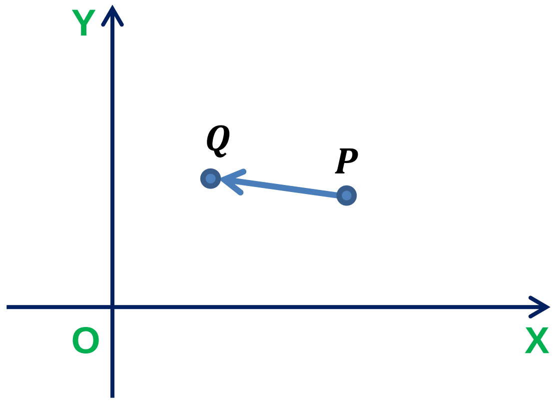
$$\overrightarrow{u} = \begin{pmatrix} x \\ y \end{pmatrix}$$

- Un vecteur peut être représenté dans le plan par un segment de droite orienté. Il a un point d'origine et un point extrémité que l'on notera resp. **P** et **Q**.

- Les composantes se calculent en utilisant les coordonnées des point **P** et **Q** par la formule :

$$\overrightarrow{u} = \begin{pmatrix} x \\ y \end{pmatrix} = \overrightarrow{PQ} = \begin{pmatrix} x_Q - x_P \\ y_Q - y_P \end{pmatrix}$$

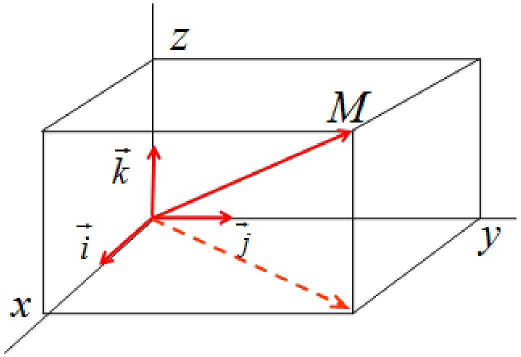
N.B.
La position du vecteur dans l'espace n'est pas importante seul compte la direction et la longueur. Ils ont les mêmes coordonnées et sont parallèles.



Les vecteurs en dimension supérieur

En dimension **3** (**Ox**, **Oy**, **Oz**) : $\overrightarrow{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ $\overrightarrow{QP} = \begin{pmatrix} x_Q - x_P \\ y_Q - y_P \\ z_Q - z_P \end{pmatrix}$

En dimension **n** : $\overrightarrow{u} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ $\overrightarrow{QP} = \begin{pmatrix} x_{1_Q} - x_{1_P} \\ x_{2_Q} - x_{2_P} \\ \vdots \\ x_{n_Q} - x_{n_P} \end{pmatrix}$



Un vecteur est toujours noté en **colonne**. L'opération qui consiste à le transformer en vecteur **ligne** s'appelle la **transposition** :

$$\overrightarrow{u} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \xrightarrow{\text{Transposition}} \overrightarrow{u}^T = (x_1, x_2, \ldots, x_n)$$

Les vecteurs en informatique

- Les vecteurs sont représentés par des **tableaux, champs ou liste** auxquels on impose des règles particulière de calcul :

    - Addition de deux tableaux ;
    - Multiplication par un nombre ;
    - Produit scalaire ;
    - …

