

# Étude de la cinématique humaine

# PLAN DÉTAILLÉ

## I/METHODES ET TECHNIQUES

a/Mesures cinématiques

b/Paramètres inertiels segmentaires

## II/APPROCHE MÉCANIQUE

a/ Étude du saut à pieds joints (2 approches)

b/ Étude du mouvement du bras dans l'eau

c/ Mouvement d'oscillation (2 approches)

d/ Étude de la marche

## III/OUTILS INFORMATIQUES

a/ Algorithme de remplissage par diffusion

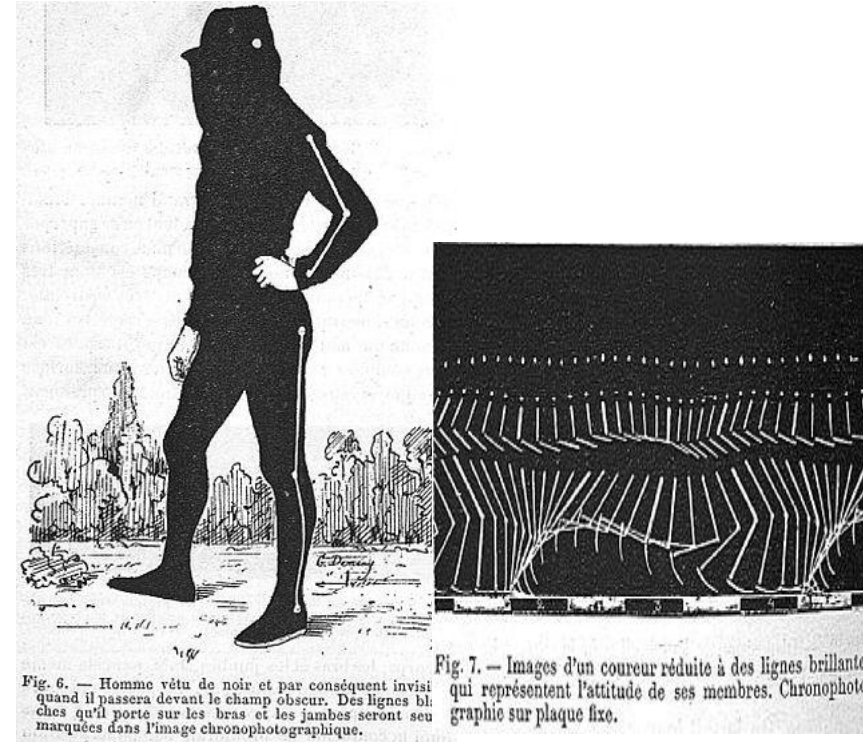
b/Algorithme de modèle de mélanges gaussiens (en 2D et en 3D)

# I/a/Mesures cinématiques

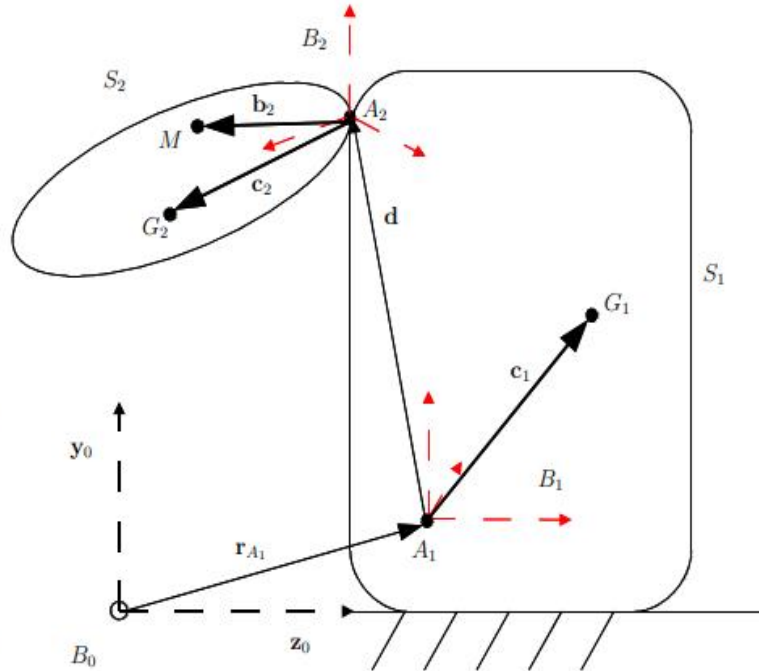
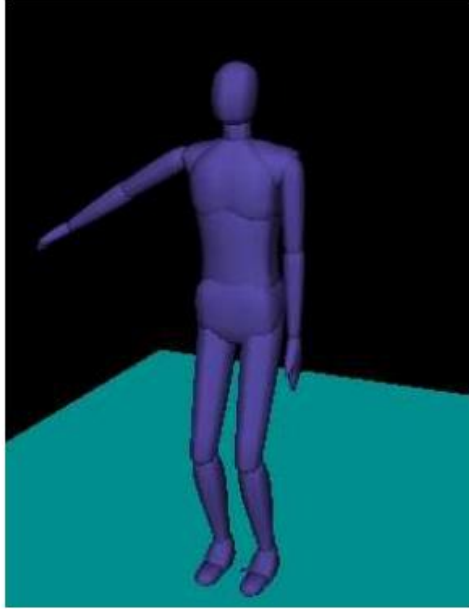
les systèmes optoélectroniques & infrarouges



