
ETUDE DYNAMIQUE ET CHIMIQUE DE GOUTTES DE SANG POUR LOCALISER LA ZONE D'EJECTION

SOMMAIRE

I – La scène de crime

- 1) Les 2 maquettes utilisées
- 2) Confrontation des modèles

II – Récupération des traces

- 1) Le choix des gouttes
- 2) Tracé avec logiciel de pointage
- 3) Expérience du sang avec Luminol

III – Analyse des résultats

- 1) Modèles physique utilisés
- 2) Exploitation des résultats
- 3) Dans la vraie vie

IV – Annexes

- 1) Code python réalisé
- 2) Equation d'oxydoréduction
- 3) Pasandideh-Fard

La première maquette sert à vérifier que nos modèles fonctionnent. Le mannequin à échelle humaine est celui sur lequel la plupart des expériences ont été faites.

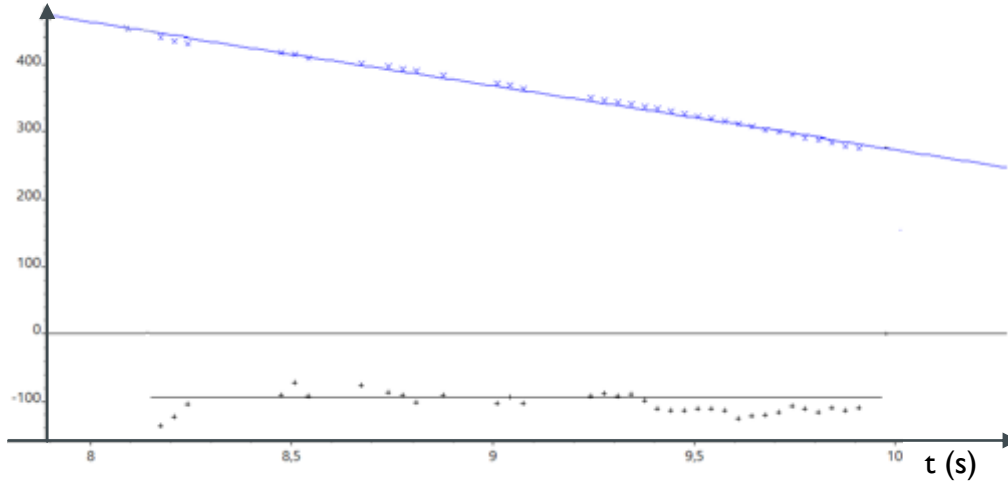


I – LA SCÈNE DE CRIME

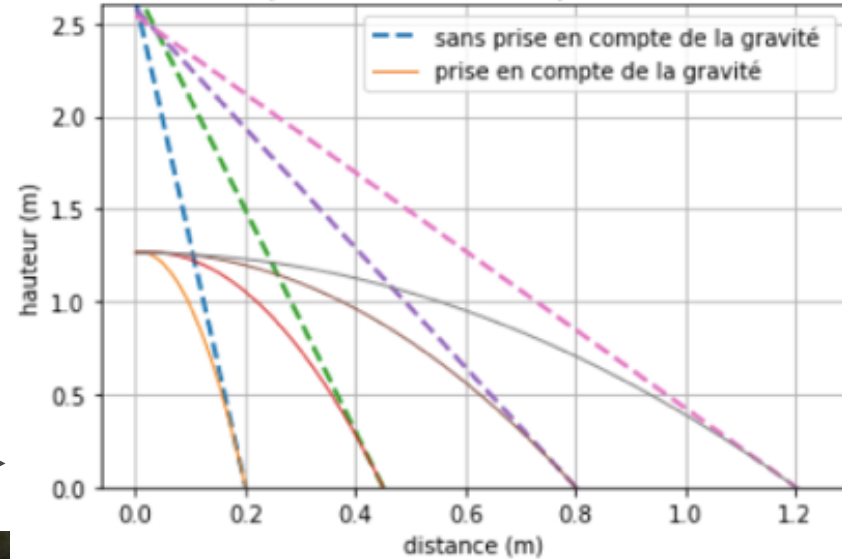
1) Les 2 maquettes utilisées

2) Confrontation des modèles

Y(mm) et Vy (m/s)



position initiale de la personne



L'équation de la courbe tracé

est:

