

Le skimboard : surfer sur la science

Ehouarn GUITTON
47909



Le skimboard



https://www.decathlon.fr/p/skimboard-100-bois-adulte/_/R-p-313077

La pratique du skimboard



Problématique

- Quelles sont les caractéristiques du skimboard qui font varier l'intensité des frottements ?
- Quels enseignements peut-on en tirer pour améliorer la pratique de ce sport ?

Objectifs

- 1) Modéliser la glisse : quels frottements ?
- 2) Rechercher des facteurs qui réduisent les frottements : étudier l'hydrophobie.
- 3) Optimiser la glisse grâce à différentes transformations du skimboard.

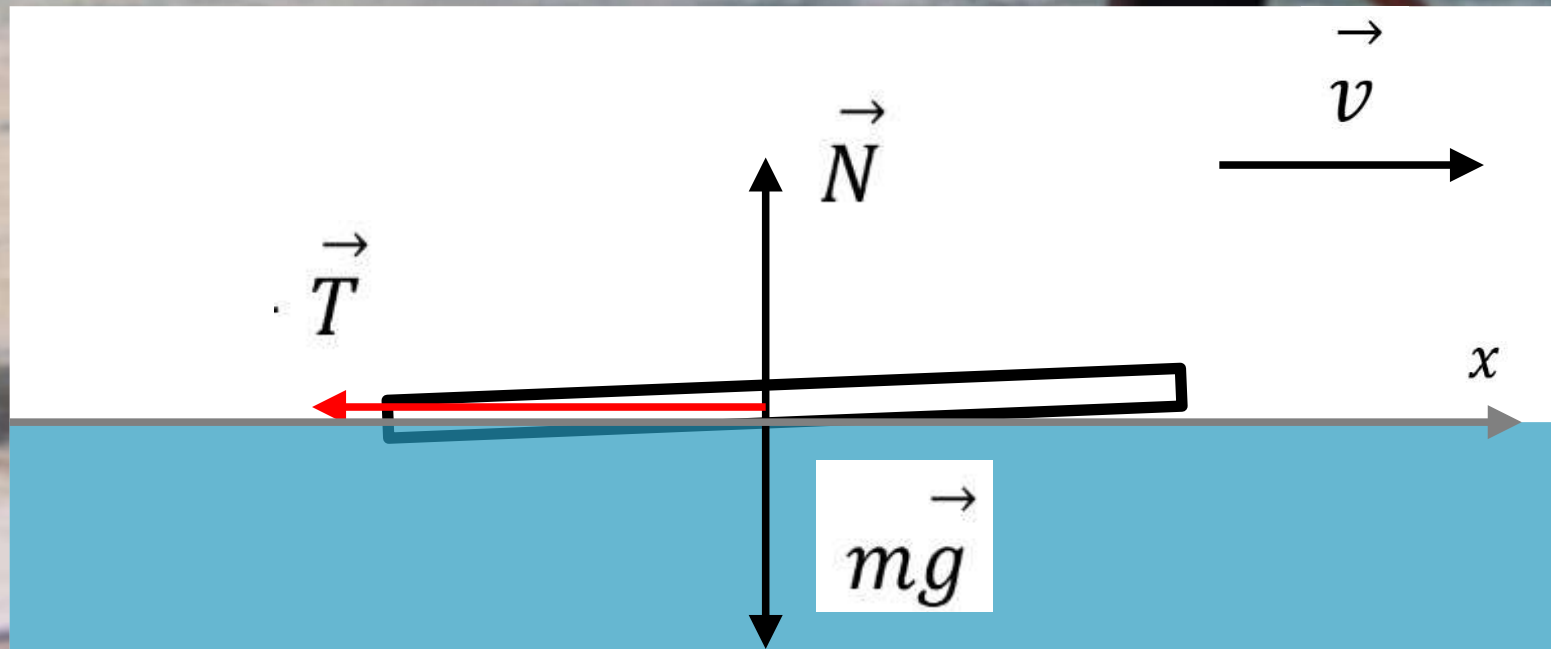
1) Modélisation de la glisse



Bilan des forces

2^{ème} loi de Newton :

$$m \frac{d\vec{v}}{dt} = m\vec{g} + \vec{N} + \vec{T}$$



Détermination des frottements

Frottements en v

$$\vec{T} = -k\vec{v}$$

Frottements en v^2

$$\vec{T} = -kv^2\vec{u}_x$$

$$R_e = \frac{\mu v L}{\eta} = 10^5 > 10^3$$

Hypothèse :