Les systèmes magnétiques



# I/b/Paramètres inertiels segmentaires

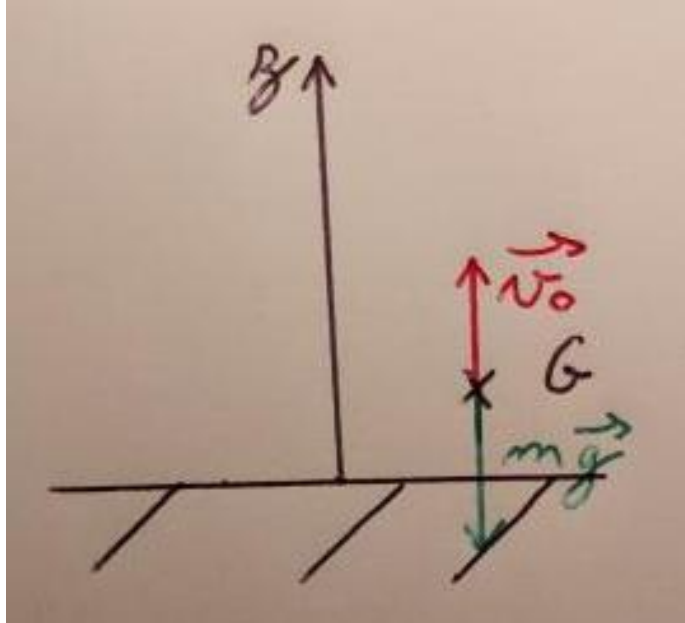


$$H = \begin{bmatrix} K_2^{(2)} & m_2 \mathbf{c}_2^{(2)} \mathbf{d}^{(1)T} & m_2 \mathbf{c}_2^{(2)} \\ \mathbf{d}^{(1)} m_2 \mathbf{c}_2^{(2)T} & K_1^{(1)} + m_2 \mathbf{d}^{(1)} \mathbf{d}^{(1)T} & m_1 \mathbf{c}_1^{(1)} + m_2 \mathbf{d}^{(1)} \\ m_2 \mathbf{c}_2^{(2)T} & (m_1 \mathbf{c}_1^{(1)} + m_2 \mathbf{d}^{(1)})^T & m_1 + m_2 \end{bmatrix}$$