$$ax + b$$

$$a = (-94.4 \pm 2.7) \text{ mm/s}$$

$$b = (1.2 \pm 0.025) \text{ m}$$



Le mannequin à taille humaine ne respectant plus les trajectoires rectilignes

I – LA SCÈNE DE CRIME

1) Les 2 maquettes utilisées

2) Confrontation des modèles

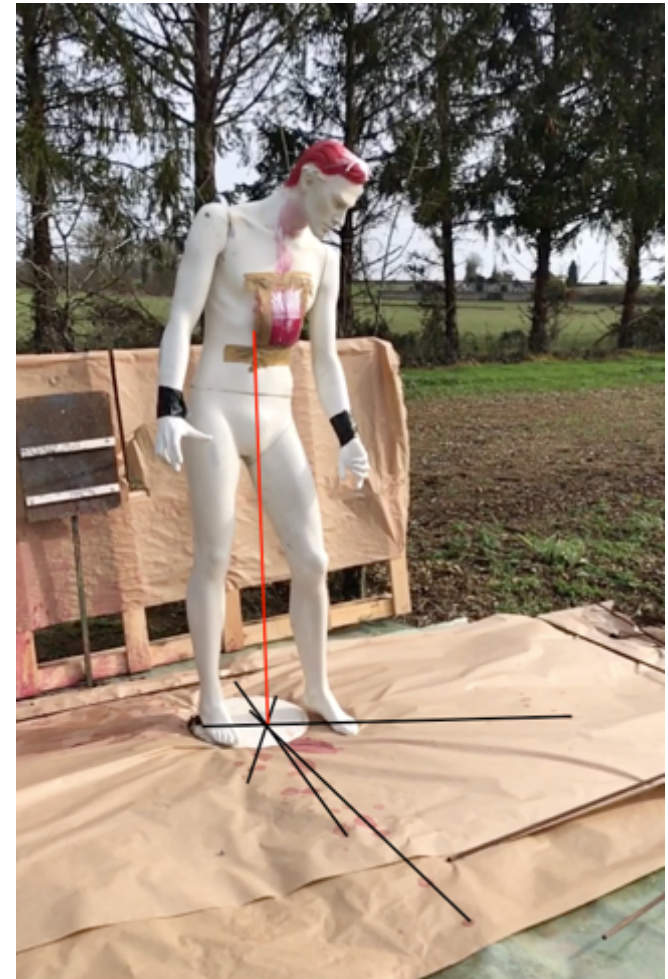
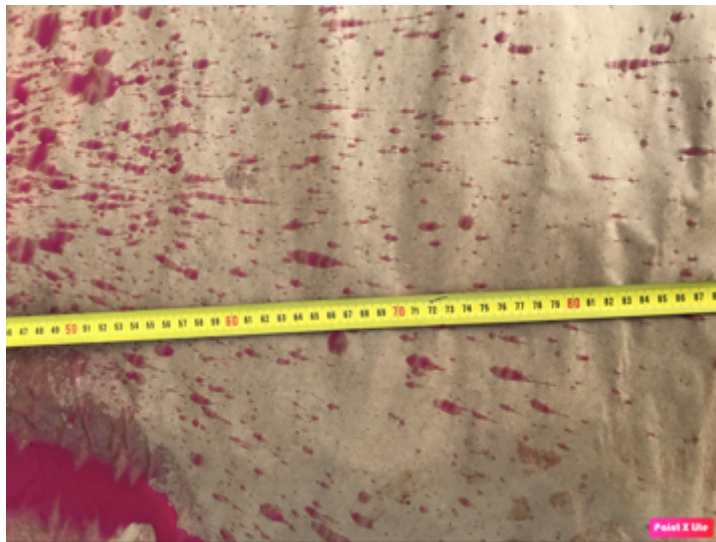