$$m \frac{d\vec{v}}{dt} = m\vec{g} + \vec{N} - kv^2\vec{u}_x$$

Dispositif expérimental

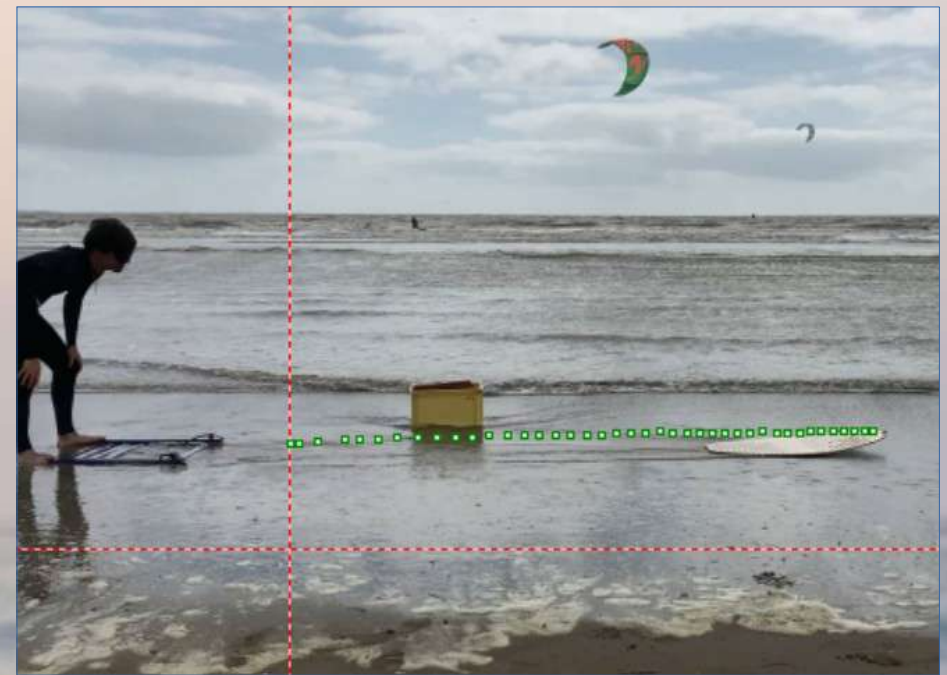


Lanceur fabriqué

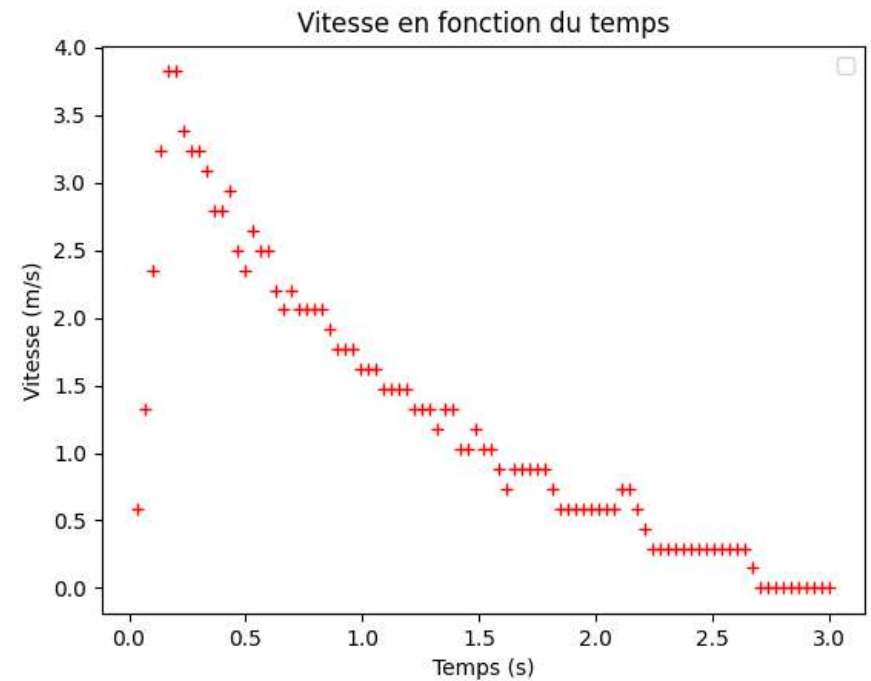
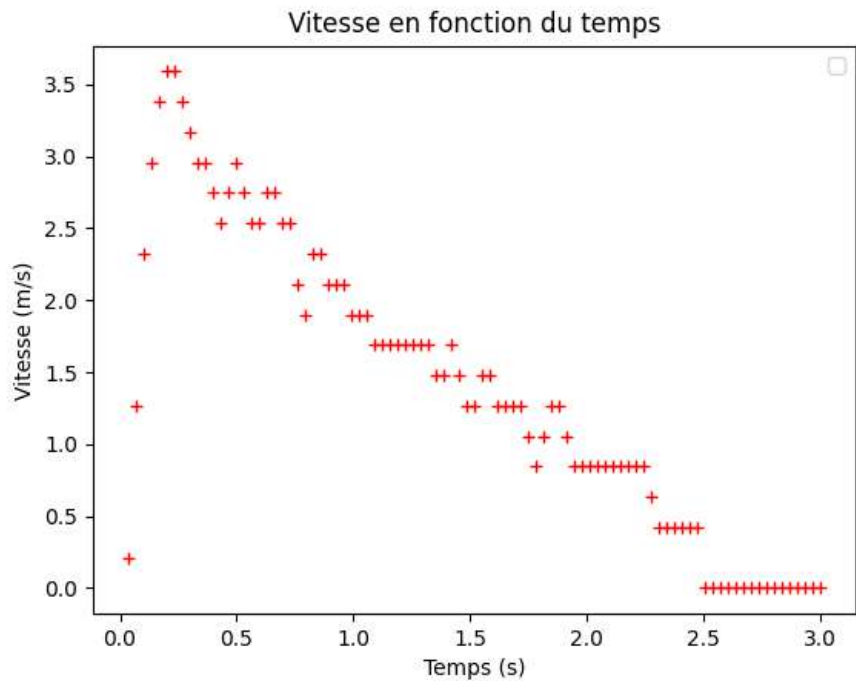