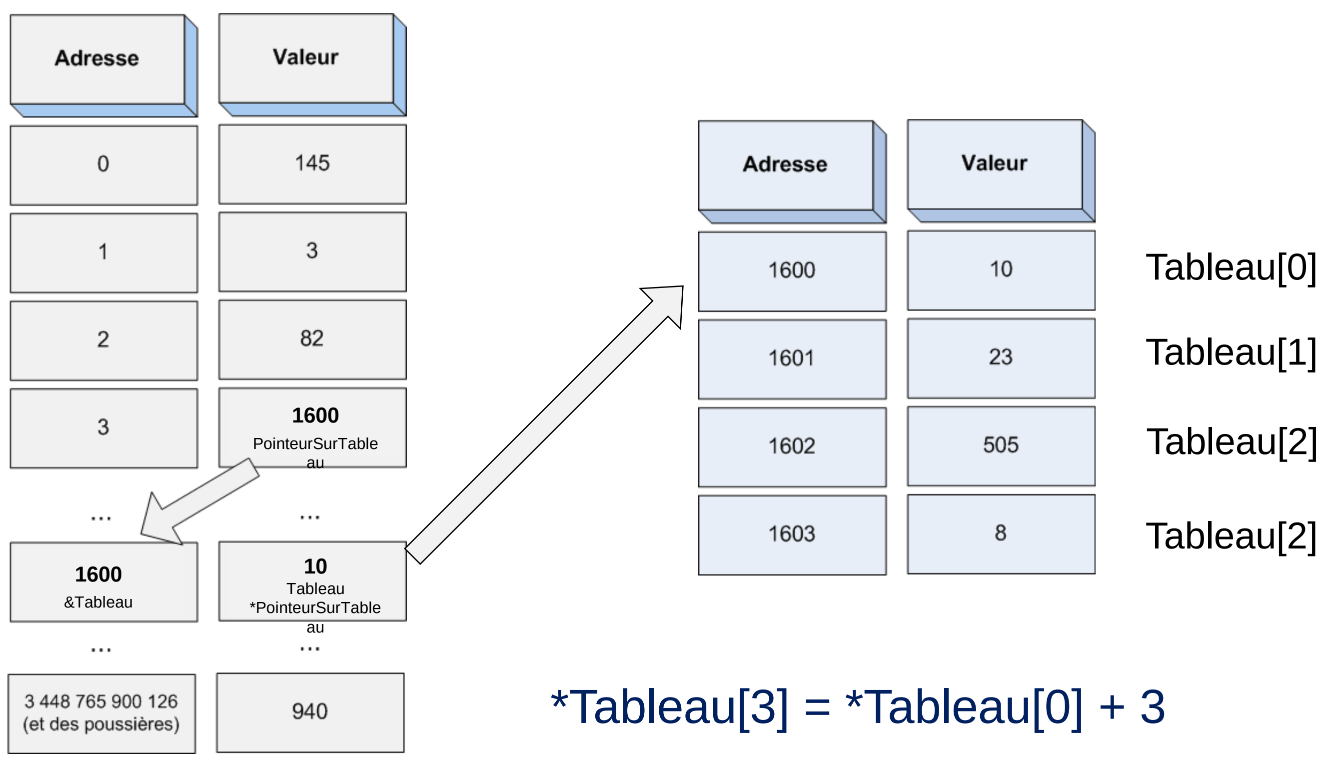
- l’indexation commence par 1 mais peut commencer par 0 dans certains langages.

- Les Tableaux permettent un accès mémoire plus rapide aux données, on parle de mémoire contigus

$$\overrightarrow{u} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Les vecteurs en informatique

- Les Tableaux permettent un accès mémoire plus rapide aux données, on parle de mémoire contigus ;
- Nous n’avons pas besoin de connaître l’adresse de tous les éléments juste le premier ;
- Le processeur calculera l’adresse d’un élément en ajoutant la position à l’adresse du premier élément



*Tableau[3] = *Tableau[0] + 3

Les vecteurs en informatique

- En C, un tableau se déclare comme suit : **type nom_tableau[dimension]**

Exemple : float mon_tableau[150]

- En Python la structure de données est la **liste** :
  - pas de déclaration ;
  - voici deux techniques pour déclarer un tableau d'une taille donnée :

```python
dimension = 5
vector = []
for i in range(dimension):
    vector.append(0)

print(vector)
```

```python
dimension = 5
vector_2 = [0 for i in range(dimension)]
print(vector_2)
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
[0, 0, 0, 0, 0]
```

Mutable objects in python

```python
dimension = 5

vector_2 = [i**2 for i in range(dimension)]
print('vector_2 : ', vector_2)

vector_3 = vector_2
vector_3[0] = vector_3[2] = -5

print('vector_2 : ', vector_2)
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
vector_2 :  [0, 1, 4, 9, 16]
vector_2 :  [-5, 1, -5, 9, 16]
```

- Le symbole ‘’ = ‘’ représente pour les ‘’ mutable objects ’’ un pointeur, donc :

vector_3 pointe vers vector_2

- Il faut initialiser chaque tableau tout seul.