II – RÉCUPÉRATION DES TRACES

1) Le choix des gouttes

2) Tracé avec logiciel de pointage

3) Expérience du sang avec luminol

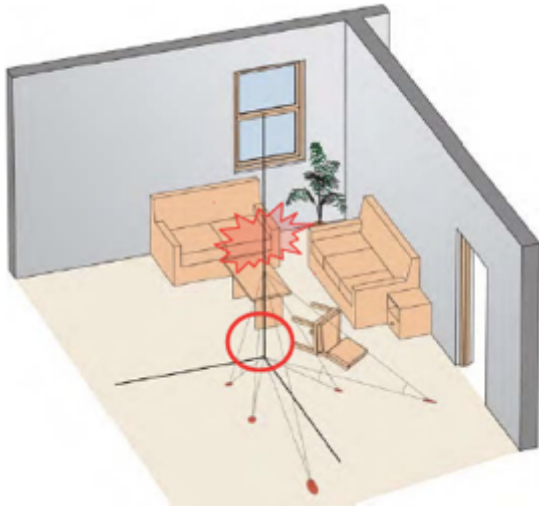


II – RÉCUPÉRATION DES TRACES

1) Le choix des gouttes

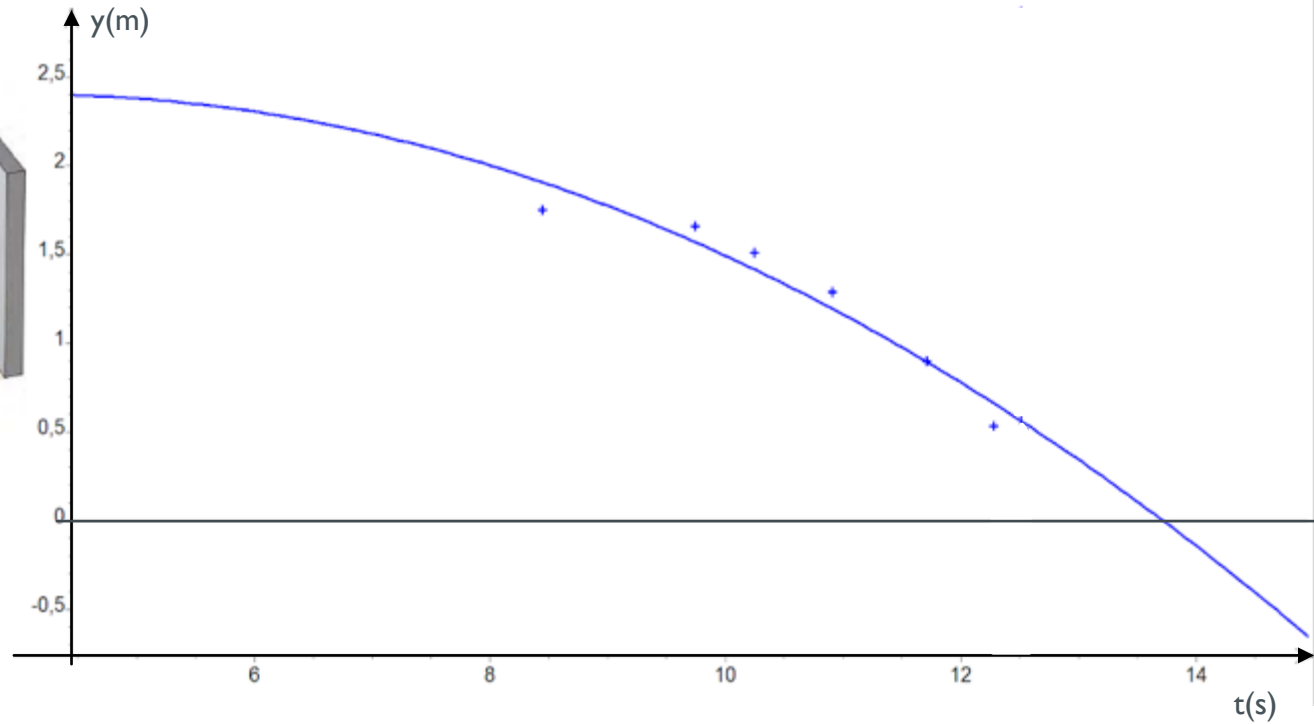
2) Tracé avec logiciel de pointage

3) Expérience du sang avec luminol



Pointage Avimeca d'une
parabole d'équation:

$$ax^2 + bx + c$$

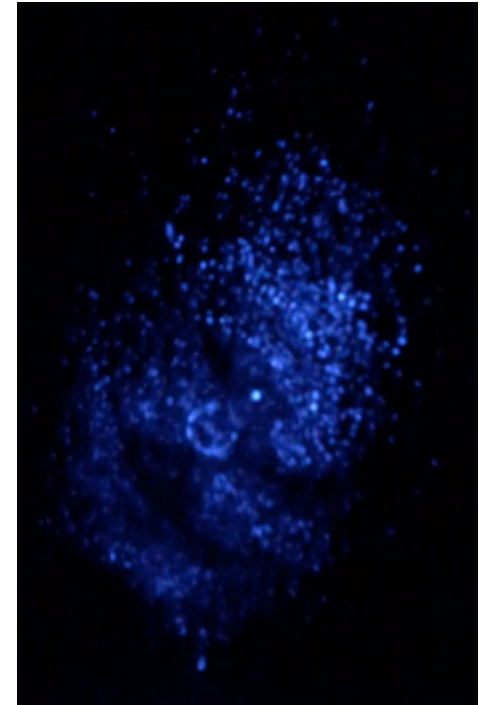
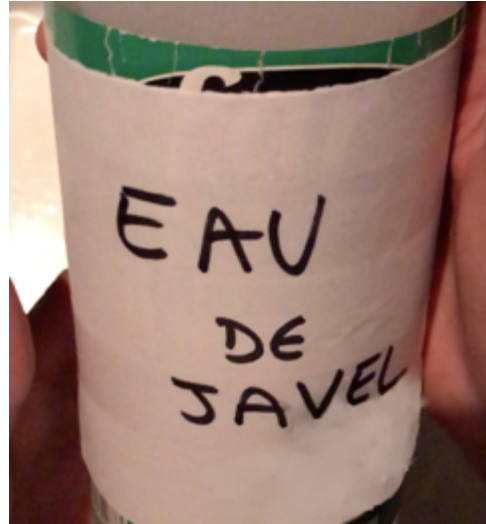


$$a = (-26 \pm 0.24) \text{ mm/s}^2$$

$$b = (-206 \pm 72) \text{ mm/s}$$

$$c = (1.98 \pm 0.24) \text{ m}$$

II – RÉCUPÉRATION DES TRACES



II – RÉCUPÉRATION DES TRACES

1) Le choix des gouttes

2) Tracé avec logiciel de pointage

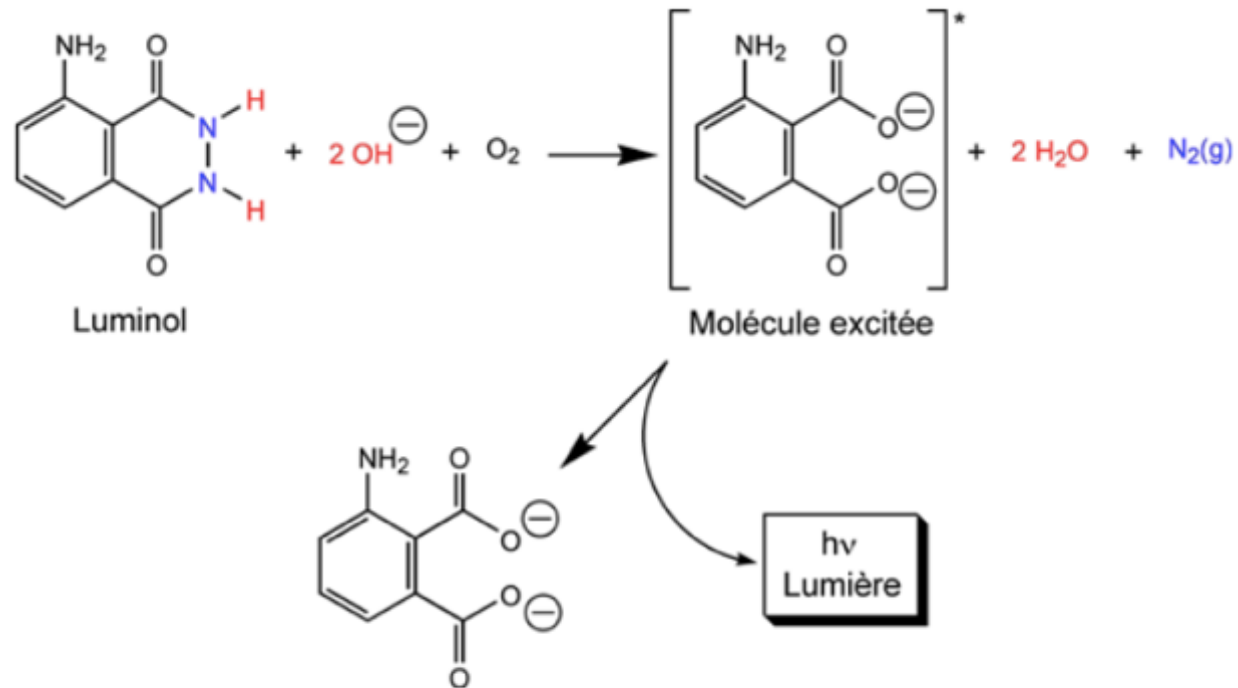
3) Expérience du sang avec luminol

Explication de la réaction

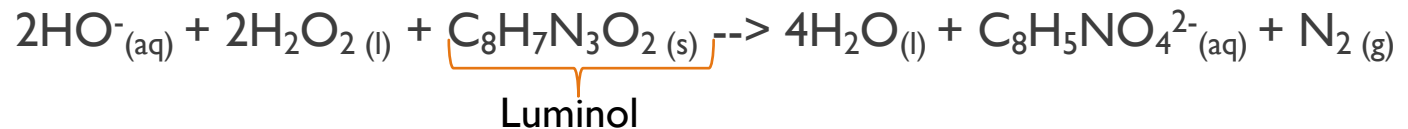
C'est une oxydation du Luminol par l'eau oxygéné

