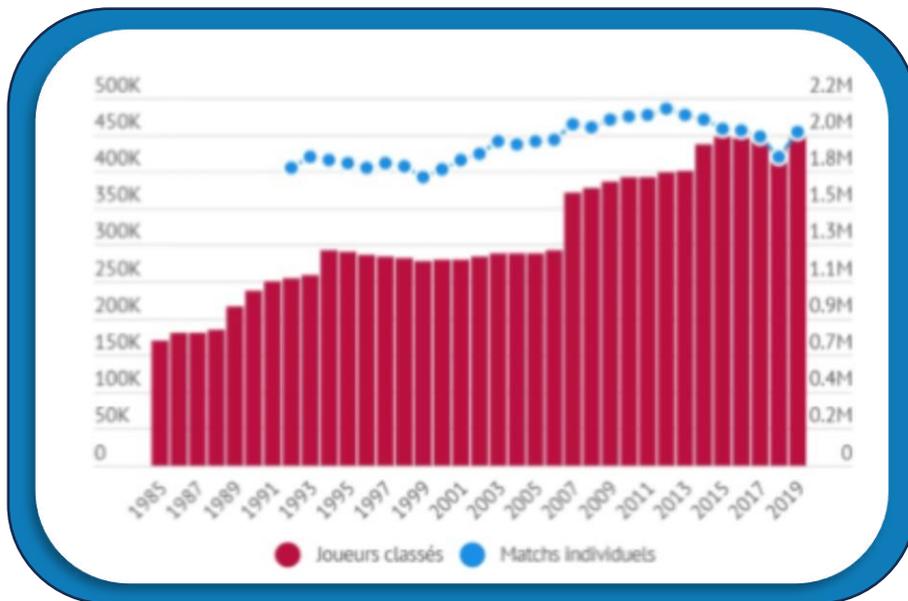
The background is a solid blue color. Two tennis rackets are depicted in a stylized, graphic manner. The racket on the left is shown from a top-down perspective, with its head and a portion of its handle visible. The racket on the right is shown from a side-on perspective, highlighting the head and the lower part of the handle. Both rackets have a blue frame and a white grid pattern representing the strings. The text is centered over the left racket.

# ÉTUDE D'UNE RAQUETTE DE TENNIS

**Martin GRANGÉ**  
**36798**

# Introduction



**Fig.1 :** Nombre de joueurs professionnels et de matchs joués, selon l'année

**- 4<sup>e</sup> activité sportive la plus pratiquée en France**

**- Enjeu de l'amélioration de l'expérience du jeu :**

- Performances
- Confort

# Problématique : Quels paramètres de la raquette de tennis permettent d'optimiser un coup droit ?

## Objectifs

### Martin Grangé

- Modéliser le système de la balle au contact de la raquette
- Étudier l'influence des paramètres de la raquette sur la vitesse de sortie de la balle

### Adrien Roy

- Modéliser la raquette comme une poutre encastrée-libre
- Étudier l'influence des paramètres de la raquette sur sa réponse vibratoire à une balle de tennis