$$A(t)HB^T(t) - B(t)HA^T(t) = C(t)$$

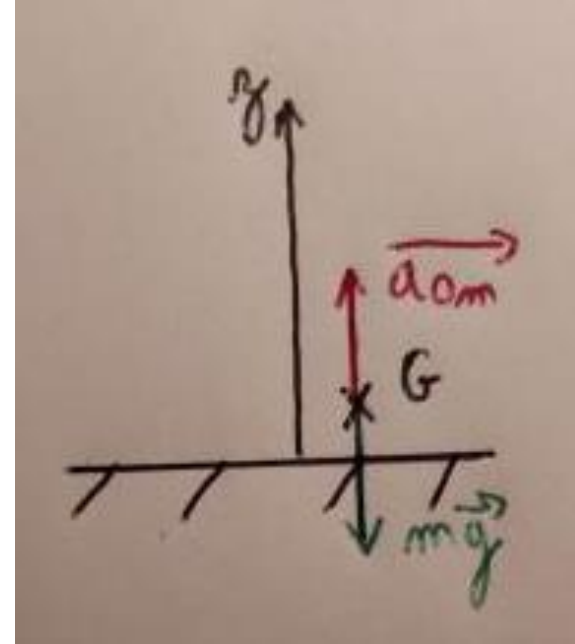
## II/a/Étude du saut à pieds joints

Approche naïve



$$z(t) = -\frac{1}{2}gt^2 + v_0t$$

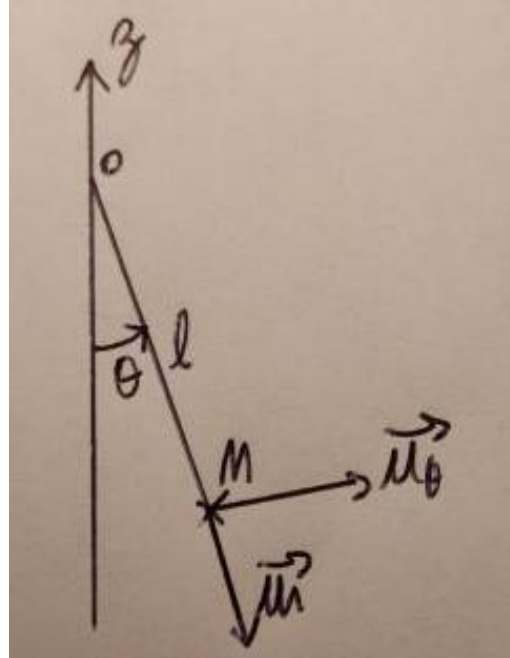
Approche plus réaliste



$$z(t) = -\frac{1}{2}(g + a_{Om})t^2$$

## II/b/Étude du mouvement du bras dans l'eau

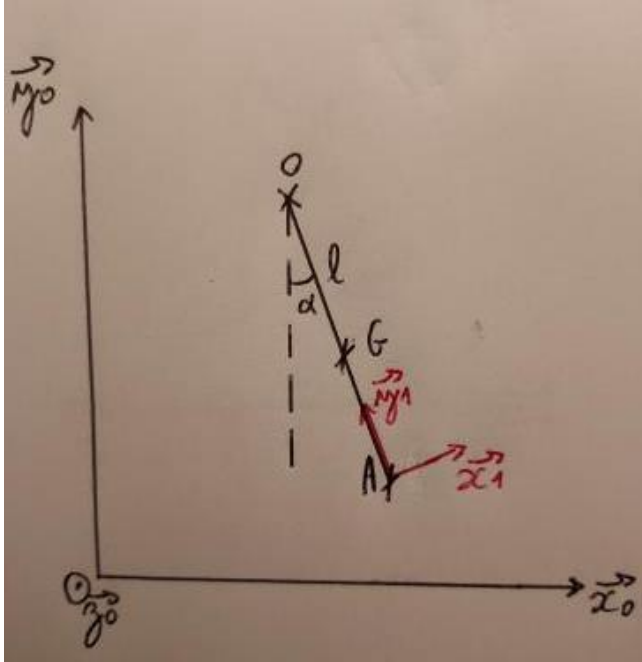
Mouvement du Crawl



$$\frac{1}{2}J\ddot{\theta} = (\rho V - m)gl\sin\theta$$

## II/c/Mouvement d'oscillation

Oscillateur harmonique

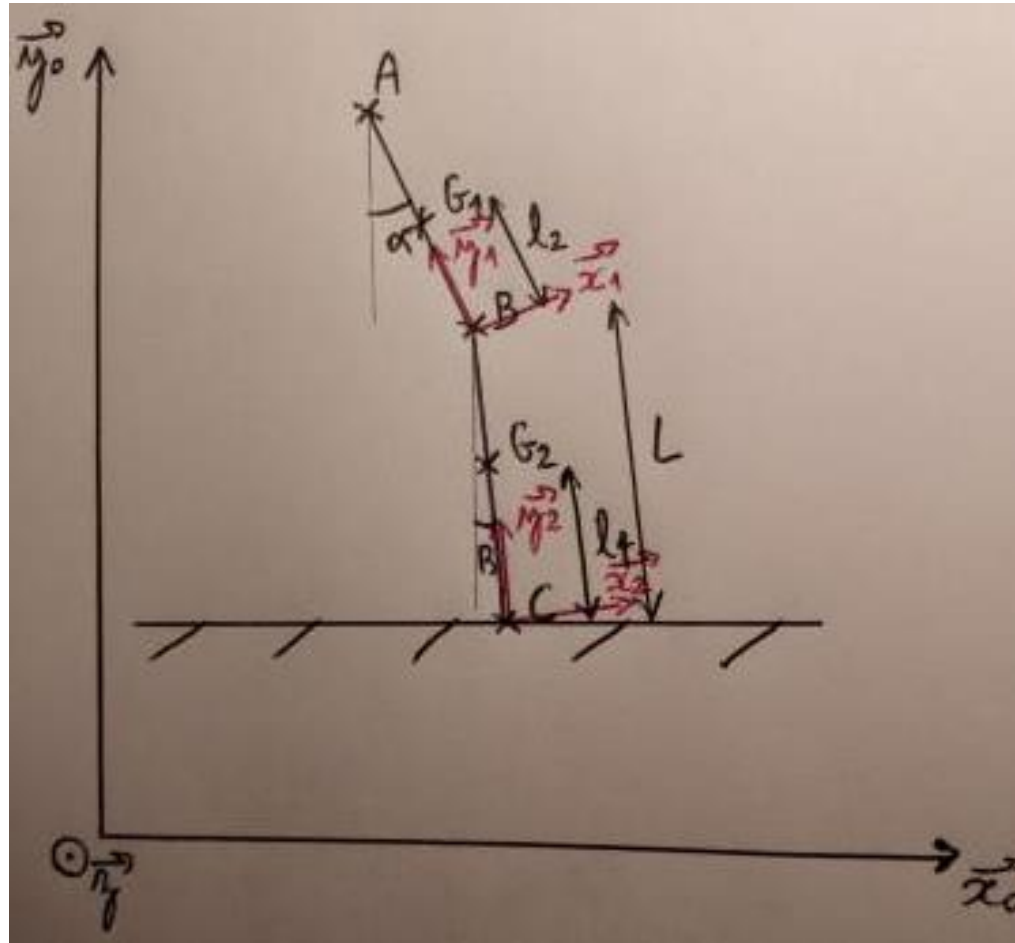


Pour un cylindre plein

$$\ddot{\alpha} + \omega_0^2 \alpha = 0 \quad (R^2 + l^2)m\ddot{\alpha} - mgl\alpha = C_A$$