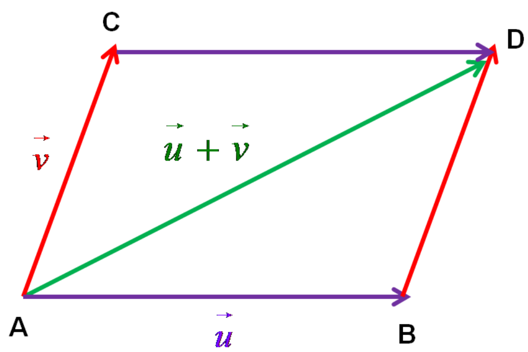
Addition de deux vecteurs

- La somme de deux vecteurs de dimension **n** est un vecteur de même dimension où chacune de ses composantes est la somme des composantes des deux vecteurs :

$$\overrightarrow{u} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \overrightarrow{v} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad \overrightarrow{u} + \overrightarrow{v} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{pmatrix}$$

- Le vecteur nul représente **l'élément neutre** : $\overrightarrow{u} + \overrightarrow{0} = \overrightarrow{0} + \overrightarrow{u} = \overrightarrow{u}$

Règle du parallélogramme :



$$\vec{u} = \overrightarrow{AB}$$

$$\vec{v} = \overrightarrow{AC}$$

$$\vec{u} + \vec{v} = \overrightarrow{AD}$$

Addition de deux vecteurs

Exemple :

$$\overrightarrow{u} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \overrightarrow{v} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \qquad \overrightarrow{u} + \overrightarrow{v} = \begin{pmatrix} 1+4 \\ 2+5 \\ 3+6 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}$$

```python
def add_vector(vector_1: list, vector_2: list):
    if len(vector_1) != len(vector_2):
        print("vector_1 and vector_2 don't have same dimension")
        exit(84)

    dimension = len(vector_1)
    sum = [0 for j in range(dimension)]
    for i in range(dimension):
        sum[i] = vector_1[i] + vector_2[i]

    return sum


vector_1 = [1, 2, 3]
vector_2 = [4, 5, 6]
vector_3 = add_vector(vector_1, vector_2)
print('vector_1 + vector_2 = ', vector_3)
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
vector_1 + vector_2 =  [5, 7, 9]
```

Multiplication d'un vecteur par un nombre

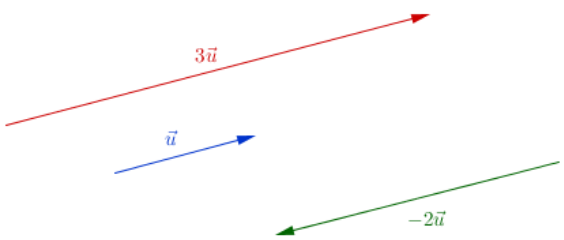
La multiplication d'un vecteur par un nombre réel λ revient à multiplier chaque composante le nombre λ

$$\overrightarrow{u} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \lambda.\overrightarrow{u} = \begin{pmatrix} \lambda.x_1 \\ \lambda.x_2 \\ \vdots \\ \lambda.x_n \end{pmatrix}$$

On parle d'homothétie (fonction zoom)

Exemple :

$$u = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad 2\overrightarrow{u} = \begin{pmatrix} 2 \times 1 \\ 2 \times 2 \\ 2 \times 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$$




```
def mult_number_vector(vector: list, number: float):
    dimension = len(vector)
    result = [0 for i in range(dimension)]
    for i in range(dimension):
        result[i] = vector[i] * number

    return result


vector_1 = [1, 2, 3]
vector_2 = mult_number_vector(vector_1, 2)
print('Multiplication de vector_1 par 2 =', vector_2)
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
Multiplication de vector_1 par 2 = [2, 4, 6]
```