# Sommaire

I

Influence du point d'impact et première modélisation

II

Amélioration du modèle : Prise en compte de l'ACOR

III

Influence du moment d'inertie et prise en compte de l'avant bras

IV

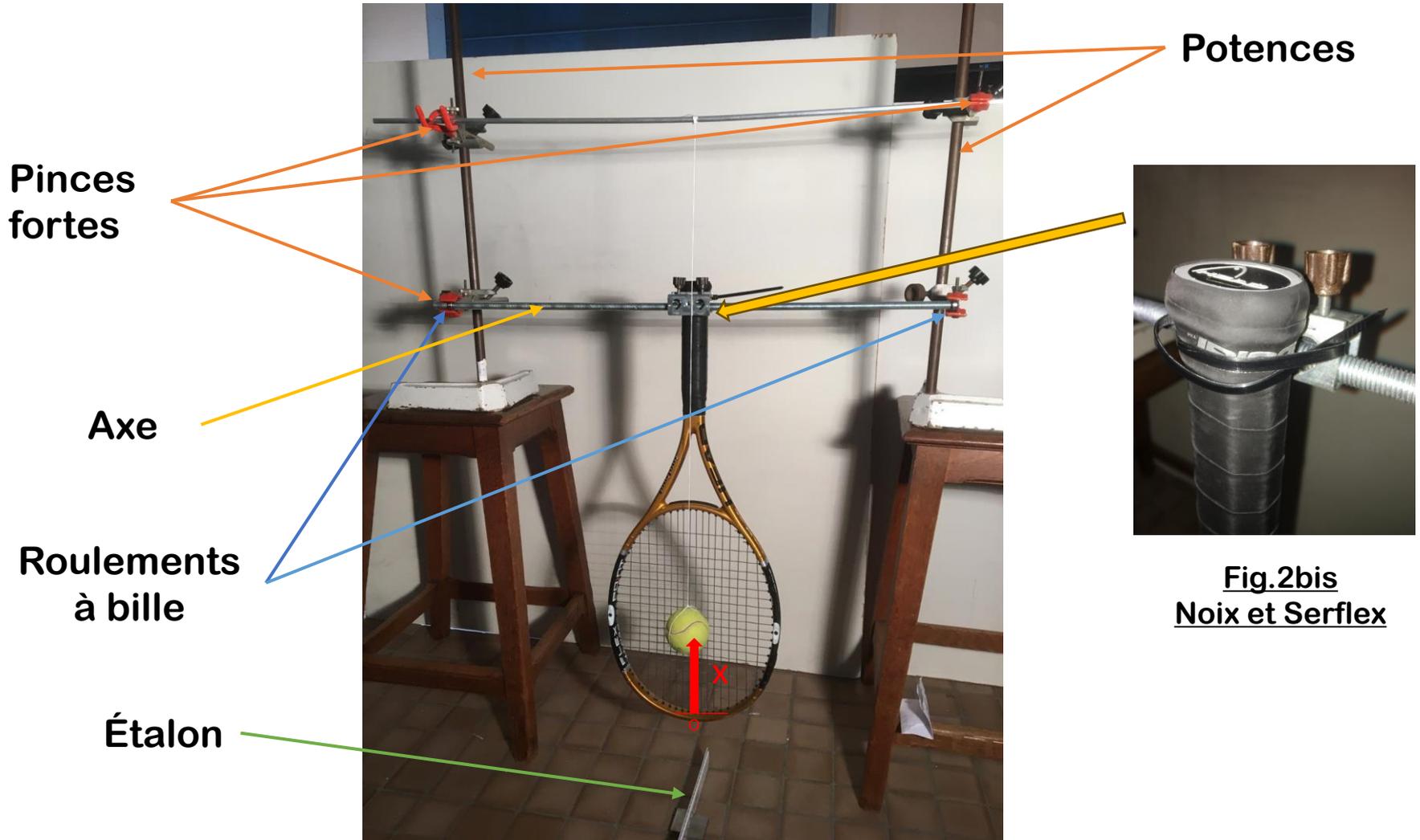
Influence du cordage sur le jeu

I - Mesure de la vitesse de sortie

II

III

IV



Potences

Pincès fortes

Axe

Roulements à bille

Étalon

Fig.2bis  
Noix et Serflex

Fig.2 Mesure de la vitesse de sortie en fonction de la position

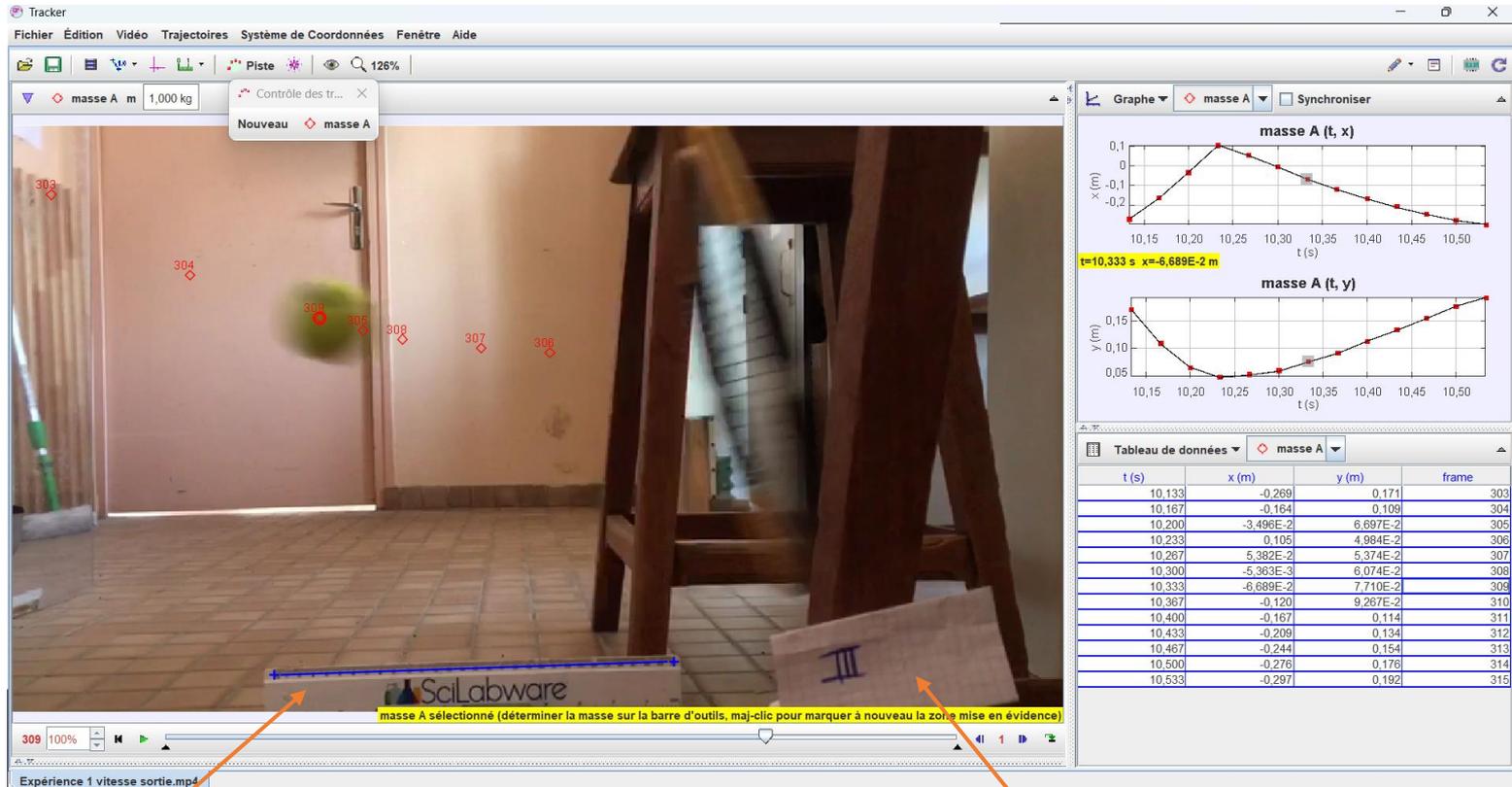
# I - Mesure de la vitesse de sortie

II

III

IV

## Acquisition des données



Étalon

Numéro de l'essai

**Fig.3 Utilisation du logiciel Tracker**

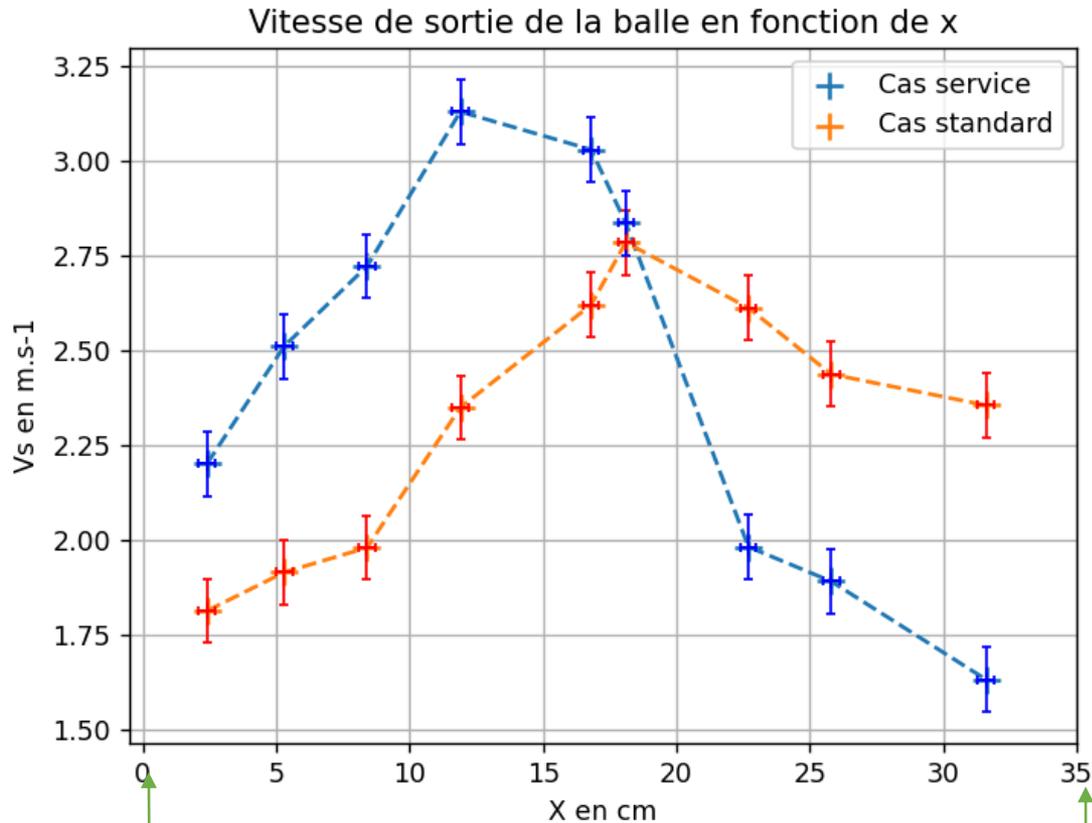
# I – Modèle du pendule simple

II

III

IV

## Résultats



- Cas service :  
Balle initialement  
au repos

- Cas standard :  
Balle avec une  
vitesse initiale



Fig.4 Résultats de l'expérience 1

# I – Modèle du pendule simple

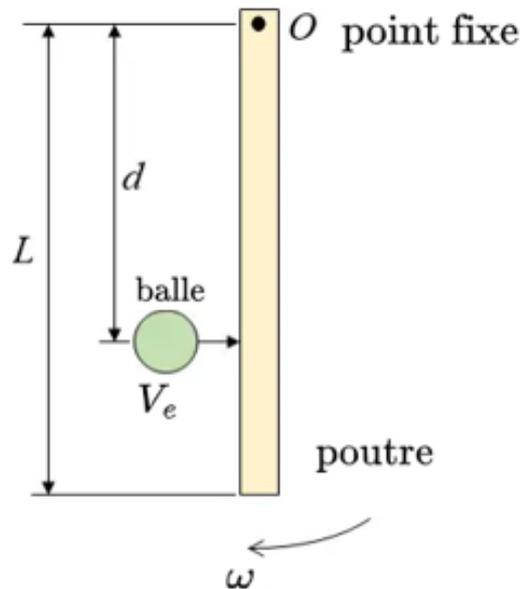
## II

## III

## IV

### Modélisation

Au point d'impact, conservation du moment cinétique



$$V_s = \frac{(V_e m d - J \omega) d - e(\omega d + V_e) J}{J + m d^2}$$

$J$  = Moment d'inertie de la raquette

$V_e$  = Vitesse d'entrée de la balle

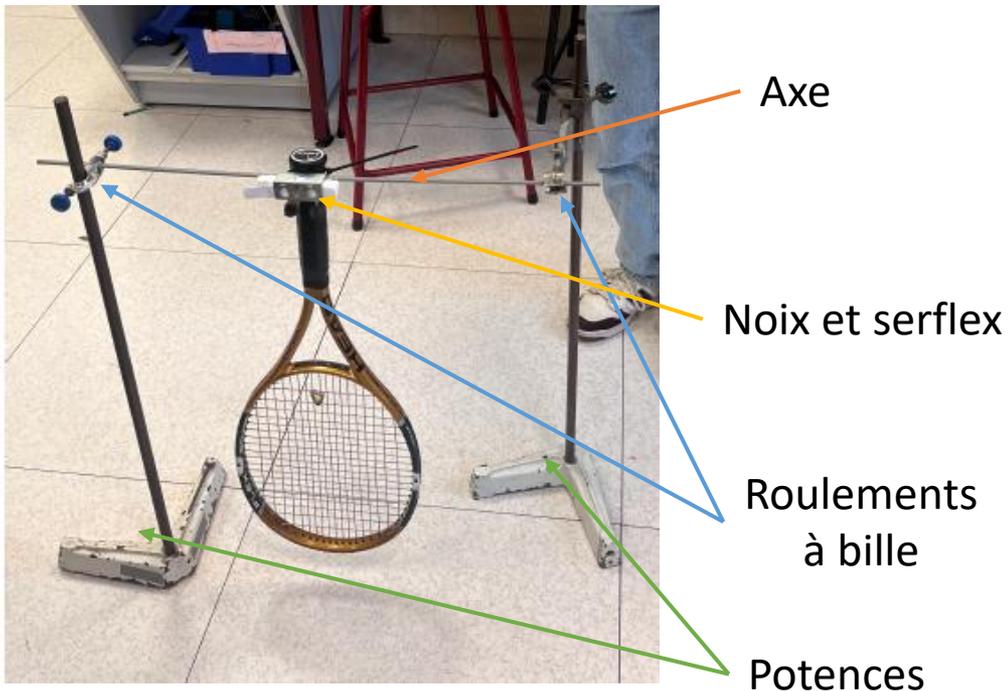
$\omega$  = Vitesse angulaire de la raquette