Le rôle du fer ici est de catalyser la dismutation de H_2O_2 .

Il n'apparaît pas dans l'équation de la réaction

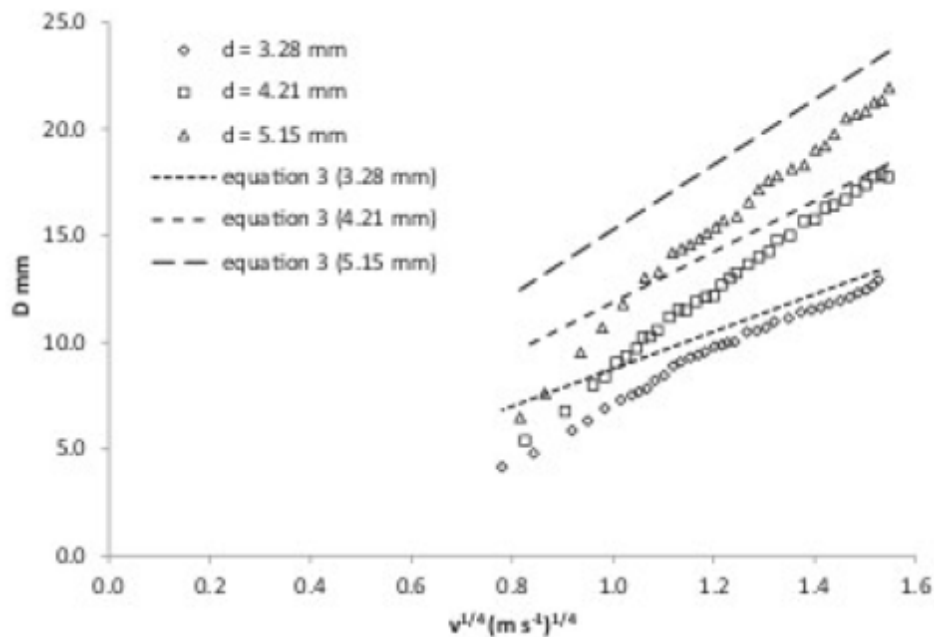


Equation de la réaction :

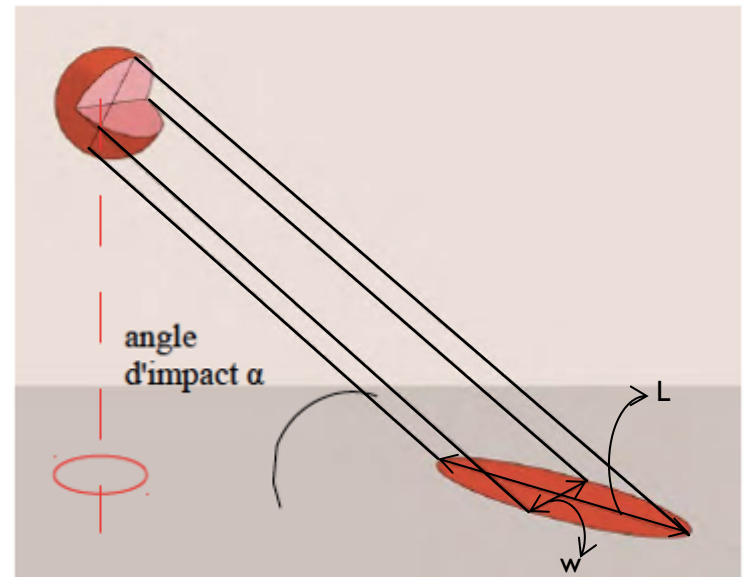


II – RÉCUPÉRATION DES TRACES

Modélisation des trajectoires des gouttes prélevées à partir de la mécanique classique, ainsi qu'avec le modèle de Pasandideh-Fard datant de 1996