Norme d'un vecteur

La norme d' un vecteur est un quantité positive qui mesure **sa longueur** définit par :

$$||\overrightarrow{u}|| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Exemple :

Dans un plan (**Oxy**) la distance entre deux points (longueur du segment **[OP]**) est : $d(O,P) = \sqrt{x^2 + y^2}$

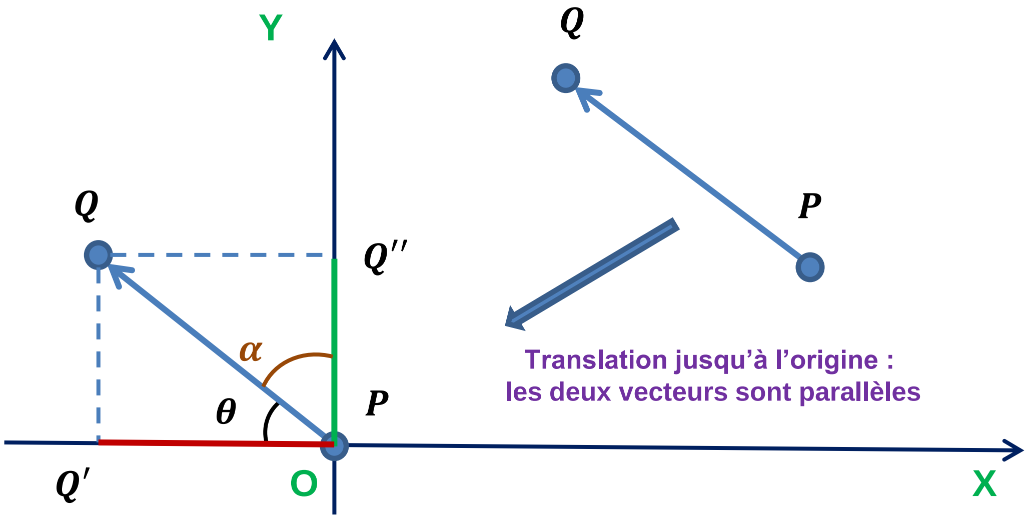
$$\overrightarrow{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$||\overrightarrow{u}|| = \sqrt{2}$$

```python
from math import sqrt

def vector_norm(vector: list):
    norm = 0.
    for i in range(len(vector)):
        norm += vector[i] ** 2

    return sqrt(norm)

vector_1 = [1, 1]
print('La norme de vector_1 : || vector_1 || = ', vector_norm(vector_1))
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
La norme de vector_1 : || vector_1 || =  1.4142135623730951
```

Les vecteurs et la trigonométrie



Le triangle **PQQ'** : $x = PQ' = QQ'' = \left\|\overrightarrow{QP}\right\|.\cos(\theta)$ $y = QQ' = PQ'' = \left\|\overrightarrow{QP}\right\|.\sin(\theta)$

Le triangle **PQQ''** : $x = PQ'' = QQ'' = \left\|\overrightarrow{QP}\right\|sin(\alpha)$ $y = QQ' = PQ'' = \left\|\overrightarrow{QP}\right\|.cos(\alpha)$

$$\overrightarrow{PQ} = \binom{x}{y} = \binom{x_Q - x_P}{y_Q - y_P} = \begin{pmatrix} \left\|\overrightarrow{QP}\right\|.\cos(\theta) \\ \left\|\overrightarrow{QP}\right\|.\sin(\theta) \end{pmatrix} = \begin{pmatrix} \left\|\overrightarrow{QP}\right\|.sin(\alpha) \\ \left\|\overrightarrow{QP}\right\|.\cos(\alpha) \end{pmatrix}$$