$e$  = ACOR: Coefficient de restitution apparent

$m$  = Masse de la balle

**Fig.5 Expérience 1 avec la raquette modélisée en poutre**

## Mesure du moment d'inertie de la raquette



**Fig.6 Dispositif de mesure des périodes d'oscillations de la raquette**

Équation du mouvement

$$J\ddot{\theta} + mgL\theta = 0$$

$$\Rightarrow w = \sqrt{\frac{mgL}{J}} = \frac{2\pi}{T}$$

$$\Rightarrow J = \frac{mgLT^2}{4\pi^2}$$

$L$  = distance axe de rotation - centre de masse

$T$  = période des oscillations

I – Modèle du pendule simple

II

III

IV

## Mesure du centre de masse de la raquette



Distance :  
extrémité du manche – centre de masse

$$L = 33 \pm 1 \text{ cm}$$

$$\Rightarrow J = \frac{mgLT^2}{4\pi^2}$$

$$J_{mes} = (4,98 \pm 0,48) \times 10^2 \text{ kg. m}^{-1}$$

Fig.7 Mesure du centre de masse

# I – Modèle du pendule simple

## II

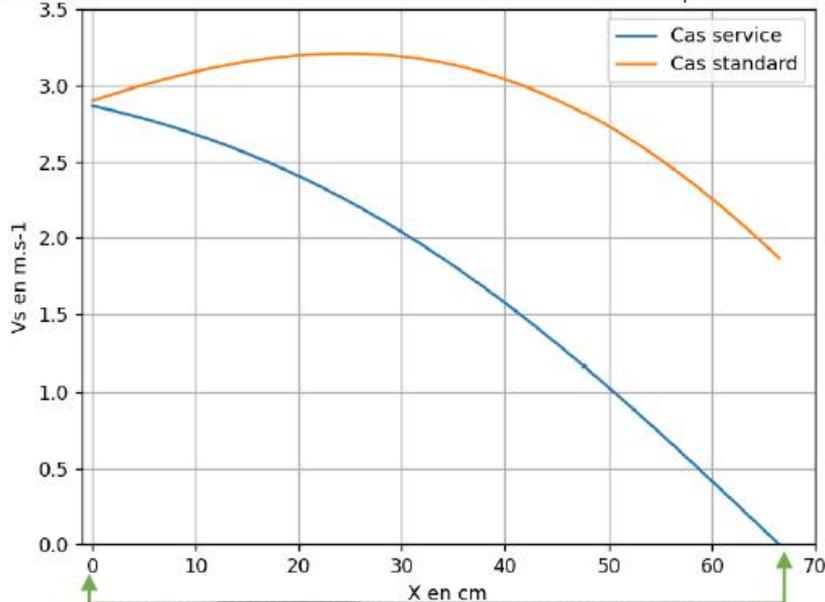
## III

## IV

## Comparaison des résultats

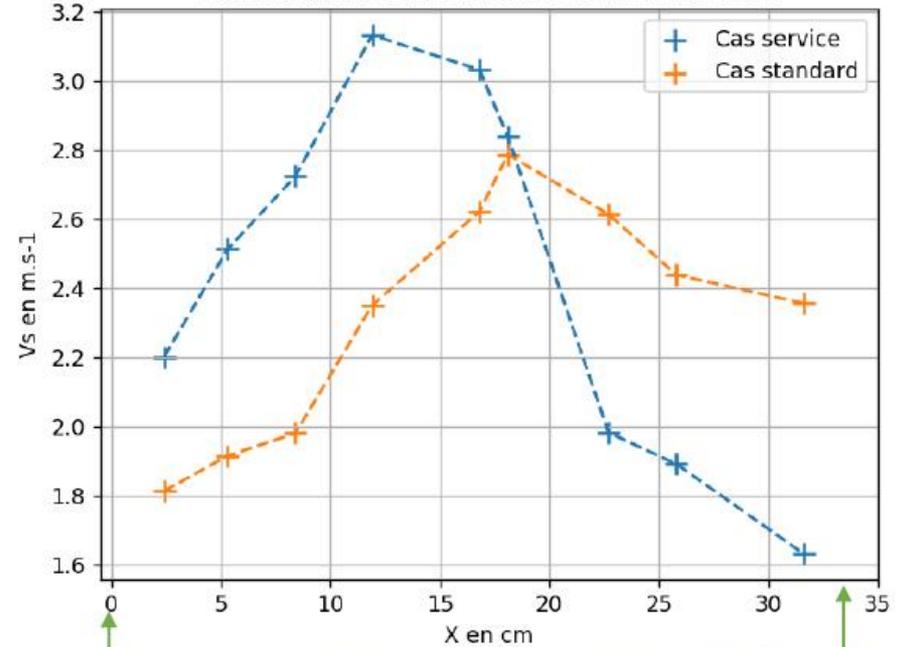
### Fig.8 Tracé théorique pour une poutre

Vitesse de sortie de la balle en fonction de la distance à la pointe de la poutre



### Fig.4 Résultats de l'expérience 1

Vitesse de sortie de la balle en fonction de  $x$



⇒ **Le modèle n'est pas adapté**

⇒ **Besoin de prise en compte des variations de l'ACOR sur le tamis**



## Mesure de l'ACOR

$$e = \frac{v_{\text{après rebond}}}{v_{\text{avant rebond}}}$$



Étalon

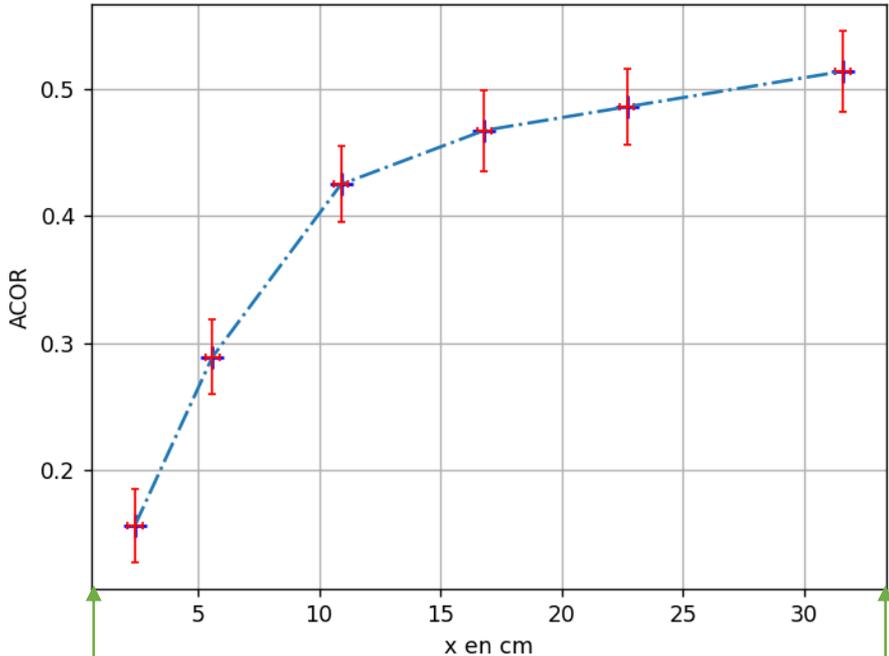
**Fig.9 : Montage mesure de l'ACOR**



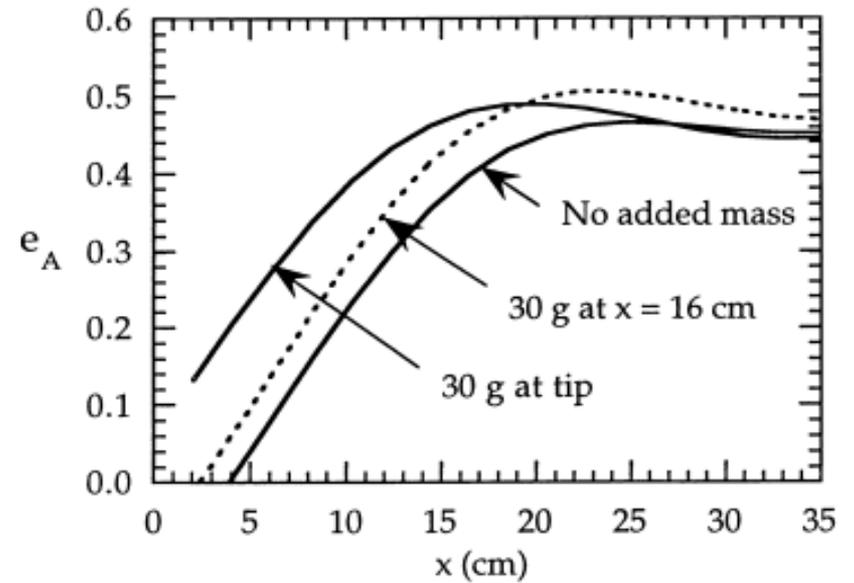
## Comparaison des résultats

**Fig.11 Résultats expérimentaux**

ACOR en fonction de la distance par rapport à la pointe de la raquette



**Fig.12 Résultats théoriques**



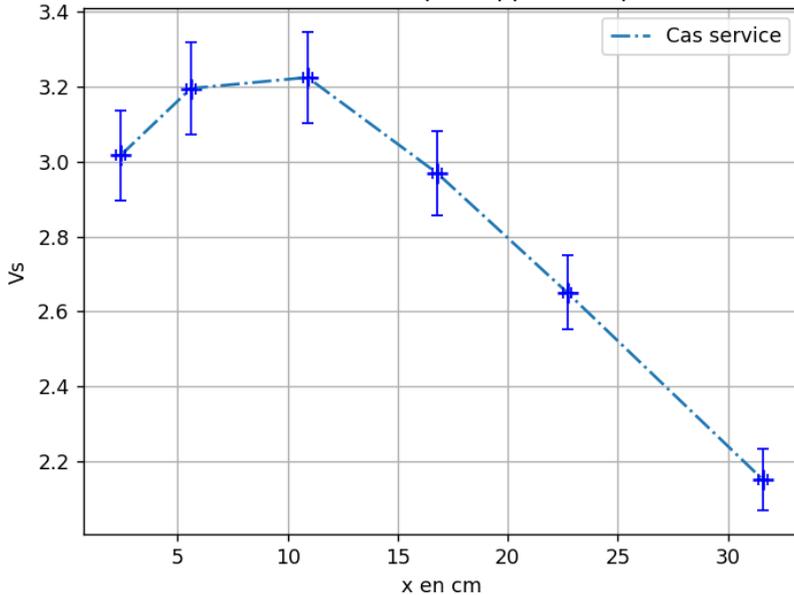
Customising a tennis racket by adding weights, Rod Cross  
Physics Department, University of Sydney, Sydney, NSW, Australia



## Modèle avec prise en compte de l'ACOR

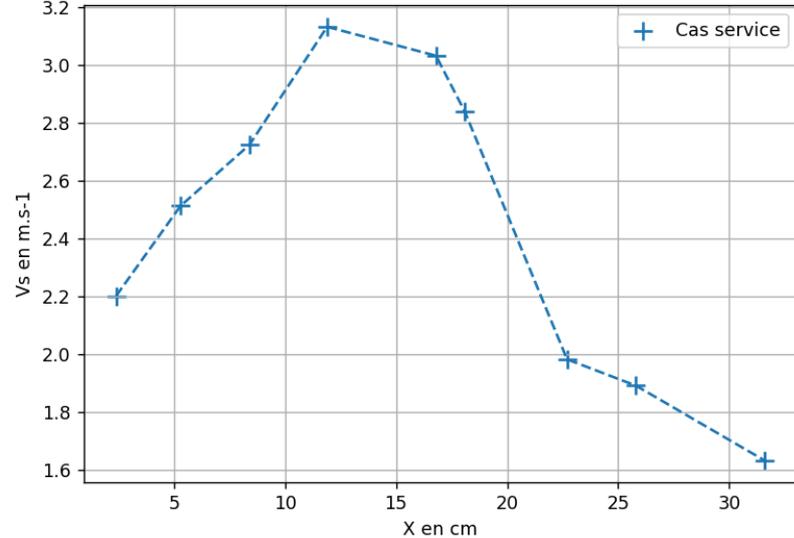
$$V_s = \omega(1 + e)(L - x)$$

Vitesse en fonction de la distance par rapport à la pointe de la raquette



**Fig.12 Nouveau modèle avec mesures expérimentales de l'ACOR**

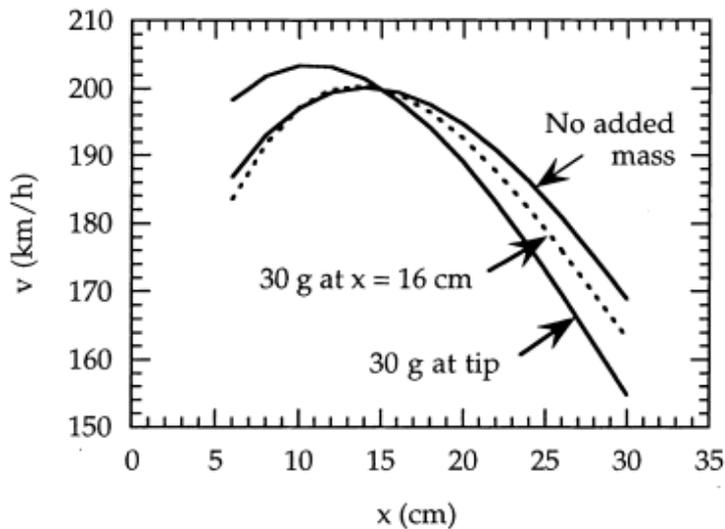
Vitesse de sortie en fonction de la distance depuis la pointe de la raquette