Conditions de l'expérience

Captage vidéo de nos expériences :

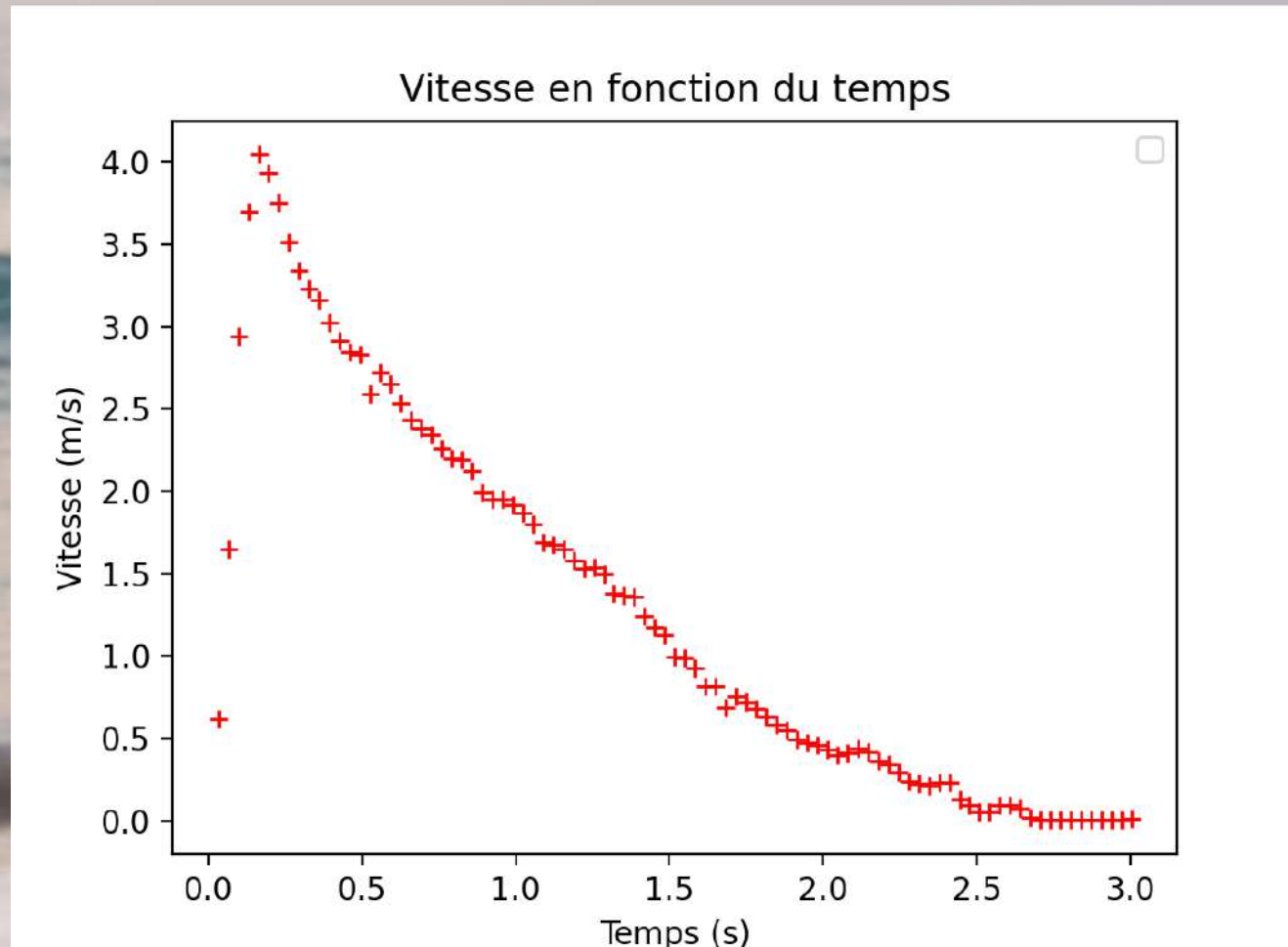


Obtention de courbes de la vitesse en fonction du temps :