## II/d/Etude de la marche

Schéma de la jambe pour la marche



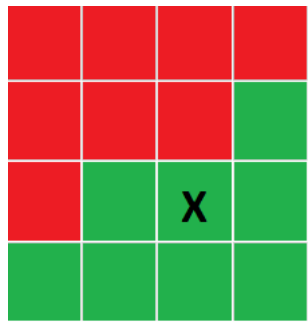


## II/d/Étude de la marche

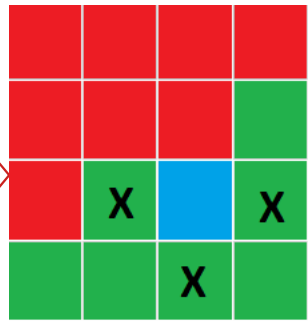
L'équation du mouvement associée à la marche

$$\begin{aligned} & \frac{d}{dt} \left( \frac{1}{2} (\dot{\alpha} - \dot{\beta})^2 m_1 R_1^2 + m_1 [L^2 \dot{\beta}^2 + l_2^2 (\dot{\alpha} - \dot{\beta})^2 + 2l_2 L \dot{\beta} (\dot{\alpha} - \dot{\beta}) \cos \beta] + \left[ \frac{1}{2} m_2 R_2^2 + m_2 l_4^2 \right] \dot{\beta}^2 \right) \\ & = \\ & m_1 g [L \dot{\beta} (\cos \beta \sin \alpha + \sin \beta \cos \alpha) + l_2 (\dot{\alpha} - \dot{\beta}) \sin \alpha] + m_2 g l_4 \dot{\beta} (\cos \beta \sin \alpha + \sin \beta \cos \alpha) + C_A (\dot{\alpha} - \dot{\beta}) + C_B \dot{\beta} \end{aligned}$$

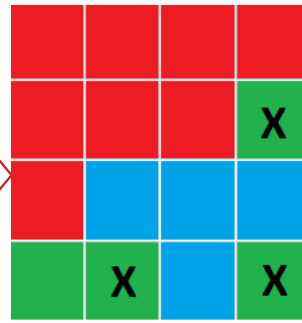
# III/a/Algorithme de remplissage par diffusion