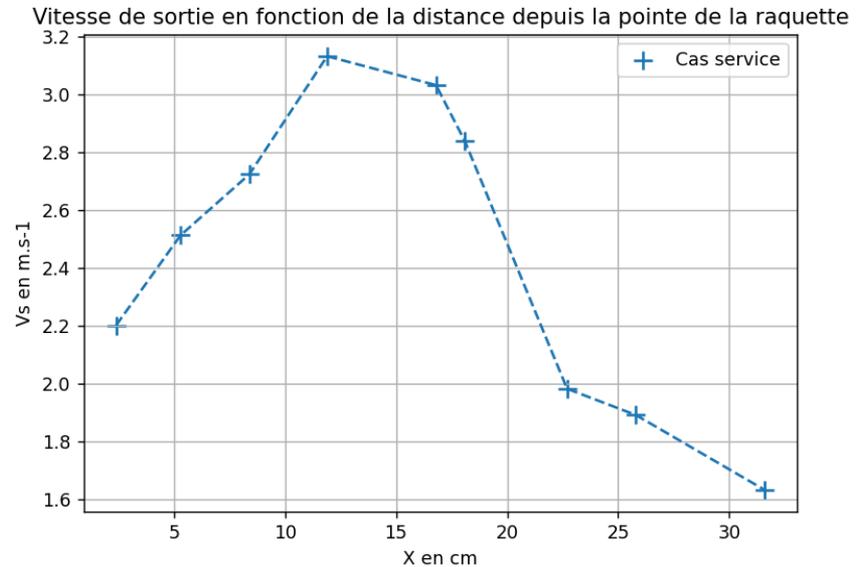
**Fig.4bis Résultat de l'expérience 1 cas du service**

## Comparaison des résultats

$$V_s = \omega(1 + e)(L - x)$$



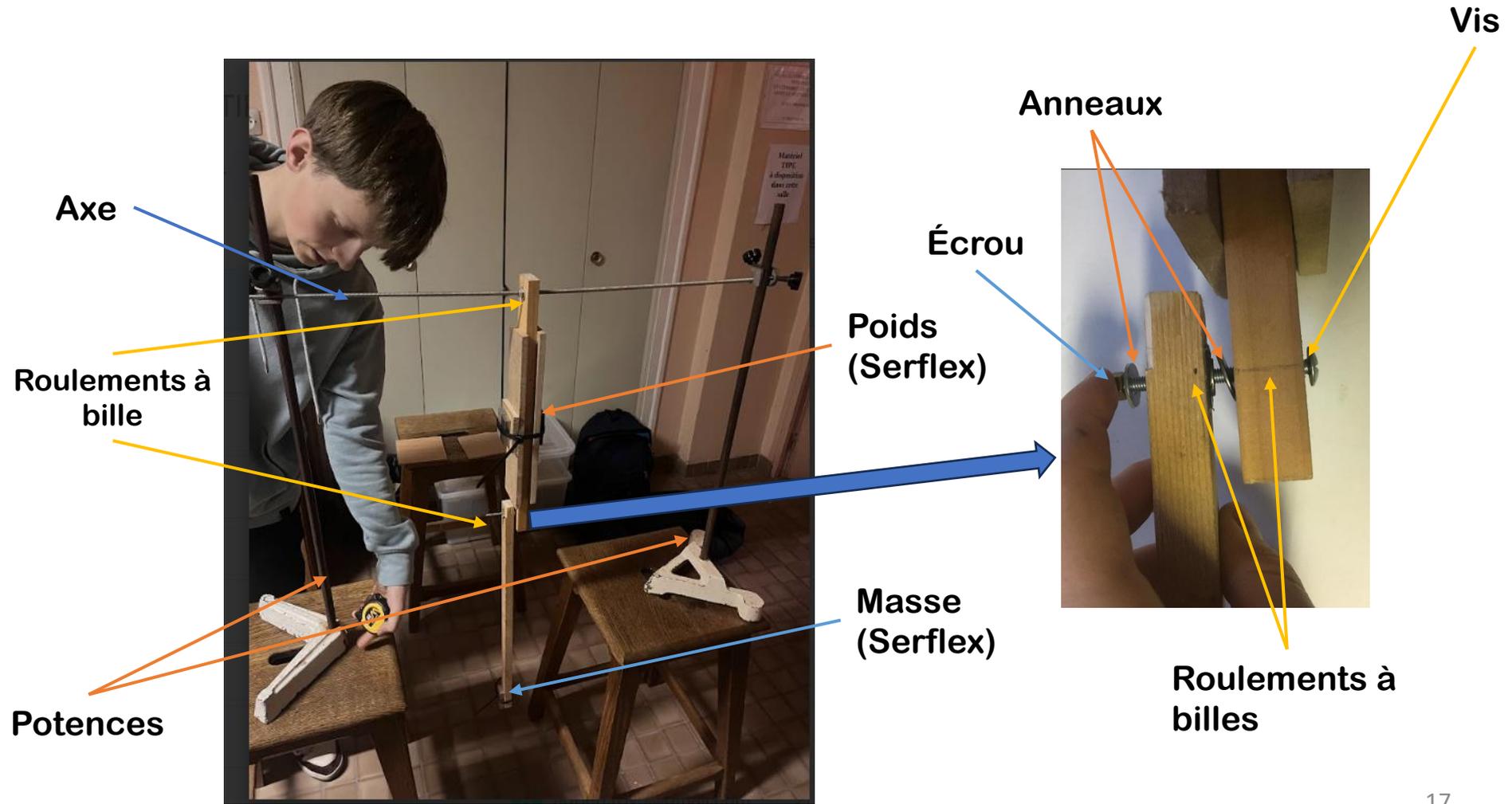
**Fig.13 Tracé du nouveau modèle avec l'ACOR théorique**



**Fig.4bis Résultat de l'expérience 1 cas du service**



**Nouveau montage : simulation de la présence de l'avant bras**



**Fig.14 Montage double pendule**

I

II

### III – Double pendule et moment d'inertie (swingweight)

IV

## Acquisition des données

Axe

Potence

masse A m 1,000 kg

Contrôle des tr...  
Nouveau masse A

Graphes masse A Synchroniser

masse A (t, x)

x (m)

t (s)

t=18,200 s x=0,624 m

masse A (t, y)

y (m)

t (s)

t=18,200 s y=-1,026 m

Tableau de données masse A

t (s)	x (m)	y (m)	v <sub>x</sub> (m/s)	v (m/s)
17,950	0,186	-1,150		
17,967	0,197	-1,150	1,430	1,442
17,983	0,234	-1,144	2,300	2,308
18,000	0,274	-1,144	2,238	2,241
18,017	0,309	-1,148	2,176	2,177
18,033	0,346	-1,146	2,114	2,136
18,050	0,379	-1,138	1,865	1,966
18,067	0,408	-1,125	1,989	2,024
18,083	0,446	-1,125	1,927	1,963
18,100	0,472	-1,113	1,803	1,888
18,117	0,506	-1,107	1,865	1,947
18,133	0,535	-1,094	1,616	1,807
18,150	0,559	-1,080	1,430	1,778
18,167	0,582	-1,059	1,368	1,767
18,183	0,605	-1,042	1,243	1,592

x=1,359 m y=-0,584 m

masse A sélectionné (déterminer la masse sur la barre d'outils, maj-clic pour marquer à nouveau la zone mise en évidence)

1092 100%

Étalon

Fig.15 Utilisation du logiciel Tracker

I

II

III – Double pendule et moment d'inertie (swingweight)

IV

## Mesures intermédiaires



Fig.16 Mesure des centres de masse

Roulements à bille

Masse (Serflex)