Produit scalaire de deux vecteurs 1

Le produit scalaire de deux vecteurs de dimension **n** est un nombre défini comme étant la somme du produit des composantes des deux vecteurs :

$$(\overrightarrow{u}, \overrightarrow{v}) = x_1.y_1 + x_2.y_2 + ... + x_n.y_n = \sum_{i=1}^{n} x_i.y_i$$

Propriétés : $(\overrightarrow{u}, \overrightarrow{v}) = (\overrightarrow{v}, \overrightarrow{u})$ $\qquad (\overrightarrow{u}, \overrightarrow{u}) = \|u\|^2$

Exemple :

$$\overrightarrow{u} = \begin{pmatrix} -1 \\ 2 \\ 5 \end{pmatrix} \qquad \overrightarrow{v} = \begin{pmatrix} 5 \\ 1 \\ -10 \end{pmatrix}$$

$$\begin{aligned}(\overrightarrow{u}, \overrightarrow{v}) &= -1 \times 5 + 2 \times 1 + 5 \times (-10) \\ &= -5 + 2 - 50 \\ &= -53\end{aligned}$$

```python
def scalar_product_1(vec_1: list, vec_2: list):
    if len(vec_1) != len(vec_2):
        print("vector_1 and vector_2 don't have same dimension")
        exit(84)

    scalar_prod = 0.
    for i in range(len(vec_1)):
        scalar_prod += vec_1[i] * vec_2[i]

    return scalar_prod


vec_1 = [-1, 2, 5]
vec_2 = [5, 1, -10]
print('Le produit scalaire de vec_1 et vec_2 : vec_1 . vec_2 = ',
      scalar_product_1(vec_1, vec_2))
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
Le produit scalaire de vec_1 et vec_2 : vec_1 . vec_2 =  -53.0
```

Produit scalaire de deux vecteurs 2

Autre formule : $$(\overrightarrow{u}, \overrightarrow{v}) = \|\overrightarrow{u}\|.\|\overrightarrow{v}\|.cos(\theta)$$

N.B. Le produit scalaire de deux vecteurs **orthogonaux** est **nul.**

```python
from math import cos, pi, acos


def scalar_product_2(vec_1: list, vec_2: list, angle: float):
    if len(vec_1) != len(vec_2):
        print("vector_1 and vector_2 don't have same dimension")
        exit(84)

    scalar_prod = vector_norm(vec_1) * vector_norm(vec_2) * cos(angle)

    return scalar_prod


vec_1 = [-1, 2, 5]
vec_2 = [5, 1, -10]
angle = 2.6100872559310079878969190758652985095977783203125  # in radians

print('Le produit scalaire de vec_1 et vec_2 :')
print(' ' * 10 + 'méthode 1 = ', scalar_product_1(vec_1, vec_2))
print(' ' * 10 + 'méthode 2 = ', scalar_product_2(vec_1, vec_2, angle))

print('\nSome float Manipulation')
print(' ' * 7 + '*  format = ', '{:.13f}'.format(scalar_product_2(vec_1, vec_2, angle)))
print(' ' * 7 + '*  round = ', round(scalar_product_2(vec_1, vec_2, angle), 13))
print(' ' * 7 + '*  fraction rationnelle a/b = ',
      scalar_product_2(vec_1, vec_2, angle).as_integer_ratio())
print(' ' * 7 + '*  ceil, floor, frexp, ...')
```

```
(venv) G:\Enseignement\Epitech\Cours\1 ere\Semestre 1 - B1>python tek1.py
Le produit scalaire de vec_1 et vec_2 :
          méthode 1 =  -53.0
          méthode 2 =  -52.99999999999999

Some float Manipulation
       *  format =  -53.0000000000000
       *  round =  -53.0
       *  fraction rationnelle a/b =  (-7459086882832383, 140737488355328)
       *  ceil, floor, frexp, ...
```