$$z_i = x_f \tan \alpha - \frac{1}{2} g \frac{(x_f)^2}{v_f^2 (\cos \alpha)^2}$$



III – ANALYSE DES RÉSULTATS

$\alpha=90$



$\alpha=60^\circ$



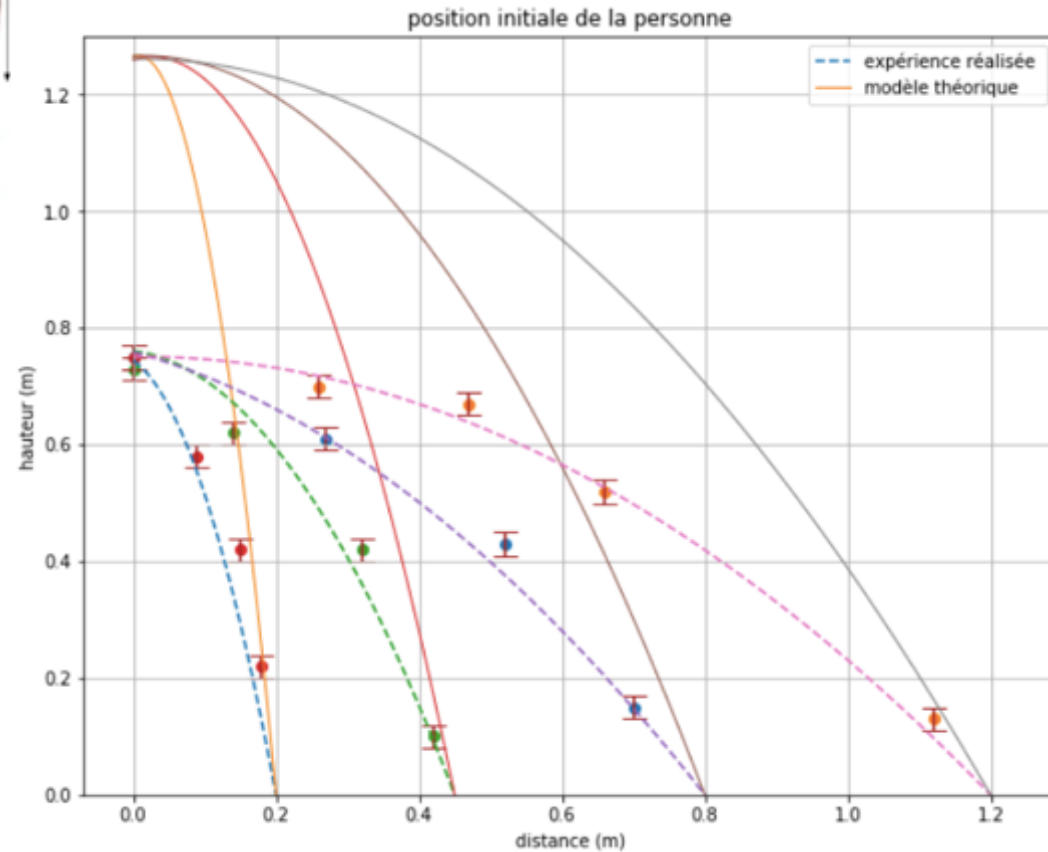
$\alpha=30^\circ$



$\alpha=20^\circ$



$$\sin(\alpha) = \frac{w}{L}$$



III – ANALYSE DES RÉSULTATS

1) Modèles physique utilisés

2) Exploitation des résultats

3) Dans la vraie vie

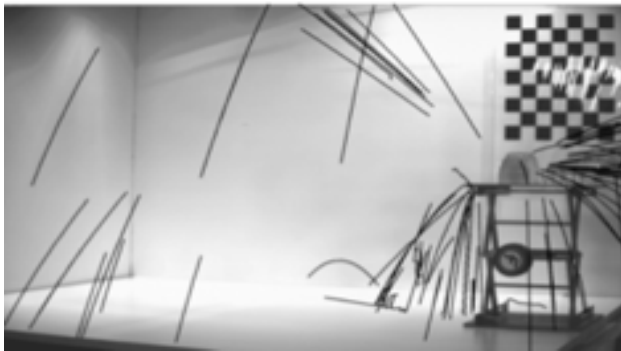
Incertitudes pour 4 gouttes à un instant t :

Les principales incertitudes sont des erreurs de lecture $Ux = Uc^2 + Ulec^2$

L'incertitudes totale sur les angles $Uy = y \sqrt{\left(\frac{Ux}{x}\right)^2 + \left(\frac{Uz}{z}\right)^2}$