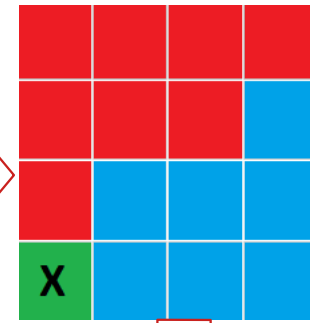
Étape 1



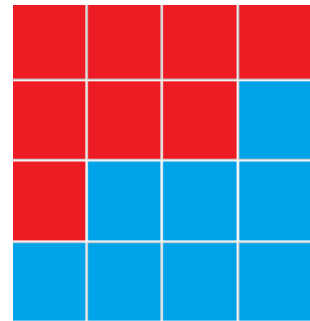
Étape 2



Étape 3



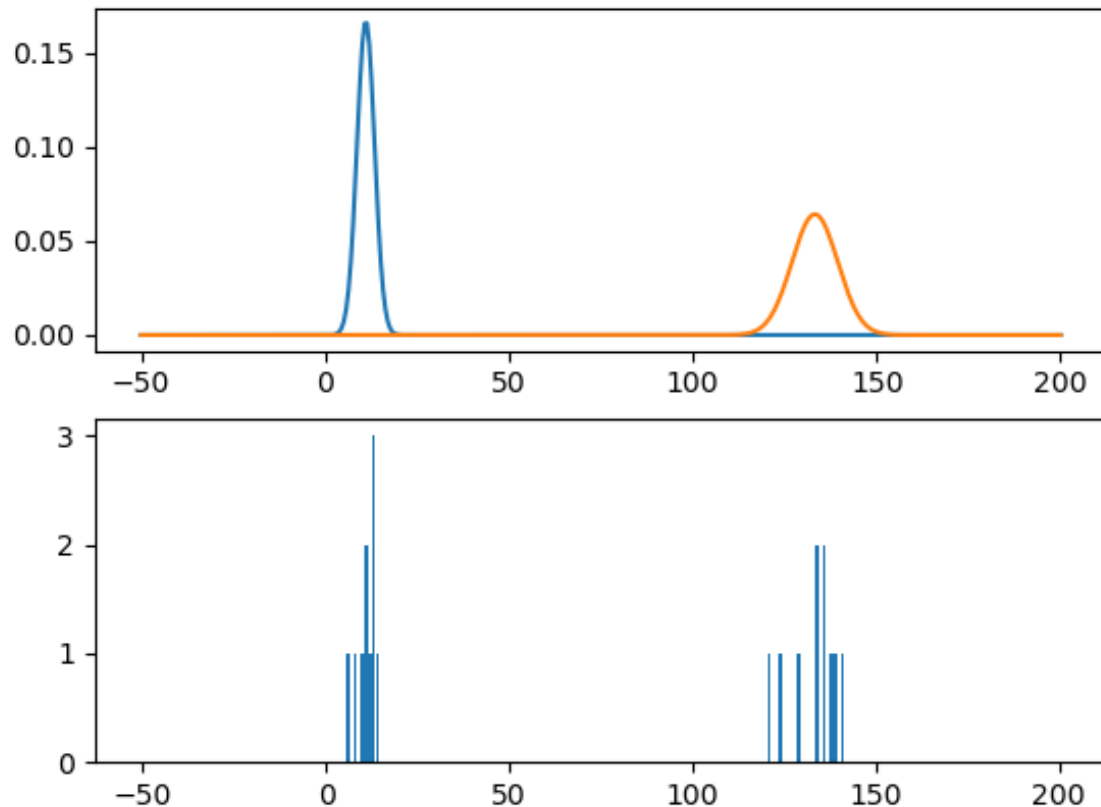
Étape 4



Étape 5 L'algorithme termine

# III/b/Algorithme de modèle de mélanges gaussiens

GMM en 2D avec histogramme

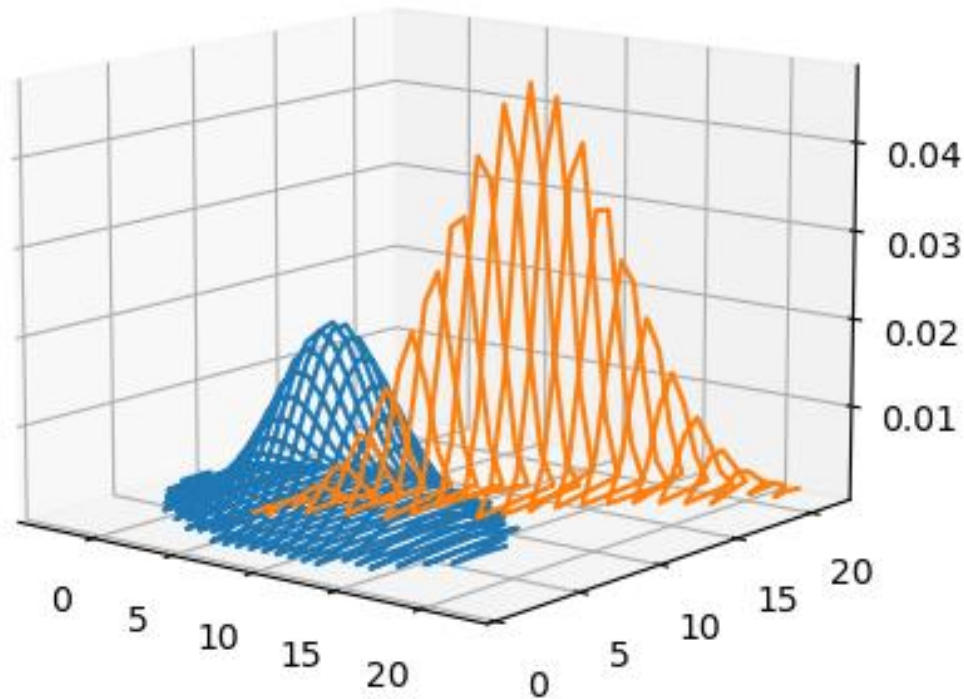


Loi normale multivariée

$$\frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (x-\mu)^\top \Sigma^{-1} (x-\mu)}$$