Axe

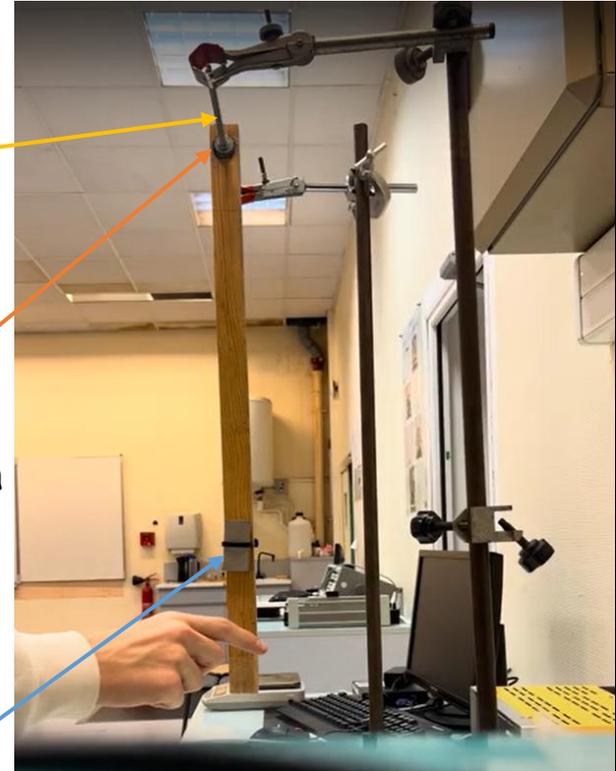
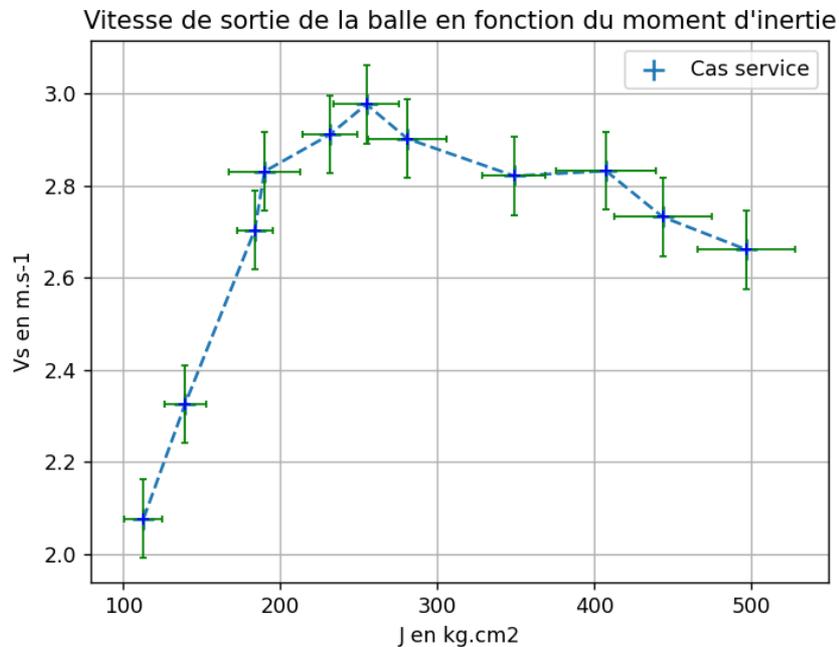
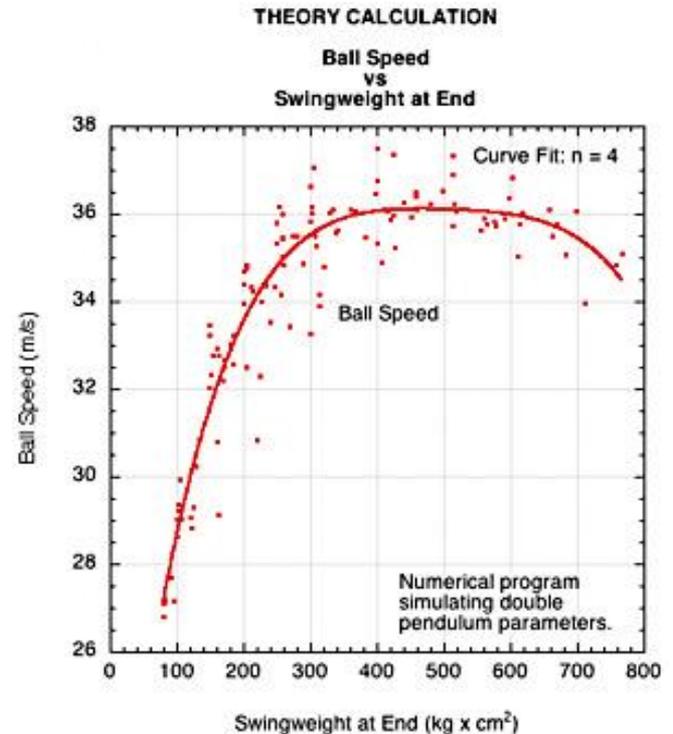


Fig.17 Mesures des moments d'inertie (swingweights)

## Comparaison des résultats

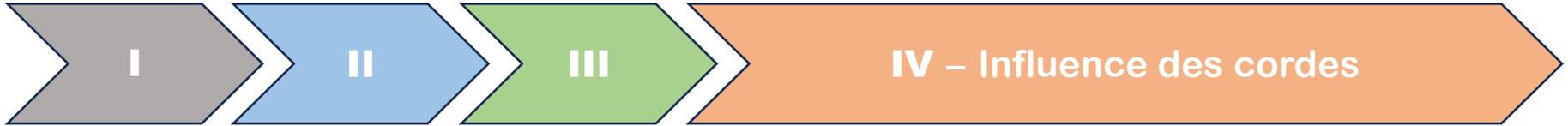


**Fig.18 Résultats expérimentaux**



**Fig.19 Résultats théoriques**

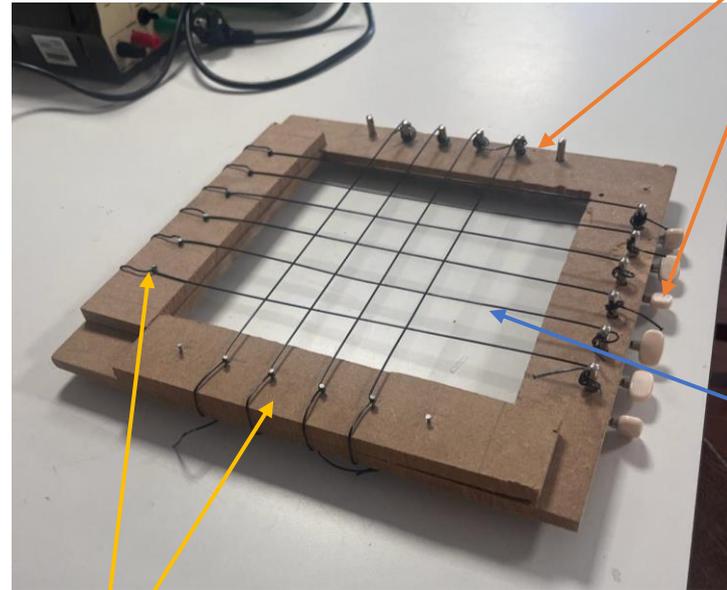
Tennis warehouse university : Effect of Customization on Swing and Ball Speed