Hauteur de la goutte à l'instant t (en m)	Incertitude correspondante (en m)
1.2450	0.0132
1.2311	0.0131
1.2179	0.0129
1.2045	0.0129

III – ANALYSE DES RÉSULTATS



Explication de l'écart observé
et comparaison avec La Police-Scientifique



III – ANALYSE DES RÉSULTATS

1) Modèles physique utilisés

2) Exploitation des résultats

3) Dans la vraie vie

```

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from math import *

X=np.linspace(0,1.23,200)
A=np.array([1.495,1.405,1.27,1.13])
L=np.array([0.20,0.45,0.80,1.2])
V=np.array([5,5.05,5.22,5.5])

A2=np.array([1.42,1.27,1,0.9])
L2=np.array([0.20,0.45,0.80,1.2])
V2=np.array([3.9,4.05,4.7,4.9])

def calcule_de_z1(x,i):
    return (-x)*tan(A[i])+tan(A[i])*L[i]

def calcule_de_z2(x,i):
    a=L[i]*tan(A[i])
    b=(0.5)*9.81
    c=(L[i])**2
    d=(V[i]*cos(A[i]))**2
    e=(x)**2
    f=x*tan(A[i])
    return (a-((b*c)/d)-(b*e/d)+(9.81*x*L[i]/d)-f)

def calcule_de_z3(x,i):
    a=L2[i]*tan(A2[i])
    b=(0.5)*9.81
    c=(L2[i])**2
    d=(V2[i]*cos(A2[i]))**2
    e=(x)**2
    f=x*tan(A2[i])
    return (a-((b*c)/d)-(b*e/d)+(9.81*x*L2[i]/d)-f)

for i in range (len(A)):
    Yz2=[calcule_de_z2(x,i) for x in X]
    Yz1=[calcule_de_z1(x,i) for x in X]
    Yz3=[calcule_de_z3(x,i) for x in X]
    zero=[0 for i in X]
    plt.plot(X,Yz3,linestyle='dashed')
    ##plt.plot(X,Yz1,linewidth=2,linestyle='--')
    plt.plot(X,Yz2,linewidth=1)
    ##plt.plot(zero,X,linewidth=3,color='red')

xValues = [0,0.29,0.5,0.72]
yValues = [0.77, 0.6, 0.4, 0.12]

xErrorValues = [0.02, 0.02, 0.02, 0.03]
yErrorValues = [0.03, 0.03, 0.03, 0.03]

x3Values = [0,0.16,0.3,0.41]
y3Values = [0.77, 0.65, 0.4, 0.12]

x2Values = [0.28,0.45,0.68,1.1]
y2Values = [0.72, 0.65, 0.51, 0.12]

x4Values = [0.16,0.13,0.07,0]
y4Values = [0.2, 0.4, 0.6, 0.77]

plt.scatter(xValues, yValues, zorder = 0)
plt.errorbar(xValues, yValues, yerr = yErrorValues, zorder = 0)
plt.scatter(x2Values, y2Values, zorder = 0)
plt.errorbar(x2Values, y2Values, yerr = yErrorValues, zorder = 0)
plt.scatter(x3Values, y3Values, zorder = 0)
plt.errorbar(x3Values, y3Values, yerr = yErrorValues, zorder = 0)
plt.scatter(x4Values, y4Values, zorder = 0)
plt.errorbar(x4Values, y4Values, yerr = yErrorValues, zorder = 0)

plt.title('position initiale de la personne')
plt.xlabel(r'distance (m)')
plt.ylabel(r'hauteur (m)')
plt.ylim(0,1.3)
plt.legend(('sans prise en compte de la gravité ',''))
plt.grid()
plt.show()