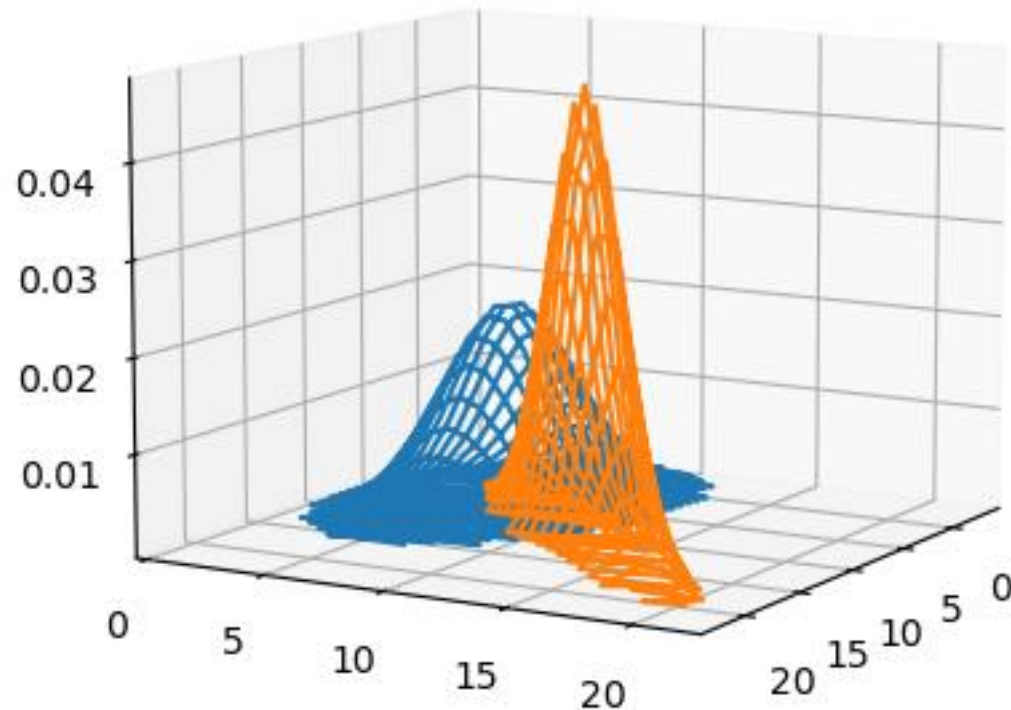
## III/b/Algorithme de modèle de mélanges gaussiens

GMM en 3D

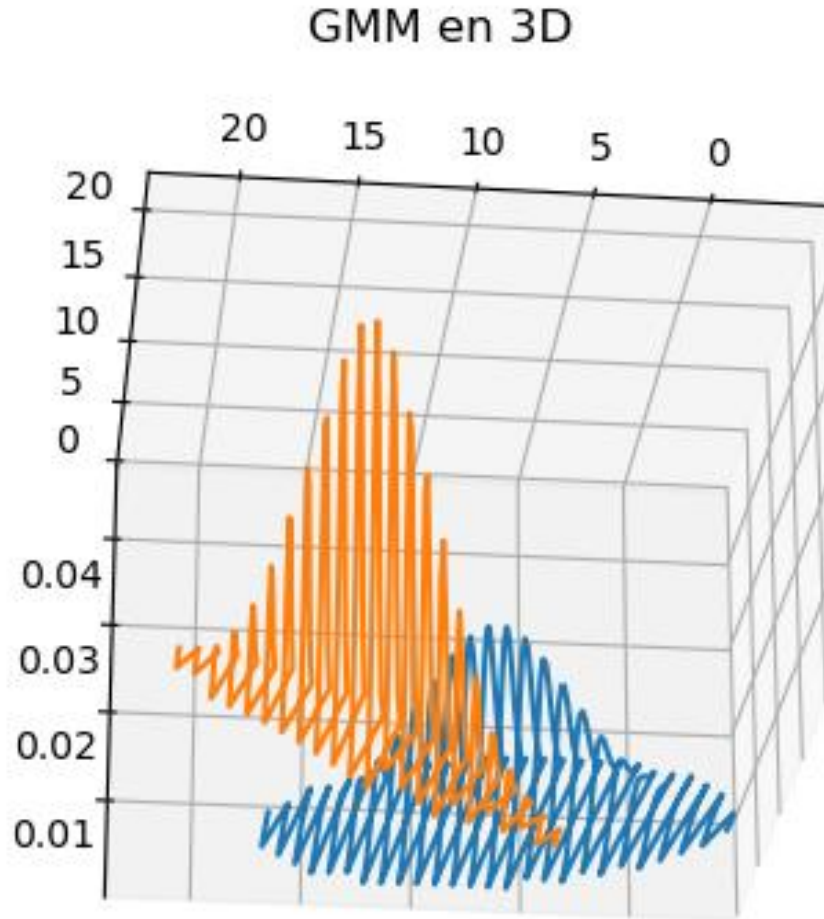


# III/b/Algorithme de modèle de mélanges gaussiens

GMM en 3D



## III/b/Algorithme de modèle de mélanges gaussiens



**Merci de votre attention**

## Annexe : Code Flood Fill

```
.from collections import deque
```

```
.# Liste des mouvements possibles pour l'algorithme
```

```
.deplacement = [
```

```
• (-1, 0), # gauche
```

```
• (1, 0), # droite
```

```
• (0, -1), # haut
```

```
• (0, 1), # bas
```

```
• (-1,-1), # diagonale haut/gauche
```

```
• (-1, 1), # diagonale bas/gauche
```