## Création d'un cadre à cordes de tension variable



Fig.20 Fabrication du cadre

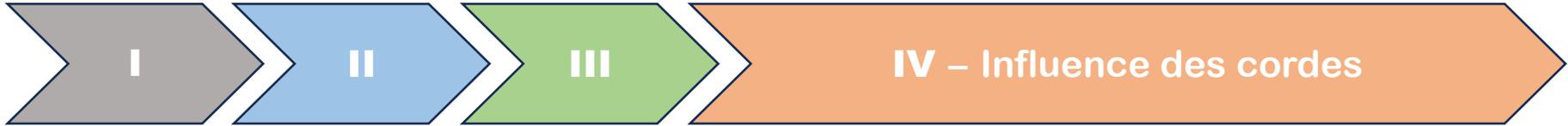


Clous

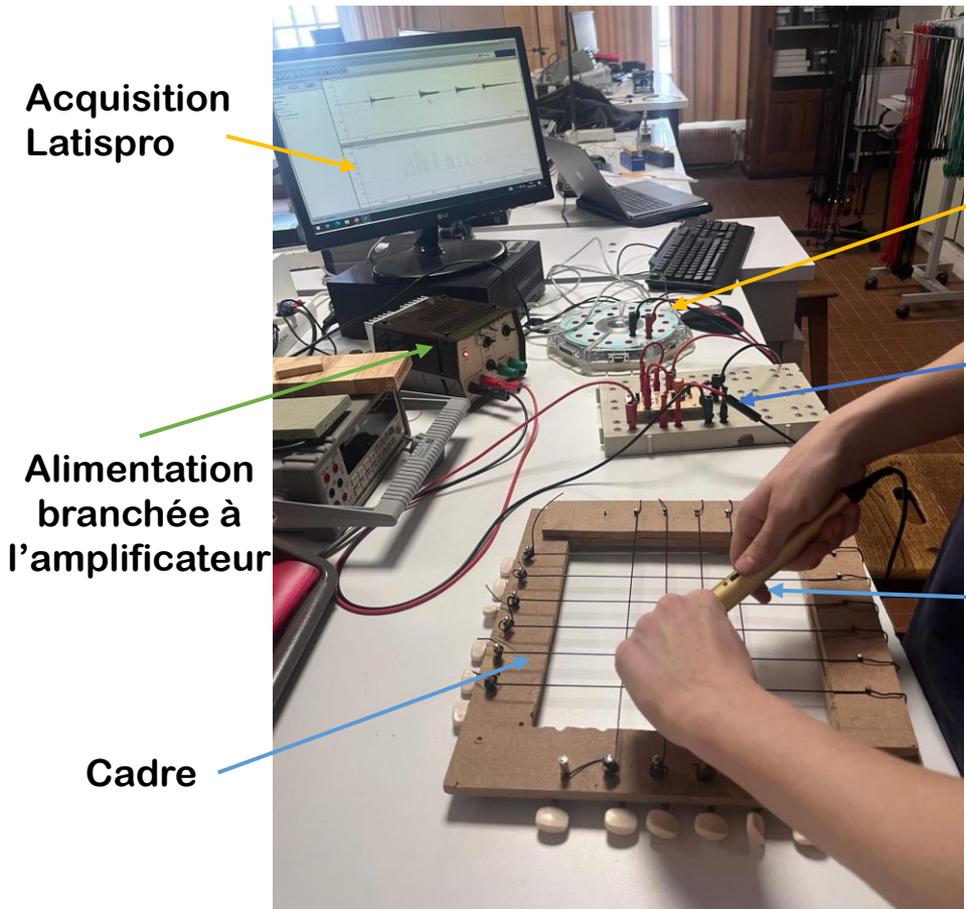
Clés de guitare

Cordage de raquette

Fig.21 Cadre achevé



## Mesure de la tension des cordes



Boitier  
d'acquisition  
Sysam

Amplificateur  
relié au micro

Micro

$$c = \sqrt{\frac{T_0}{\mu_\ell}} = f\lambda$$

$$\lambda = \frac{2L}{n}$$

$$\Rightarrow T_0 = \mu_\ell f^2 \frac{4L^2}{n^2}$$

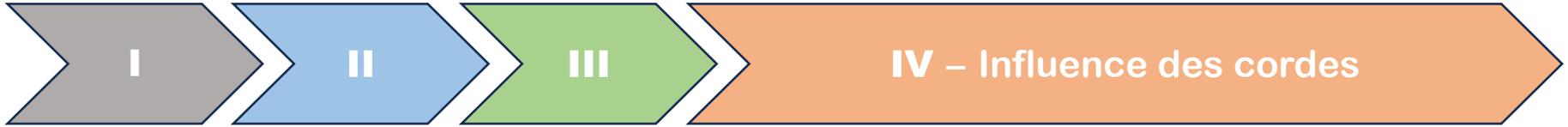
$L$  = longueur de la corde

$\mu_\ell$  = masse linéique

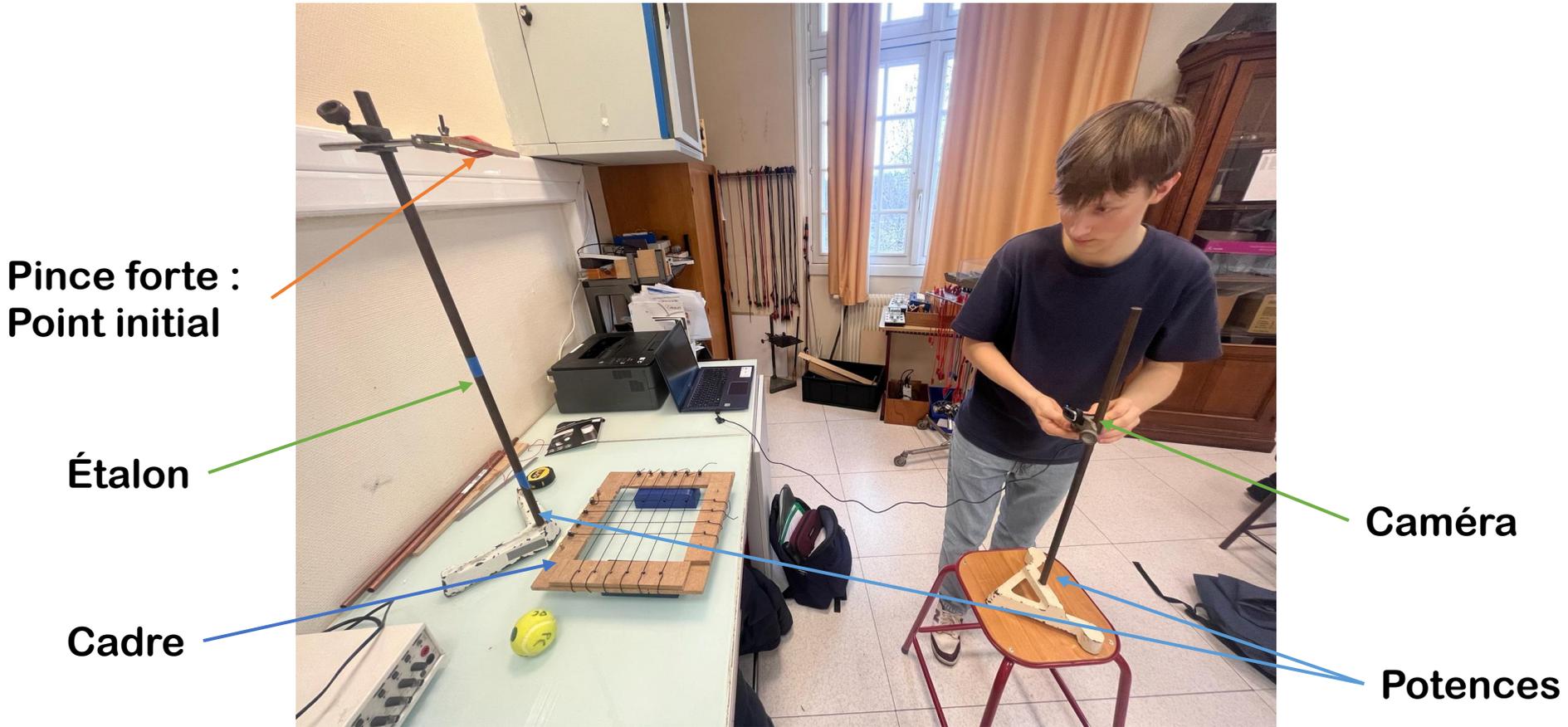
$n$  = ordre du mode propre, ici  $n=1$

$f$  = fréquence de vibration de la corde

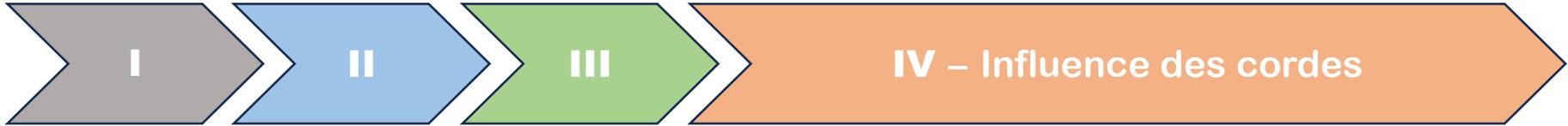
**Fig.22 Mesure de la fréquence**



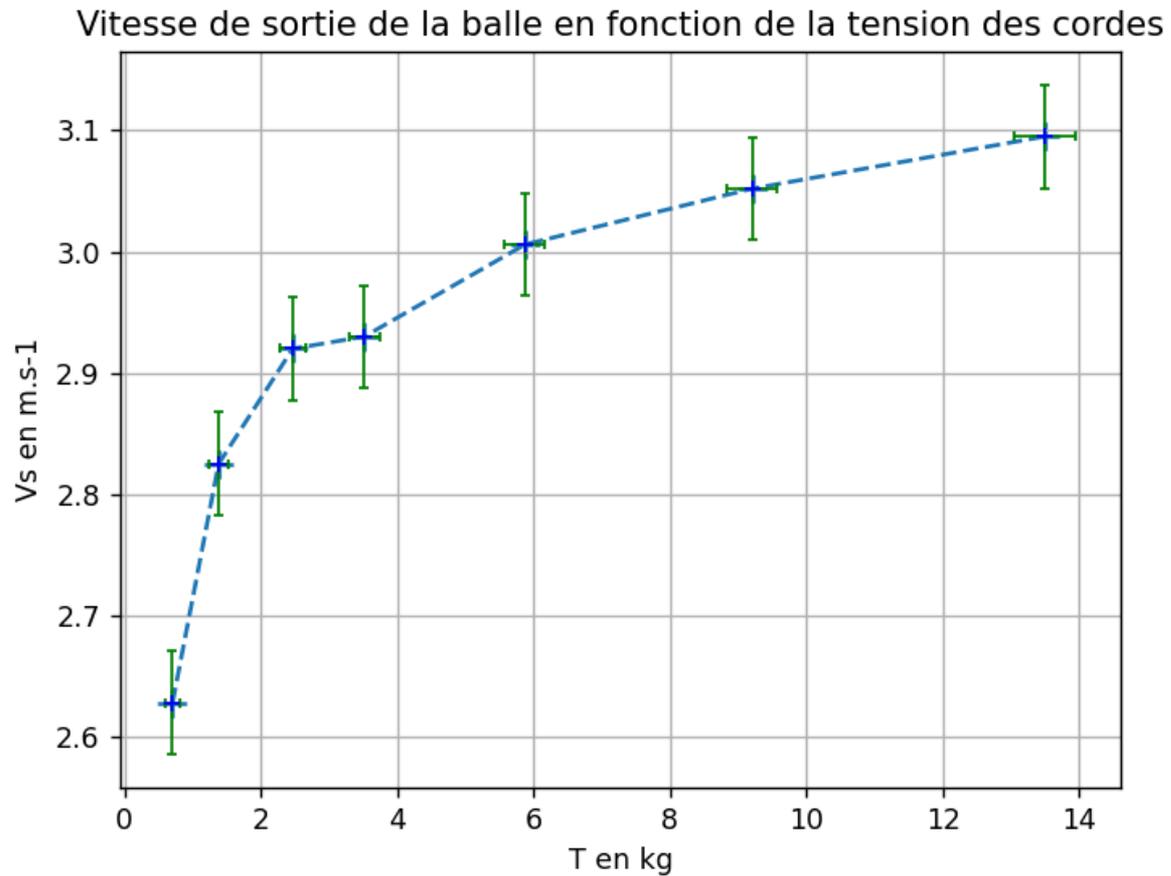
## Dispositif expérimental



**Fig.23 : Montage expérience cadre**



## Résultats



**Fig.24 Résultat expérience 4**

# Conclusion

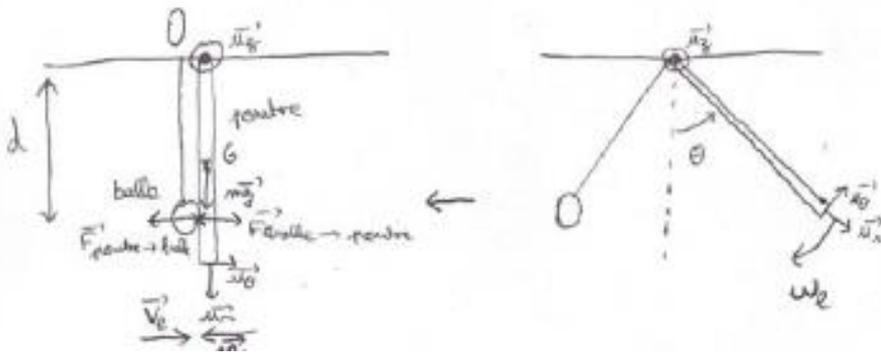
		Augmentation de la vitesse de la balle	Inconvénients	Autres avantages
Position	Frapper autour de 15 cm depuis la pointe du tamis	✓	Pas de prise en compte de l'angle d'incidence de la balle	Présence d'un nœud de vibration entre 10 et 25 cm depuis la pointe
Moment d'inertie	Augmenter le <i>swingweight</i>	✓	Existence d'un maximum Diminution de la maniabilité	
Cordage	Augmenter la tension des cordes	✓	Réduction de la durée de vie du cordage	Excitation de fréquences de résonance moins probable

- Importance du ressenti du joueur
- Modification par l'expérience du joueur

**Merci pour votre attention**

# Annexe 1 : Démonstrations

## Démonstration de l'expression de la vitesse de sortie en fonction de la position



TMC au système {balle + poutre} autour de l'axe fixe  $e_z$   
dans le réf galiléen terrestre

$$\begin{aligned} \frac{d\vec{L}_O}{dt} &= \vec{M}(m\vec{g}) + \vec{M}(\vec{F}_{\text{poutre} \rightarrow \text{balle}}) + \vec{M}(\vec{F}_{\text{balle} \rightarrow \text{poutre}}) + \vec{M}(\vec{F}_{\text{pivot}}) \\ &= \vec{OM} \wedge m\vec{g} + \vec{M}(\vec{F}_{\text{poutre} \rightarrow \text{balle}}) + \vec{M}(-\vec{F}_{\text{poutre} \rightarrow \text{balle}}) \\ &= r\vec{e}_r \wedge mg\vec{e}_r \\ &= \vec{0} \end{aligned}$$

$$\begin{aligned} \Rightarrow \vec{L}_O &= \vec{cte} \\ \Rightarrow L_{O_i} &= L_{O_f} \\ \Rightarrow mdV_e - J\omega_e &= J\omega_s + V_s md \quad (1) \end{aligned}$$