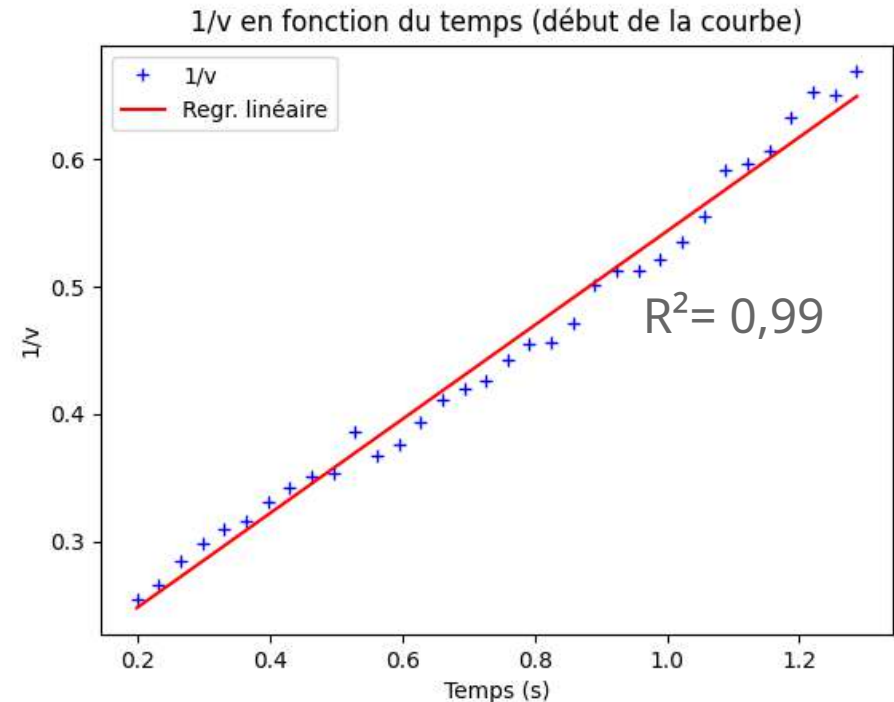
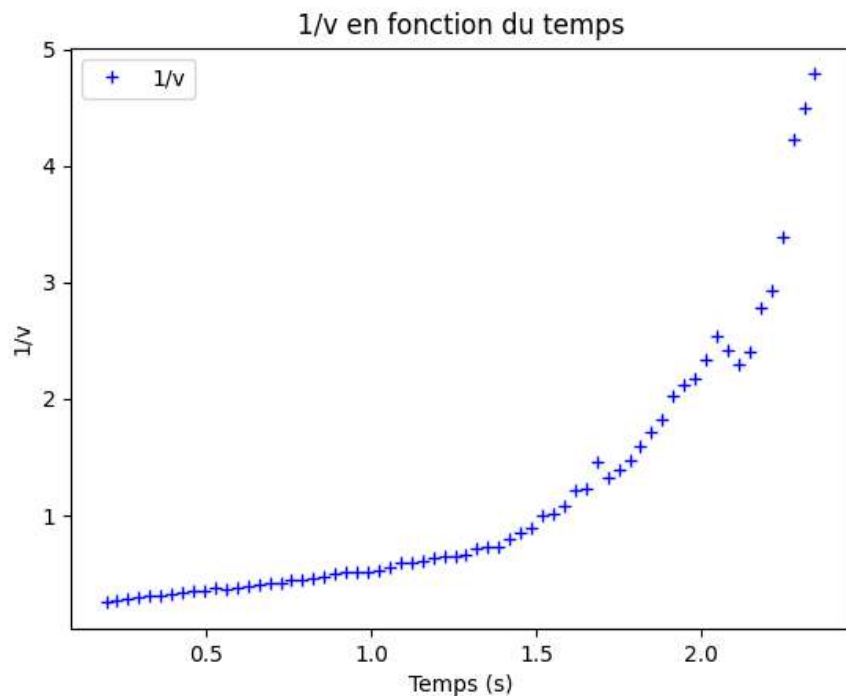
Mesures expérimentales

Moyenne sur 10 lancers :



Test de l'hypothèse

$$m \frac{dv}{dt} = -kv^2 \quad \Leftrightarrow \quad -\frac{dv}{v^2} = \frac{k}{m} dt \quad \Leftrightarrow \quad \frac{1}{v} = \frac{k}{m} t + \frac{1}{v_0}$$



Résultat

- Les frottements en v^2 seuls ne permettent pas de modéliser la glisse de façon satisfaisante.

Nécessité d'un nouveau modèle.

Ajout d'une force de frottement solide :

$$m \frac{d\vec{v}}{dt} = m\vec{g} + \vec{N} - kv^2\vec{u}_x - fmg\vec{u}_x$$

Nouveau modèle

$$\frac{dv}{dt} = -\frac{k}{m}v^2 - fg \quad \Leftrightarrow \quad \frac{dv}{dt} = -a - bv^2$$

En posant :

$$a = fg \quad b = \frac{k}{m}$$

Sachant que :

$$\frac{dv}{dt} = \frac{dx}{dt} \frac{dv}{dx} = v \frac{dv}{dx}$$

On en déduit :

$$x = \frac{1}{2b} \ln \left(\frac{a+bv_0^2}{a+bv^2} \right)$$

Module python :