## Annexe : Code Flood Fill

- `def remplissageDiffusionIteratif(image, pointDepart, nouvelleCouleur):`
- `"""`
- `Fonction de remplissage par diffusion itérative. Cette fonction utilise la`
- `méthode de parcourt en largeur dans la théorie des graphes.`
- `Les arguments de la fonction sont:`
- `- L'image sous forme de liste de Liste`
- `- Les coordonnées du point de départ de la diffusion`
- `- La nouvelle couleur à utiliser`
- `"""`

## Annexe : Code Flood Fill

.def remplissageDiffusionRecuratif(image, pointDepart, nouvelleCouleur):

- """
- Fonction de remplissage par diffusion récursive. Cette fonction utilise la
- méthode de parcourt en profondeur dans la théorie des graphes.
- Les arguments de la fonction sont:
  - - L'image sous forme de liste de Liste
  - - Les coordonnées du point de départ de la diffusion
  - - La nouvelle couleur à utiliser

• """

## Annexe : Code Flood Fill

.image = [

- ['J', 'J', 'J', 'G', 'G', 'G', 'G', 'G', 'G', 'G'],
- ['J', 'J', 'J', 'J', 'J', 'J', 'G', 'V', 'V', 'V'],
- ['G', 'G', 'G', 'G', 'G', 'G', 'G', 'V', 'V', 'V'],
- ['B', 'B', 'B', 'B', 'B', 'G', 'G', 'G', 'G', 'V'],
- ['B', 'R', 'R', 'R', 'R', 'R', 'G', 'V', 'V', 'V'],
- ['B', 'B', 'B', 'R', 'R', 'G', 'G', 'V', 'V', 'V'],
- ['B', 'M', 'B', 'R', 'R', 'R', 'R', 'R', 'R', 'V'],
- ['B', 'M', 'M', 'M', 'M', 'R', 'R', 'V', 'V', 'V'],

['B', 'M', 'M', 'V', 'M', 'M', 'M', 'M', 'V', 'V']

## Annexe : Code GMM 2D

```
import random

import matplotlib.pyplot as plt

données=[]

for loop in range(10):
    données.append(random.randint(1,15))
    données.append(random.randint(100,150))

print(données)

import numpy as np

def v(x_moyenne, variance):
```

## Annexe : Code GMM 2D

```
for i in range(20):
```

- 

```
pourcentage_y.append(y(données[i],30,5)/(y(données[i],30,5)+y(données[i],60,6)))
```

- 

```
pourcentage_z.append(y(données[i],60,6)/(y(données[i],30,5)+y(données[i],60,6)))
```

•#On a calculé les pourcentages de chaque point des données

•moyenne\_y=0

•moyenne\_z=0

•variance\_y=0

## Annexe : Code GMM 2D

```
.for i in range(20):  
    variance_y+=((données[i]-moyenne_y)**2)*pourcentage_y[i]/coeff_y  
    variance_z+=((données[i]-moyenne_z)**2)*pourcentage_z[i]/coeff_z  
.variance_y=np.sqrt(variance_y)  
.variance_z=np.sqrt(variance_z)  
.listevariance_y.append(variance_y)  
.listevariance_z.append(variance_z)  
  
.#On a initialisé une fois la boucle pour donner la condition de fin pour  
la boucle à venir
```

## Annexe : Code GMM 2D

- `moyenne_z=0`
- `moyenne_y=0`
- `variance_y=0`
- `variance_z=0` #On réinitialise les stats
- `for i in range(20):`
  - `moyenne_y=moyenne_y+données[i]*pourcentage_y[i]/coeff_y`
  - `moyenne_z=moyenne_z+données[i]*pourcentage_z[i]/coeff_z`
- `listemoyenne_y.append(moyenne_y)`
- `listemoyenne_z.append(moyenne_z)`
- `for i in range(20):`

## Annexe : Code GMM 2D

```
.print(listemoyenne_y)
.moyenne_y=listemoyenne_y[len(listemoyenne_y)-1]
.moyenne_z=listemoyenne_z[len(listemoyenne_z)-1]
.variance_y=listevariance_y[len(listevariance_y)-1]
.variance_z=listevariance_z[len(listevariance_z)-1]
.#On affiche les courbes
.x=np.linspace(-50,200,400)
.courbe_y=y(x,moyenne_y,variance_y)
.courbe_z=y(x,moyenne_z,variance_z)
```



## Annexe : Code GMM 2D

• #On affiche l'histogramme

• quantite = {}

• for elem in données:

•   if elem in quantite:

•     quantite[elem] += 1

•   else:

•     quantite[elem] = 1

• val = quantite.values()

## Annexe : Code GMM 3D

```
.import matplotlib.pyplot as plt
.import numpy as np
.from mpl_toolkits.mplot3d import Axes3D
.import matplotlib.animation as animation
.def f(x,mu,sigma):
    . F=1/(2*np.pi*np.sqrt(float(np.linalg.det(sigma)))) * np.exp(-1/2
    *np.dot(np.dot(np.transpose(x-mu),np.linalg.inv(sigma)),(x-mu)))
    . return F
.import random
.random.seed(18)
```