Or  $V_s - v_s = e_A(v_e - V_e)$

avec  $v$  vitesse de la poutre

$V$  vitesse de la balle

$e_A$  ACOR

$$\text{donc } V_s - \omega_s d = e_A(-\omega_e d - V_e)$$

$$\Rightarrow \omega_s = \frac{V_s + e_A(\omega_e d - V_e)}{d} \quad (2)$$

(2) dans (1) :

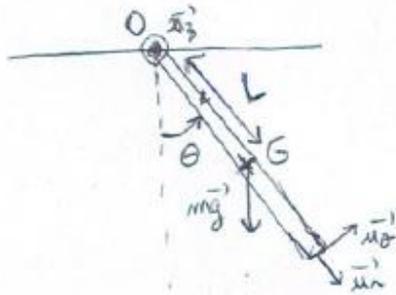
$$mdV_e - J\omega_e = \frac{JV_s}{d} + \frac{Je_A}{d}(\omega_e d + V_e) + V_s md$$

$$\Rightarrow V_s = \frac{(mdV_e - J\omega_e)d - e_A J(\omega_e d + V_e)}{md^2 + J}$$

# Annexe 1 : Démonstrations

## Démonstration de l'expression du moment d'inertie

TMC au système {poutre} dans le référentiel terrestre galiléen autour de l'axe fixe  $\vec{e}_z$



$$J\ddot{\theta} = M(\vec{P}) + M(\vec{F}_{pivot})$$
$$\Rightarrow J\ddot{\theta} = -mgL \sin(\theta)$$

Pour  $\theta \ll 1$ ,  $\sin(\theta) \simeq \theta$

$$\Rightarrow \ddot{\theta} + \frac{mgL}{J} \sin(\theta) \simeq \ddot{\theta} + \frac{mgL}{J} \theta = 0$$

$$\omega_0 = \sqrt{\frac{mgL}{J}}$$

$$\theta(t) = A \cos(\omega_0 t) + B \sin(\omega_0 t)$$
$$\dot{\theta}(0) = 0 \Rightarrow B = 0$$

$$\Rightarrow \theta(t) = \theta_0 \cos(\omega_0 t)$$

$$\omega_0 = \sqrt{\frac{mgL}{J}} = \frac{2\pi}{T}$$
$$\Rightarrow J = mgL \frac{T^2}{4\pi^2}$$

# Annexe 2

## Utilisation logiciel tracker : mesures sur images

Tracker

Fichier Édition Vidéo Trajectoires Système de Coordonnées Fenêtre Aide

ruban A Ruler Increment 0: longueur à l'échelle 0,311 m angle du ruban -2,8°

Graphe ruban A Synchroniser

ruban A (t, L)

L (m)

t (s)

t=0,000 s L=0,311 m

ruban A (t,  $\theta$ )

$\theta$

t (s)

t=0,000 s  $\theta$ =-2,781

Tableau de données ruban A

t (s)	L (m)	$\theta$
0,000	0,311	-2,8°
0,100	0,311	-2,8°
0,200	0,311	-2,8°
0,300	0,311	-2,8°
0,400	0,311	-2,8°
0,500	0,311	-2,8°
0,600	0,311	-2,8°
0,700	0,311	-2,8°
0,800	0,311	-2,8°
0,900	0,311	-2,8°

x=0,160 y=-8,805E-3

ruban A sélectionné (définir la longueur pour modifier l'échelle, définir l'angle pour modifier l'orientation de l'axe des x)

Distance balle raquette II .jpg

Numéro de l'essai

Étalon

# Annexe 3 :

## Codes Python

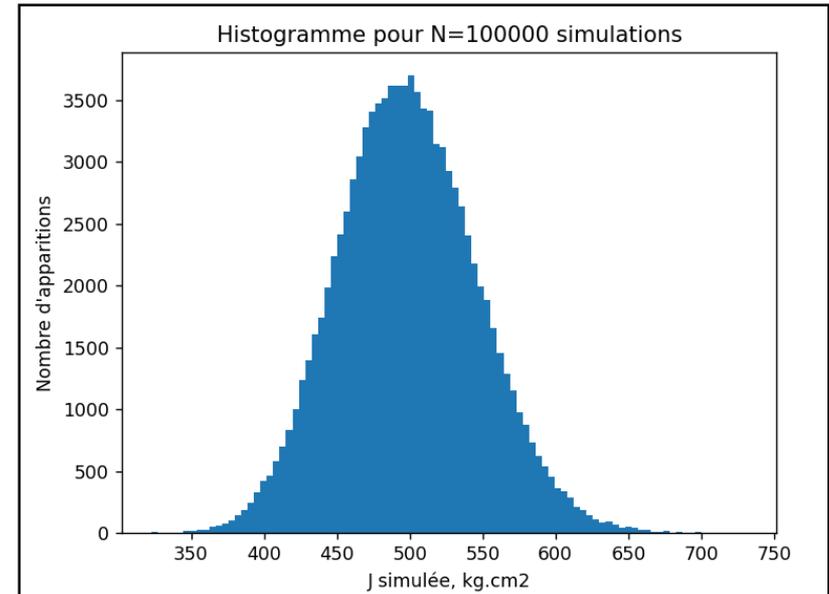
```
1  ## Importation des bibliothèques
2
3  from math import *
4  import matplotlib.pyplot as plt
5  import numpy as np
6
7  ## Données
8
9  u_X=0.3 #cm
10 u_x = 2*10**(-3) #m = incertitude sur le pointage
11 dt = 1/30 #s = intervalle de temps entre 2 frames
12 u_Vs = 2**0.5*u_x/(dt) #m.s-1 = incertitude sur la vitesse
13
14 # CAS 1 : Service (balle au repos)
15
16 X=np.array([2.4,5.3,8.4,11.9,16.8,18.1,22.7,25.8,31.6]) # distance à partir du bout de la raquette en cm
17 Vs1=np.array([2.201,2.512,2.723,3.131,3.031,2.837,1.982,1.891,1.632]) # mesures
18
19 # CAS 2 : Coup standard (balle en mouvement)
20
21 Vs2 = np.array([1.813,1.915,1.980,2.35,2.621,2.786,2.613,2.438,2.356]) # mesures
22
23
24 ## Tracé
25
26
27 plt.scatter(X,Vs1,s=100,marker='+',label='Cas service')
28 plt.plot(X,Vs1,linestyle='--')
29
30 plt.scatter(X,Vs2,s=100,marker='+',label='Cas standard')
31 plt.plot(X,Vs2,linestyle='--')
32 plt.errorbar(X,Vs2,xerr=u_X,yerr=u_Vs,fmt='r+',linewidth=1,capsize=2)
33 plt.errorbar(X,Vs1,xerr=u_X,yerr=u_Vs,fmt='b+',linewidth=1,capsize=2)
34
35 plt.legend()
36 plt.ylabel('Vs en m.s-1')
37 plt.xlabel('X en cm')
38 plt.xlim(-0.5,35)
39 plt.title('Vitesse de sortie de la balle en fonction de x')
40 plt.grid()
41 plt.show()
42
```

**Tracé des résultats de  
l'expérience de la vitesse de  
sortie en fonction de la  
position**

# Annexe 3 :

## Codes Python

```
1  ## Bibliothèques
2
3  import numpy as np
4  from math import *
5  import matplotlib.pyplot as plt
6  import numpy.random as rd
7
8
9  ## Données
10
11  N=100000
12
13  p_m=0.1 #g = précision balance
14  u_T = 1/16 #s = incertitude sur la période
15  u_L = 1 #cm = incertitude sur les centres de masse
16
17
18  L= 33 #cm = distance du centre de masse
19  T= 22/16 #s = période des oscillations
20  g= 9.81 #m.s-2 = accélération de la pesanteur
21  m= 320.9 #g = masse de la raquette
22
23
24  ## Simulation
25
26  T_sim= T + u_T*rd.normal(0,1,N)
27  L_sim = L + u_L*rd.normal(0,1,N)
28  m_sim = m + rd.uniform(-p_m,p_m,N)
29
30  Jcalc_sim = (m_sim*10**(-3)*g*10**2*L_sim)*T_sim**2/(4*pi**2)
31
32
33  J, u_J = np.mean(Jcalc_sim), np.std(Jcalc_sim, ddof=1)
34
35  ## Tracé
36
37
38  plt.hist(Jcalc_sim, bins='rice')
39  plt.title('Histogramme pour N=100000 simulations')
40  plt.xlabel('J simulée, kg.cm2')
41  plt.ylabel("Nombre d'apparitions")
42  plt.show()
43
44  print("J =", format(J, "#.4f"), "kg.cm2")
45  print('u_J =', format(u_J, "#.1f"), 'kg.cm2')
46
```



$J = 498.6588 \text{ kg.cm}^2$   
 $u_J = 47.7 \text{ kg.cm}^2$

**Incertitudes et résultats sur  
la mesure du moment  
d'inertie de la raquette**

# Annexe 3 : Codes Python

## Tracé de la vitesse de sortie théorique en fonction de la position (modèle 1)

```
1 from math import *
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5
6
7 d1 = np.linspace(0,66.5,100)
8 d = d1*0.01
9
10 def tracer(J,V,m,e,w,j,L):
11     plt.figure(1)
12     V0 = -((V*m*d-J*w)*d-e*(w*d+V)*J)/(J+m*d**2)
13     plt.plot((0.665-d)*100,V0,linestyle='-', label=L[j-1])
14
15 L=['Cas standard','Cas service','Raquette initialement au repos']
16 # Cas 1
17
18 J = 0.045 # kg.m**2 moment d'inertie de la poutre
19 V1 = 3.734 # m.s-1 vitesse d'arrivée de la balle
20 m = 0.04949 # kg masse de la balle
21 e = 0.5 # ACOR coefficient de restitution
22 w1 = 4.27 # s-1 vitesse angulaire de la raquette
23
24 # Cas 2
25
26 V2 = 0 # m.s-1 vitesse d'arrivée de la balle
27 w2 = 4.27 # s-1 vitesse angulaire de la raquette
28
29 # Cas 3
30
31 V3 = 3.734 # m.s-1 vitesse d'arrivée de la balle
32 w3 = 0 # s-1 vitesse angulaire de la raquette
33
34 # Test
35
36
37
38 tracer(J,V2,m,e,w2,2,L)
39 tracer(J,V1,m,e,w1,1,L)
40 #tracer(J,V3,m,e,w3,3,L)
41
42 plt.legend()
43 plt.ylim(0,3.5)
44 plt.xlim(-1,70)
45 plt.ylabel('V0 en m.s-1')
46 plt.xlabel('d en cm')
47 plt.title('Vitesse de sortie de la balle en fonction de la distance à la pointe de la poutre')
48 plt.grid()
49 plt.show()
```

```
1 ## Importation des bibliothèques
2
3 from math import *
4 import matplotlib.pyplot as plt
5 import numpy as np
6
7 ## Données
8
9 u_X=0.3
10 u_x=2*10**-3 #m = incertitude pointage
11 dt=1/30 #s = intervalle entre les frames
12 u_V=2**0.5*u_x/dt #m.s-1
13
14 X=np.array([2.4,5.6,10.9,16.8,22.7,31.6]) # distance à partir du bout de la raquette
15     en cm
16 Ve=np.array([2.956,2.990,3.077,2.917,3.101,2.987]) #vitesses d'entrée
17 Vs=np.array([0.461,0.864,1.308,1.364,1.508,1.536]) #vitesses de sortie
18 e=Vs/Ve
19 n=len(e)
20
21 u_e=np.zeros(n)
22 for i in range (n):
23     u_e[i]=e[i]*((u_V/Vs[i])**2+(u_V/Ve[i])**2)**0.5
24
25 ## Tracé
26
27 plt.scatter(X,e,s=100,marker='+',c='b')
28 plt.errorbar(X,e,xerr=u_X,yerr=u_e,fmt='r+',linewidth=1,capsize=2)
29 plt.plot(X,e,linestyle='-.')
30 plt.ylabel('ACOR')
31 plt.xlabel('x en cm')
32 plt.title('ACOR en fonction de la distance par rapport à la pointe de la raquette')
33 plt.grid()
34 plt.show()
```