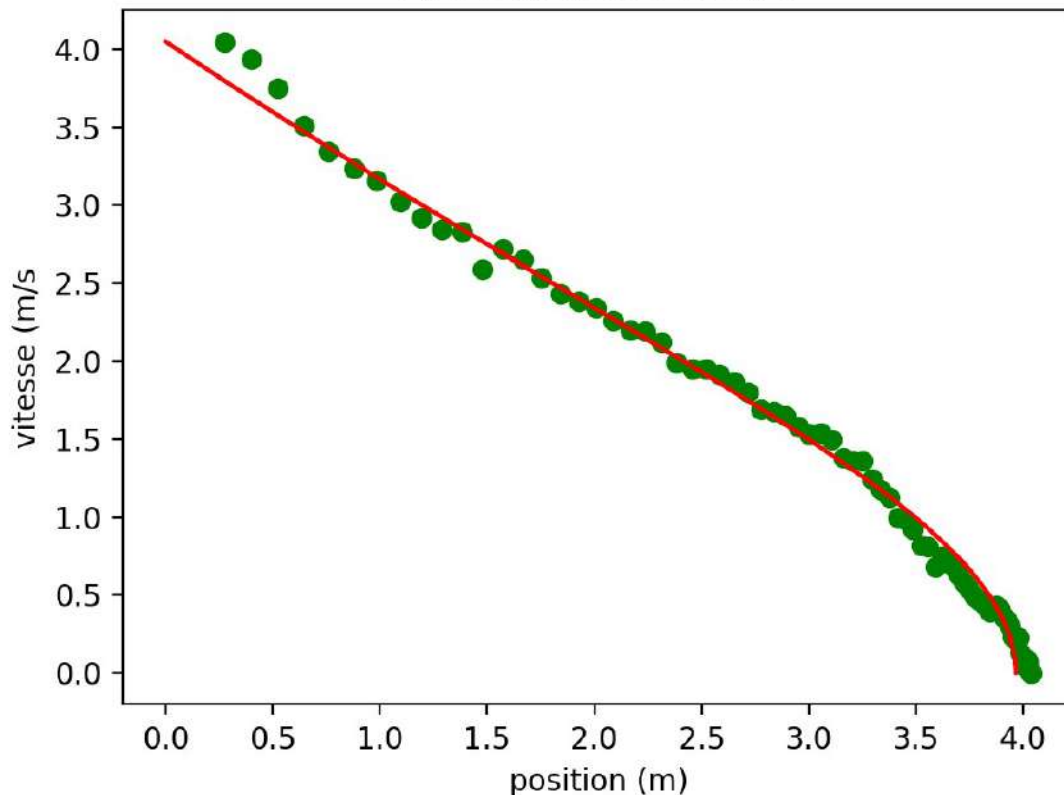
SciPy.Optimise

Fonction :

Curv_fit

$$x = \frac{1}{2b} \ln \left(\frac{a + bv_0^2}{a + bv^2} \right)$$

Ajustement de courbe



En vert : points expérimentaux

En rouge : courbe modélisée par *Curv_fit*

Nouveau modèle

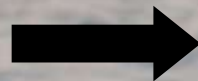
Rappel :

$$\frac{dv}{dt} = -\frac{k}{m}v^2 - fg$$

$$a = fg \quad b = \frac{k}{m}$$

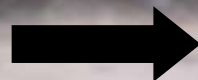
Valeurs obtenues :

$$a = 0,98 \text{ N} \cdot \text{m}^{-1}$$



$$f = 0,1$$