```

```

for i in range (len(A)):
    Yz2=[calcule_de_z2(x,i) for x in X]
    Yz1=[calcule_de_z1(x,i) for x in X]
    Yz3=[calcule_de_z3(x,i) for x in X]
    zero=[0 for i in X]
    plt.plot(X,Yz3,linestyle='dashed')
    ##plt.plot(X,Yz1,linewidth=2,linestyle='--')
    plt.plot(X,Yz2,linewidth=1)
    ##plt.plot(zero,X,linewidth=3,color='red')

xValues = [0,0.29,0.5,0.72]
yValues = [0.77, 0.6, 0.4, 0.12]

xErrorValues = [0.02, 0.02, 0.02, 0.03]
yErrorValues = [0.03, 0.03, 0.03, 0.03]

x3Values = [0,0.16,0.3,0.41]
y3Values = [0.77, 0.65, 0.4, 0.12]

x2Values = [0.28,0.45,0.68,1.1]
y2Values = [0.72, 0.65, 0.51, 0.12]

x4Values = [0.16,0.13,0.07,0]
y4Values = [0.2, 0.4, 0.6, 0.77]

plt.scatter(xValues, yValues, zorder = 0)
plt.errorbar(xValues, yValues, yerr = yErrorValues, zorder = 0)
plt.scatter(x2Values, y2Values, zorder = 0)
plt.errorbar(x2Values, y2Values, yerr = yErrorValues, zorder = 0)
plt.scatter(x3Values, y3Values, zorder = 0)
plt.errorbar(x3Values, y3Values, yerr = yErrorValues, zorder = 0)
plt.scatter(x4Values, y4Values, zorder = 0)
plt.errorbar(x4Values, y4Values, yerr = yErrorValues, zorder = 0)

plt.title('position initiale de la personne')
plt.xlabel(r'distance (m)')
plt.ylabel(r'hauteur (m)')
plt.ylim(0,1.3)
plt.legend(('sans prise en compte de la gravité ',''))
plt.grid()
plt.show()

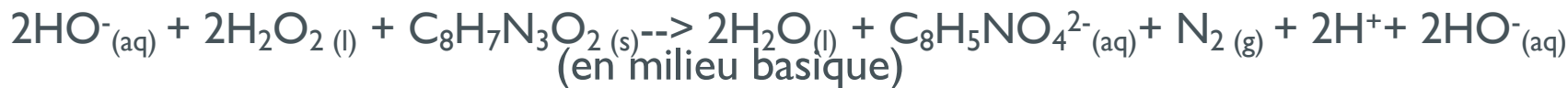
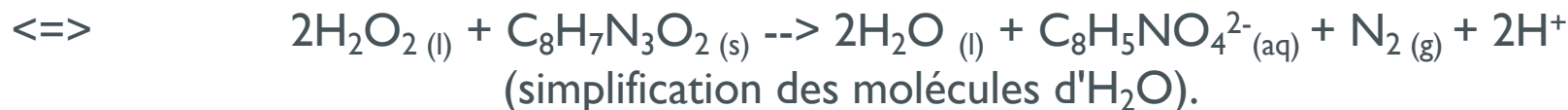
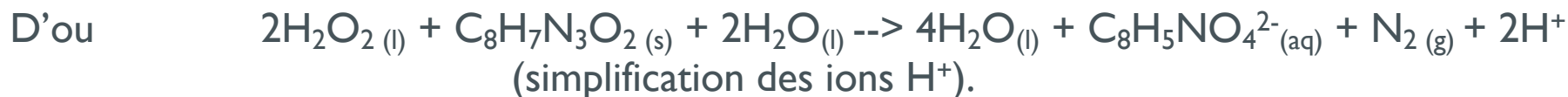
```

IV – ANNEXES

Les couples d'oxydoréduction sont :

- Pour l'eau oxygénée : $\text{H}_2\text{O}_2 (\text{l}) / \text{H}_2\text{O} (\text{l})$
- Pour le luminol : $\text{C}_8\text{H}_5\text{NO}_4^{2-} (\text{aq}) / \text{C}_8\text{H}_7\text{N}_3\text{O}_2 (\text{s})$

Les demi-équation rédox:



IV – ANNEXES

Modèle de Pasandideh-Fard

A cause de la viscosité du fluide, une partie de l'énergie cinétique initiale de la goutte va être dissipée lorsque la goutte va s'étaler sur le substrat. La variation de l'énergie mécanique de la goutte est égale au travail des forces de frottement.

$$\frac{1}{2} \left(\frac{\rho \pi d}{6} \right)^2 v^2 + \pi d^2 \gamma = \frac{\pi \gamma D^2}{4} (1 - \cos \theta_e) + \frac{\pi}{3\sqrt{Re}} d D^2 \rho v^2$$

v	est la vitesse d'impact
d	est le diamètre de la goutte
D	est le diamètre maximal du film liquide l'étalement
θ_e	est l'angle de contact
ρ	est la masse volumique du fluide
γ	son coefficient de tension superficielle

Le premier terme représente l'énergie cinétique initiale de la goutte ($E_c = 1/2 m v^2$ où $m = 4/3 \pi \rho$), le second et le troisième termes désignent respectivement la tension de surface de la goutte avant et après impact et le dernier terme, qui est le plus difficile à déterminer.

IV – ANNEXES