Comment déterminer l'angle entre deux vecteurs

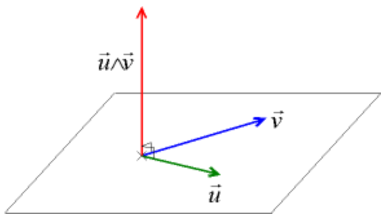
Exemple : $\overrightarrow{u} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ $\overrightarrow{v} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ $cos(\theta) = \frac{(\overrightarrow{u}, \overrightarrow{v})}{\|\overrightarrow{u}\|.\|\overrightarrow{v}\|} = \frac{1}{\sqrt{2}.\sqrt{2}} = \frac{1}{2}$

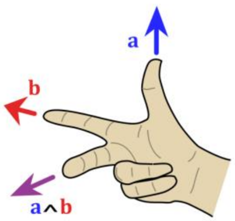
En appliquant la fonction **acos**(), inverse de la fonction **cosinus :** $\theta = \pi/3$

Produit vectoriel de deux vecteurs en dimensions 3

Le produit vectoriel (produit extérieur) de deux vecteurs est le vecteur défini par :

$$\overrightarrow{u} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \overrightarrow{v} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \qquad \overrightarrow{u} \wedge \overrightarrow{v} = \begin{pmatrix} x_2.y_3 - x_3.y_2 \\ x_3.y_1 - x_1.y_3 \\ x_1.y_2 - x_2.y_1 \end{pmatrix}$$





Exemple :

$$\overrightarrow{u} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \overrightarrow{v} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \qquad \overrightarrow{u} \wedge \overrightarrow{v} = \begin{pmatrix} 2 \times 6 - 3 \times 5 \\ 3 \times 4 - 1 \times 6 \\ 1 \times 5 - 2 \times 4 \end{pmatrix} \begin{pmatrix} -3 \\ 6 \\ -3 \end{pmatrix}$$

Autre formule :

$$\|\overrightarrow{u} \wedge \overrightarrow{v}\|) = \|\overrightarrow{u}\|.\|\overrightarrow{v}\|.sin(\theta)$$

Propriété d’antisymétrie :

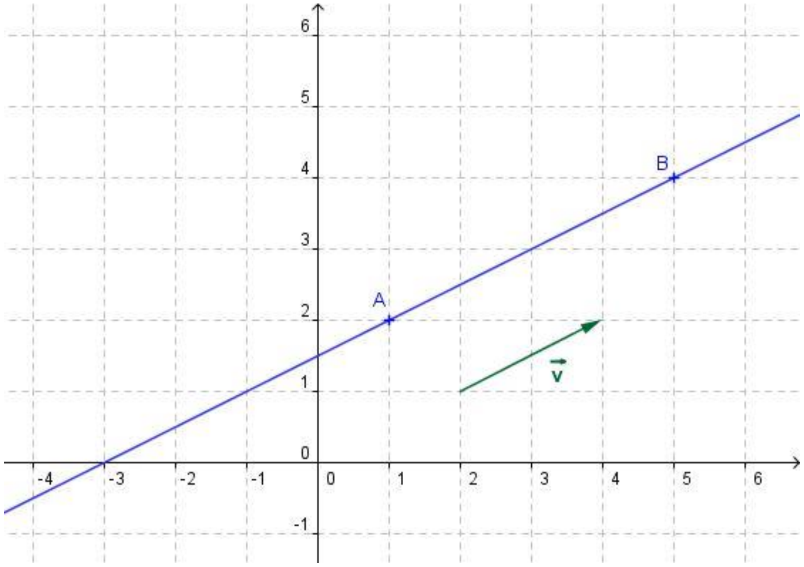
$$\overrightarrow{u} \wedge \overrightarrow{v} = -\overrightarrow{v} \wedge \overrightarrow{u}$$

Equations paramétriques de la droite

Une droite est déterminée par :

- deux point distinct (Euclide) ;
- ou par un **vecteur directeur** et un point **A.**

$$D = \{P \quad | \quad \overrightarrow{AP} = \lambda.\overrightarrow{u} \; = \begin{pmatrix} \lambda.x_1 \\ \lambda.x_2 \\ \vdots \\ \lambda.x_n \end{pmatrix} \lambda \text{ dans } \mathbb{R}\}$$