$$b = 0,17 \text{ m}^{-1}$$



$$k = 0,37 \text{ kg} \cdot \text{m}^{-1}$$

Conclusion 1

- On peut modéliser les frottements sur fine couche d'eau par une force de frottement fluide en v^2 et une force de frottement solide.

$$m \frac{d\vec{v}}{dt} = m\vec{g} + \vec{N} - kv^2\vec{u}_x - fmg\vec{u}_x$$

2) L'hydrophobie : réduire le frottement



Choix d'un matériau

Wax industrielle



« Wax » artisanale



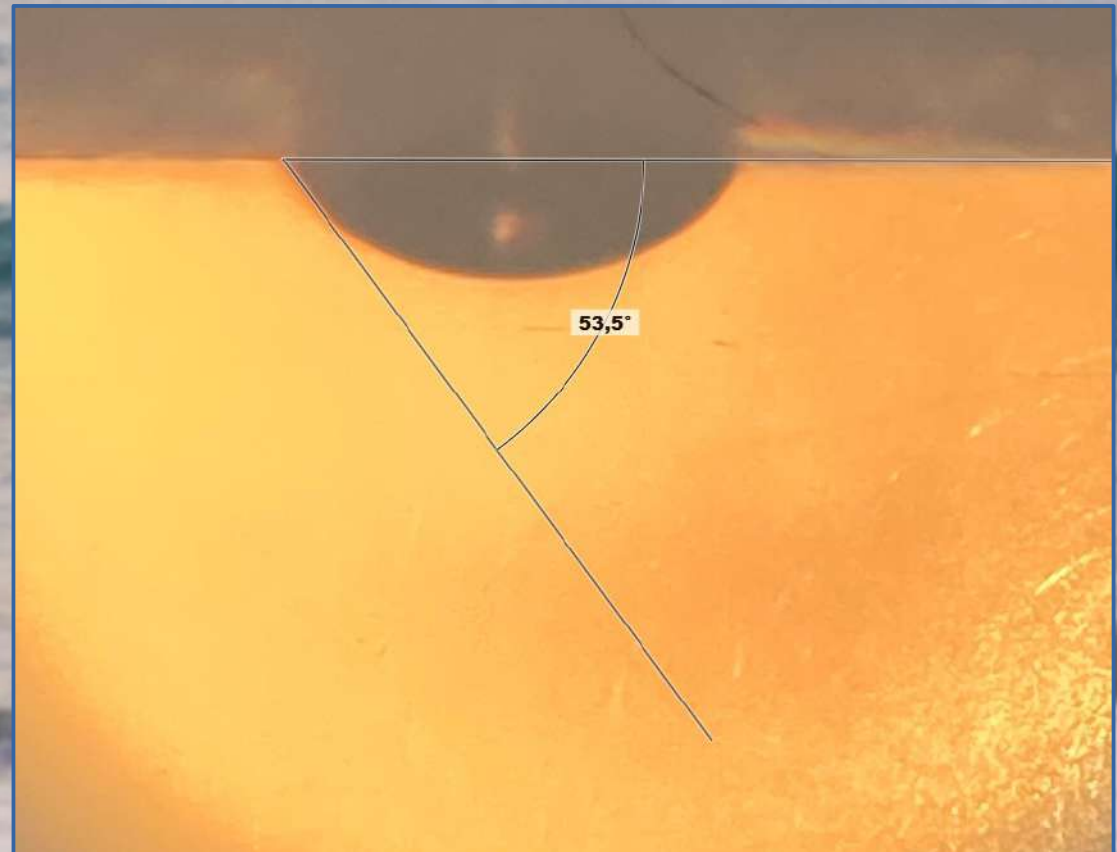
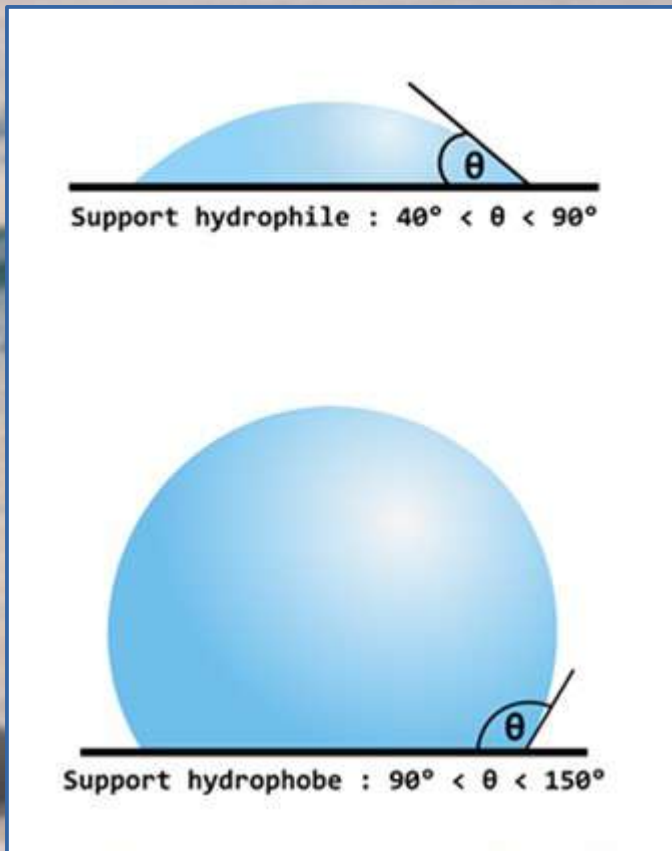
Wax indus.