## Tracé de l'ACOR en fonction de la position avec incertitudes (composées)

# Annexe 3 : Codes Python

```
1  ## Bibliothèques
2
3  from math import *
4  import matplotlib.pyplot as plt
5  import numpy as np
6  import numpy.random as rd
7
8  ## Données
9
10 L=0.665 #m = longueur raquette
11 w=4.27 #rad.s-1
12 X=np.array([2.4,5.6,10.9,16.8,22.7,31.6]) # distance à partir du bout de la raquette en cm
13 Ve=np.array([2.956,2.990,3.077,2.917,3.101,2.987]) #vitesses d'entrée
14 Vs=np.array([0.461,0.864,1.308,1.364,1.508,1.536]) #vitesses de sortie
15 e=Vs/Ve #ACOR
16
17 u_X=2*10**-3 #m = incertitude pointage
18 dt=1/30 #s = intervalle entre les frames
19 u_v=2**0.5*u_X/dt #m.s-1
20 u_w=u_v/L #s-1
21 u_L = 0.0005 #cm
22
23
24 ## Incertitudes composées sur e
25
26 nb = len(e)
27 u_e=np.zeros(nb)
28 for i in range(nb):
29     u_e[i]=e[i]*((u_v/Vs[i])**2+(u_v/Ve[i])**2)**0.5
30
31 ## Monte Carlo
32
33 N=10000
34 w_sim = w + u_w*rd.normal(0,1,N)
35
36 L_sim = L + u_L*rd.normal(0,1,N)
37
38 Vcalc_sim = np.zeros((nb,N))
39 V = np.zeros(nb)
40 u_V = np.zeros(nb)
41
42 e_sim = np.zeros((nb,N))
43 X_sim = np.zeros((nb,N))
44
45 for i in range(nb):
46     e_sim[i] = e[i] + u_e[i]*rd.normal(0,1,N)
47     X_sim[i] = X[i] + u_X*rd.normal(0,1,N)
48
49 for i in range(nb):
50     Vcalc_sim[i] = w_sim*(L_sim-X_sim[i]*10**-2)*(1+e_sim[i])
51     V[i], u_V[i] = np.mean(Vcalc_sim[i]), np.std(Vcalc_sim[i], ddof=1)
52
53 ## Tracé
54
55 plt.plot(X,V,linestyle='-.',label="Cas service")
56 plt.scatter(X,V,s=100,marker='+',c='b')
57 plt.errorbar(X,V,xerr=u_X*10**2,yerr=u_V,fmt='b+',ecolor='b',linewidth=1,capsize=3)
58 plt.ylabel('Vs')
59 plt.xlabel('x en cm')
60 plt.legend()
61 plt.title('Vitesse en fonction de la distance par rapport à la pointe de la raquette')
62 plt.grid()
63 plt.show()
64
```

Tracé de la vitesse de sortie  
théorique en fonction de la  
position avec prise en compte de  
l'ACOR (modèle 2) + Incertitudes  
par Monte Carlo

# Annexe 3 : Codes Python

```
1  ## Importation des bibliothèques
2
3  import numpy as np
4  from math import *
5  import matplotlib.pyplot as plt
6  import numpy.random as rd
7
8  ## Données
9
10 p_m=0.1 #g = précision balance
11
12 u_T = np.array([1/20,1/22,1/32,1/16,1/24,1/22,1/20,1/29,1/20,1/22,1/24]) #s = incertitude sur la période
13 u_L = 0.5 #cm = incertitude sur les centres de masse
14
15 u_x = 1*10**(-3) #m = incertitude sur le pointage
16 dt = 1/60 #s = intervalle de temps entre 2 frames
17 u_Vs = 2*0.5*u_x/(dt) #m.s-1 = incertitude sur la vitesse
18
19 L=np.array([18.3,20.4,22.5,24.6,26.8,28.9,31.1,33.2,35.3,37.4,38.5]) # distance centres de masse en cm
20 T=np.array([0.95,1.35/32,17/16,27/24,25/22,23/20,36/29,26/20,29/22,33/24]) # périodes des oscillations
21 g=9.81 #m.s-2 = accélération de la pesanteur
22 m=274.3 #g = masse de la poutre qui modélise la raquette
23
24 Vs1=np.array([2.076,2.325,2.703,2.831,2.911,2.977,2.902,2.821,2.832,2.732,2.661]) # mesures
25
26 ##Incertitudes avec Monte Carlo
27
28 N = 100000
29 nb = len(T)
30
31 m_sim = m + rd.uniform(-p_m,p_m,N)
32 T_sim = np.zeros((nb,N))
33 L_sim = np.zeros((nb,N))
34 Jcalc_sim = np.zeros((nb,N))
35 J = np.zeros(nb)
36 u_J = np.zeros(nb)
37
38 for i in range(nb):
39     T_sim[i] = T[i] + u_T[i]*rd.normal(0,1,N)
40     L_sim[i] = L[i] + u_L*rd.normal(0,1,N)
41
42 for i in range(nb):
43     Jcalc_sim[i] = (m_sim*10**(-3))*g*10**2*L_sim[i]*T_sim[i]**2/(4*pi**2)
44     J[i], u_J[i] = np.mean(Jcalc_sim[i]), np.std(Jcalc_sim[i], ddof=1)
45
46 ## Tracé
47
48 plt.figure(1)
49 plt.scatter(J,Vs1,s=100,marker='+',label='Cas service')
50 plt.plot(J,Vs1,linestyle='--')
51 plt.errorbar(J,Vs1,xerr=u_J,yerr=u_Vs,fmt='b+',ecolor='g',linewidth=1,capsize=2)
52 plt.legend()
53 plt.ylabel('Vs en m.s-1')
54 plt.xlabel('J en kg.cm2')
55 plt.title("Vitesse de sortie de la balle en fonction du moment d'inertie")
56 plt.grid()
57 plt.show()
```

Tracé des résultats de  
l'expérience du double  
pendule avec incertitudes  
par Monte Carlo

# Annexe 3 :

## Codes Python

```
1  ## Importation des bibliothèques
2
3  from math import *
4  import matplotlib.pyplot as plt
5  import numpy as np
6  import numpy.random as rd
7
8  ## Données
9
10 l=68.5*10**-2 #m
11 u_l=0.001 #m incertitude
12 m=1.15*10**-3 #kg
13 p_m=0.1*10**-3 #kg = précision balance
14 u_f=10 #Hz
15 g=9.81 #m.s-2 = accélération de la pesanteur
16 L = 23.3*10**-2 #longueur cordes du cadre en m
17 f=np.array([137.5,192.5,257.5,307.5,397.5,497.5,602.5]) #fréquence en Hz
18 Vs=np.array([2.628,2.825,2.920,2.930,3.006,3.052,3.095]) #vitesse de sortie en m.s-1
19
20 u_x = 1*10**(-3) #m = incertitude sur le pointage
21 dt = 1/30 #s = intervalle de temps entre 2 frames
22 u_Vs = 2*0.5*u_x/(dt) #m.s-1 = incertitude sur la vitesse
23
24 ## Incertitudes avec Monte Carlo
25
26 N = 100000
27 nb = len(f)
28
29 m_sim = m + rd.uniform(-p_m,p_m,N)
30 l_sim = l + u_l*rd.normal(0,1,N)
31 mu_calc_sim = m_sim/l_sim #kg.m-1
32
33 m_lin,u_mu = np.mean(mu_calc_sim), np.std(mu_calc_sim)
34
35 L_sim = L + u_l*rd.normal(0,1,N)
36 f_sim = np.zeros((nb,N))
37 Tcalc_sim = np.zeros((nb,N))
38 T = np.zeros(nb)
39 u_T = np.zeros(nb)
40
41 for i in range(nb):
42     f_sim[i] = f[i] + u_f*rd.normal(0,1,N)
43
44 for i in range(nb):
45     Tcalc_sim[i] = 4*((f_sim[i]*L_sim)**2)*m_lin/g
46     T[i], u_T[i] = np.mean(Tcalc_sim[i]), np.std(Tcalc_sim[i], ddof=1)
47
48 ## Tracé
49
50 plt.scatter(T,Vs,s=100,marker='+')
51 plt.plot(T,Vs,linestyle='--')
52 plt.ylabel('Vs en m.s-1')
53 plt.xlabel('T en kg')
54 plt.errorbar(T,Vs,xerr=u_T,yerr=u_Vs,fmt='b+',ecolor='g',linewidth=1,capsize=2)
55 print("masse linéique =", format(m_lin*10**3, ".4f"), "g.m-1")
56 print('u_mu =', format(u_mu*10**3, ".4f"), 'g.m-1')
57 plt.title('Vitesse de sortie de la balle en fonction de la tension des cordes')
58 plt.grid()
59 plt.show()
```

### Tracé des résultats de l'expérience du cadre avec incertitudes par Monte Carlo