## Annexe : Code GMM 3D

```
.for i in range(20):
```

```
•
```

```
pourcentage_y.append(f(données[i],listemoyenne_y[0],listeMCov_y[0])  
/(f(données[i],listemoyenne_y[0],listeMCov_y[0])+f(données[i],listemo  
yenne_z[0],listeMCov_z[0])))
```

```
•
```

```
pourcentage_z.append(f(données[i],listemoyenne_z[0],listeMCov_z[0])  
/(f(données[i],listemoyenne_y[0],listeMCov_y[0])+f(données[i],listemo  
yenne_z[0],listeMCov_z[0])))
```

```
•#On a calculé le pourcentage de chaque point des données
```

```
•moyenne_y=np.array([0,0])
```

```
•moyenne_z=np.array([0,0])
```

## Annexe : Code GMM 3D

```
.for i in range(20):  
    variance0_y+=((données[i][0]-  
moyenne_y[0])**2)*pourcentage_y[i]/coeff_y  
    variance1_y+=((données[i][1]-  
moyenne_y[1])**2)*pourcentage_y[i]/coeff_y  
    cov_y+=(données[i][0]-moyenne_y[0])*(données[i][1]-  
moyenne_y[1])*pourcentage_y[i]/coeff_y  
    variance0_z+=((données[i][0]-  
moyenne_z[0])**2)*pourcentage_z[i]/coeff_z  
    variance1_z+=((données[i][1]-  
moyenne_z[1])**2)*pourcentage_z[i]/coeff_z
```

## Annexe : Code GMM 3D

```
.while abs(listemoyenne_y[len(listemoyenne_y)-1][0]-  
listemoyenne_y[len(listemoyenne_y)-2][0])>0.01:
```

- `coeff_y=0`
- `coeff_z=0`
- `pourcentage_y=[]`
- `pourcentage_z=[]` #on réinitialise les pourcentages
- `for i in range(20):`
- `pourcentage_y.append(f(données[i],moyenne_y,MCov_y)/(f(données[i]`  
`,moyenne_y,MCov_y)+f(données[i],moyenne_z,MCov_z)))`

## Annexe : Code GMM 3D

- `for i in range(20):`
- `variance0_y+=((données[i][0]-  
moyenne_y[0])**2)*pourcentage_y[i]/coeff_y`
- `variance1_y+=((données[i][1]-  
moyenne_y[1])**2)*pourcentage_y[i]/coeff_y`
- `cov_y+=(données[i][0]-moyenne_y[0])*(données[i][1]-  
moyenne_y[1])*pourcentage_y[i]/coeff_y`
- `variance0_z+=((données[i][0]-  
moyenne_z[0])**2)*pourcentage_z[i]/coeff_z`
- `variance1_z+=((données[i][1]-  
moyenne_z[1])**2)*pourcentage_z[i]/coeff_z`

## Annexe : Code GMM 3D

```
.#La partie graphique
.a=np.linspace(-20,60,100)
.A=[]
.X=[]
.Y=[]
.X_deux=[]
.Y_deux=[]
.courbe_y,courbe_z=[],[]
.for i in range(len(a)):
    . for j in range(len(a)):
```

## Annexe : Code GMM 3D

```
.for i in range(len(courbe_y)):  
    . if courbe_y[i]<0.01*max(courbe_y):  
        . X[i],Y[i]=0,0  
.courbe_y=[elem for elem in courbe_y if elem>=0.01*max(courbe_y)]  
.X=[elem for elem in X if elem!=0]  
.Y=[elem for elem in Y if elem!=0]  
.#On enlève les points plus petits que 1% du max  
.for i in range(len(courbe_z)):  
    . if courbe_z[i]<0.01*max(courbe_z):  
        . X_deux[i],Y_deux[i]=0,0
```



# Annexe : Liens

<http://un-medecin-vous-informe.blogspot.com/2013/04/physiologie-de-la-marche.html>

<https://tel.archives-ouvertes.fr/tel-02269606/document>

<https://www.digitalsecuritymagazine.com/fr/2021/12/23/vicon-desarrolla-camara-multisensor-v1020-wir-360-grandes-espacios/>

[https://fr.wikipedia.org/wiki/Capture de mouvement#:~:text=La%20capture%20magn%C3%A9tique%20repose%20sur,capteurs%20%C2%BB%20dans%20ce%20champ%20%C3%A9lectromagn%C3%A9tique](https://fr.wikipedia.org/wiki/Capture_de_mouvement#:~:text=La%20capture%20magn%C3%A9tique%20repose%20sur,capteurs%20%C2%BB%20dans%20ce%20champ%20%C3%A9lectromagn%C3%A9tique)

<https://core.ac.uk/download/pdf/15497383.pdf>