Wax
artisanale

Angle d'incidence d'une goutte

LOGICIEL : *Mesurim2*



Montage

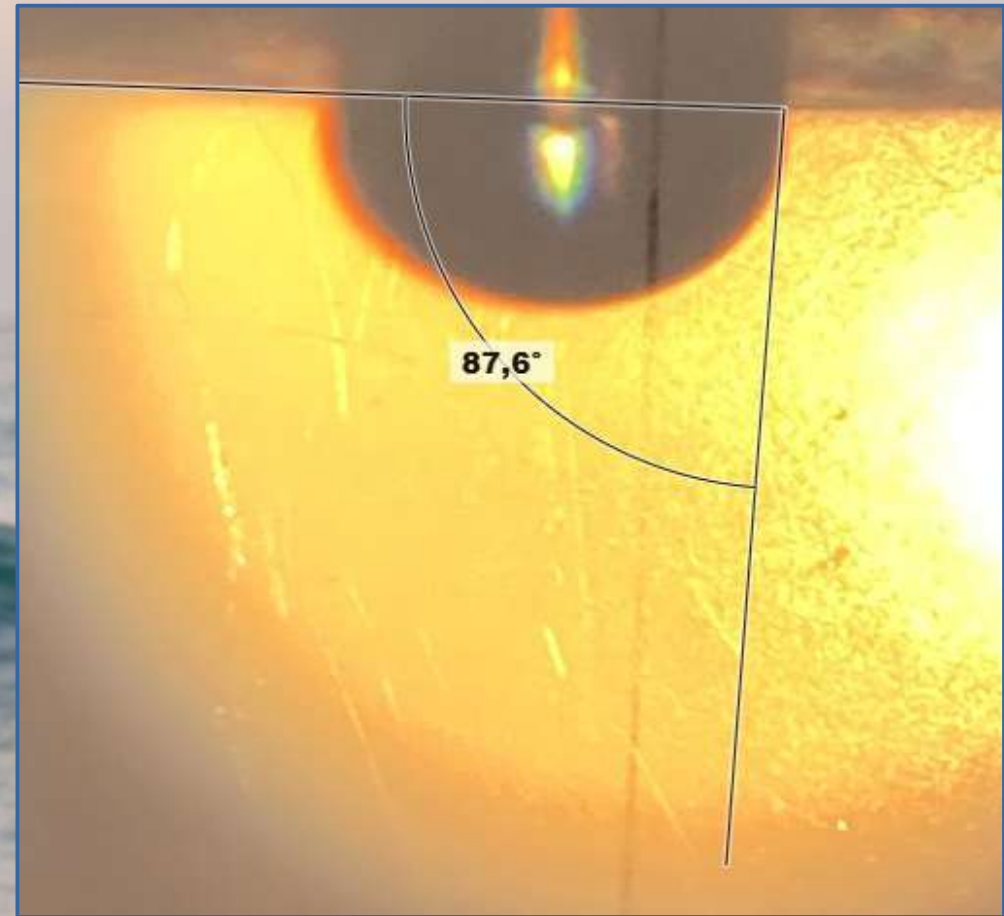


Sans wax



Moyenne : **$52^{\circ} \pm 1^{\circ}$**

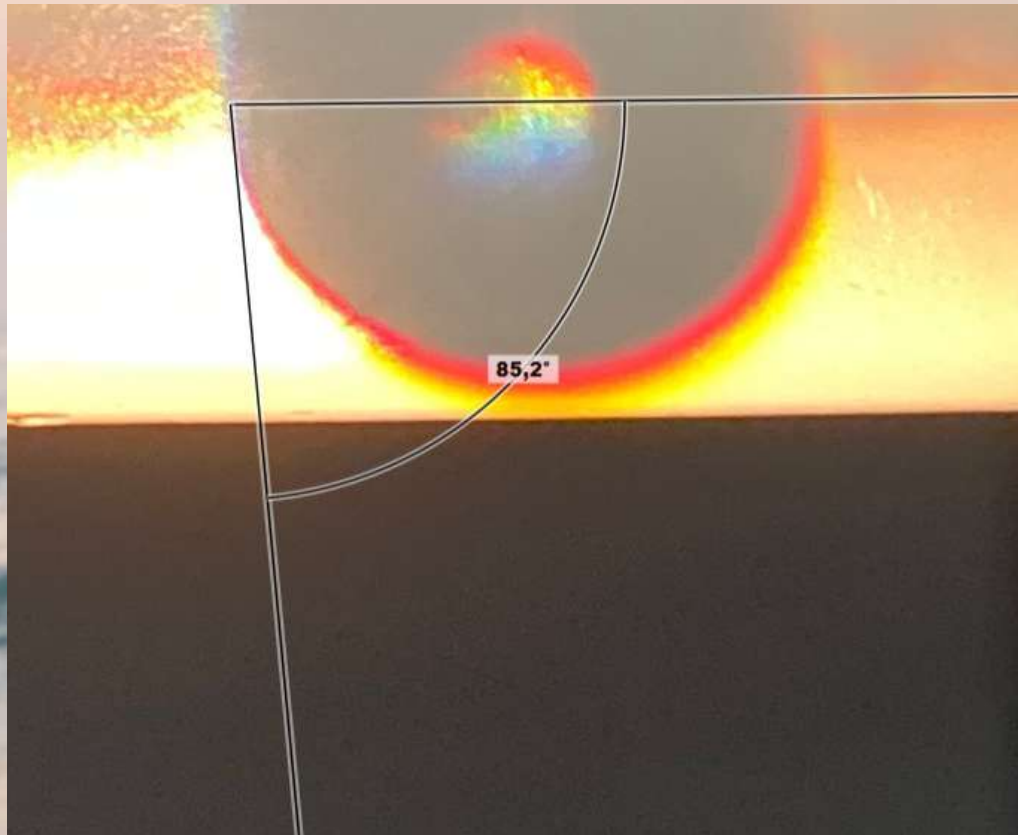
Avec wax indus.



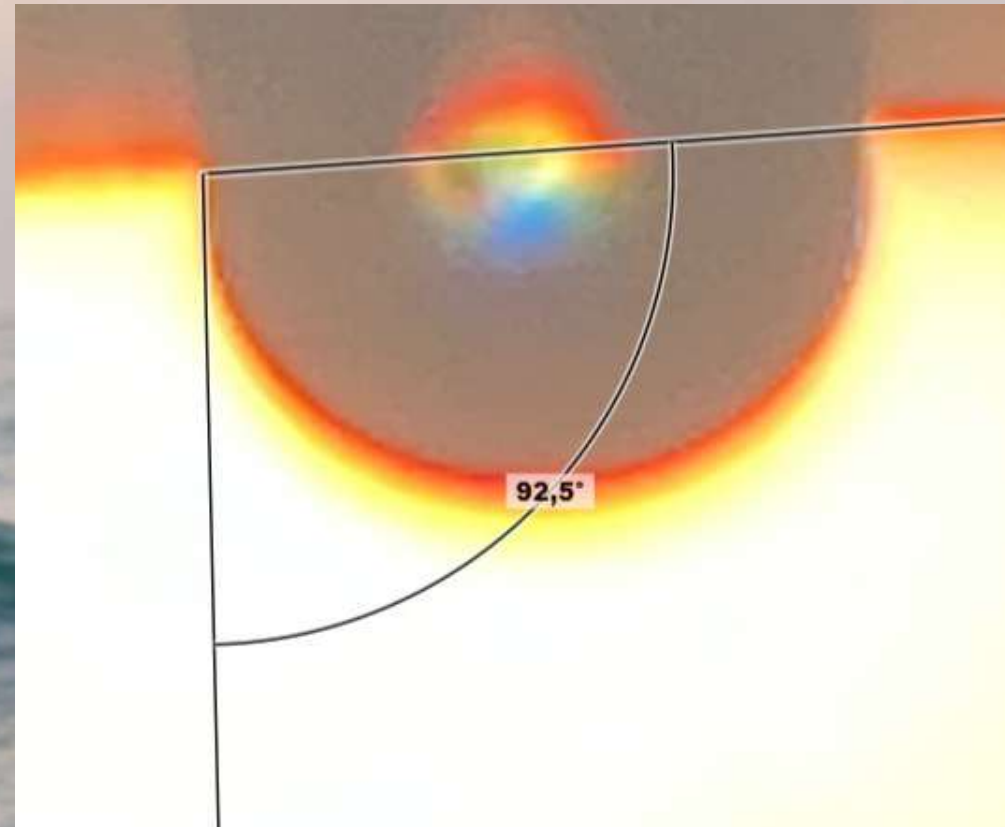
Moyenne : **$86^{\circ} \pm 1^{\circ}$**

Différence d'angle $\sim 34^{\circ}$

Wax indus.