Equations paramétriques de la droite

Une droite est déterminé par deux point distinct ou par un vecteur directeur et un point **A** :

$$D = \{P \quad | \quad \overrightarrow{AP} = \lambda.\overrightarrow{u} \qquad \lambda \text{ dans } \mathbb{R}\}$$

Equations paramétriques de la droite **D** , en dimension **n**:

$$A = (a_1, a_2, ..., a_n) \qquad \overrightarrow{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \qquad P(x_1, x_2, ..., x_n) \; = \begin{cases} x_1 = a_1 + \lambda.u_1 \\ x_2 = a_2 + \lambda.u_2 \\ \vdots \\ x_n = a_n + \lambda.u_n \end{cases}$$

Exemple : **A** (3 , -1) $\overrightarrow{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $P \; = \begin{cases} x = \lambda + 3 \\ y = \lambda - 1 \end{cases}$ $x - y - 2 = 0$

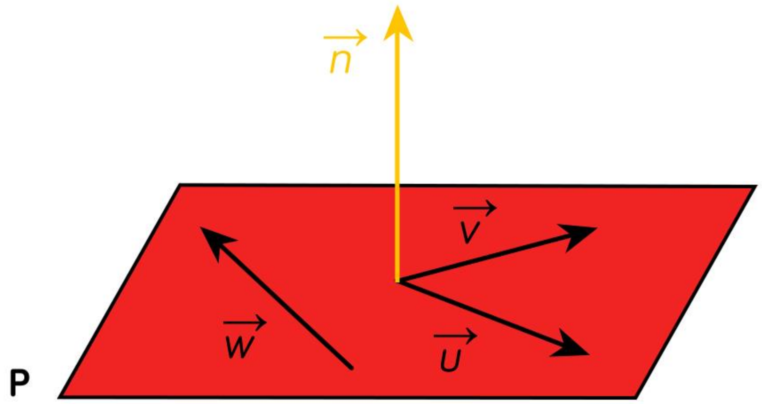
Equation cartésienne du plan

Un plan est déterminé par deux vecteurs non colinéaires et un point **A** :

$$\{P \quad | \quad \overrightarrow{AP} = \lambda.\overrightarrow{u} + \mu.\overrightarrow{v} \qquad \lambda, \mu \text{ dans } \mathbb{R}\}$$

L'équation cartésienne du plan **P**, en dimension **n** est déterminée par $\vec{\boldsymbol{n}} = (\boldsymbol{a_1}, \dots, \boldsymbol{an})$ le vecteur normal au plan :

$$a_1.x_1 + a_2.x_2 + \dots + a_n.x_n + a_{n+1} = 0$$



Equation paramétrique du plan

Un plan est déterminé par deux vecteurs non colinéaires et un point **A** :

$$\{P \quad | \quad \overrightarrow{AP} = \lambda.\overrightarrow{u} + \mu.\overrightarrow{v} \qquad \lambda, \mu \text{ dans } \mathbb{R}\}$$

Equations paramétriques du plan **P**, en dimension **n**:

$$A = (a_1, a_2, ..., a_n) \qquad \overrightarrow{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \quad \overrightarrow{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \qquad P(x_1, x_2, ..., x_n) = \begin{cases} x_1 = a_1 + \lambda.u_1 + \mu.v_1 \\ x_2 = a_2 + \lambda.u_2 + \mu.v_2 \\ \vdots \\ x_n = a_n + \lambda.u_n + \mu.v_n \end{cases}$$

Exemple : **A** (3 , 1, 4)

$$\overrightarrow{u} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \overrightarrow{v} = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \qquad P = \begin{cases} x = \lambda + 2\mu + 3 \\ y = \lambda - \mu + 1 \\ y = \lambda + 3\mu + 4 \end{cases}$$