Wax artisanale



Moyenne : $85^\circ \pm 1^\circ$

Moyenne : $93^\circ \pm 1^\circ$

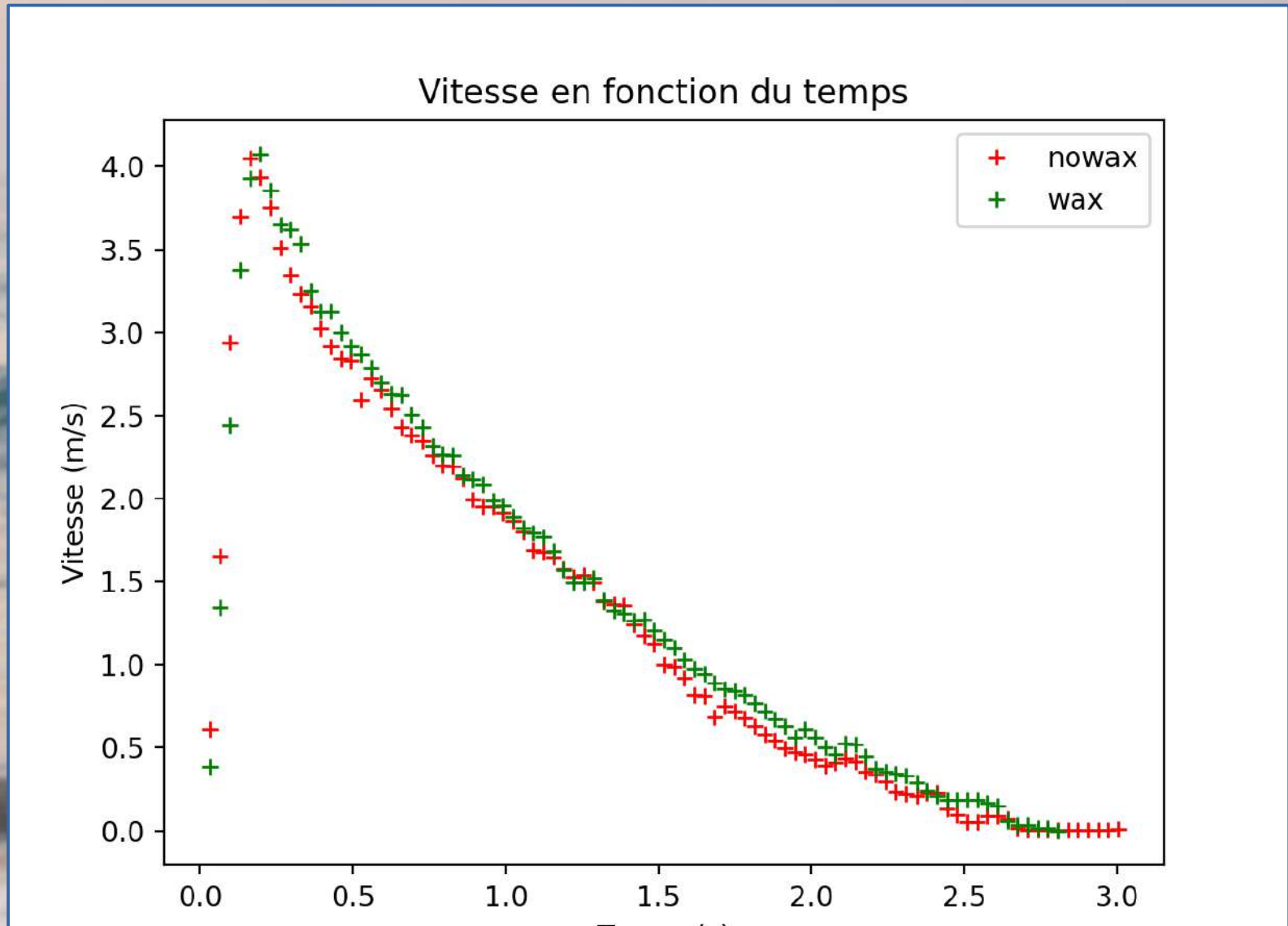
Conclusion 2

- La wax industrielle et la wax artisanale rendent la surface du skimboard hydrophobe
- La wax artisanale a un effet hydrophobe plus important que la wax industrielle

3) L'impact de la wax sur la glisse



L'impact de la wax indus.



L'impact de la wax indus.

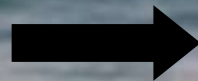
- Valeurs des constantes

$$a = 0,88 \text{ N} \cdot \text{m}^{-1}$$

$$b = 0,19 \text{ m}^{-1}$$

Wax indus.

$$k = 0,41 \text{ kg} \cdot \text{m}^{-1}$$



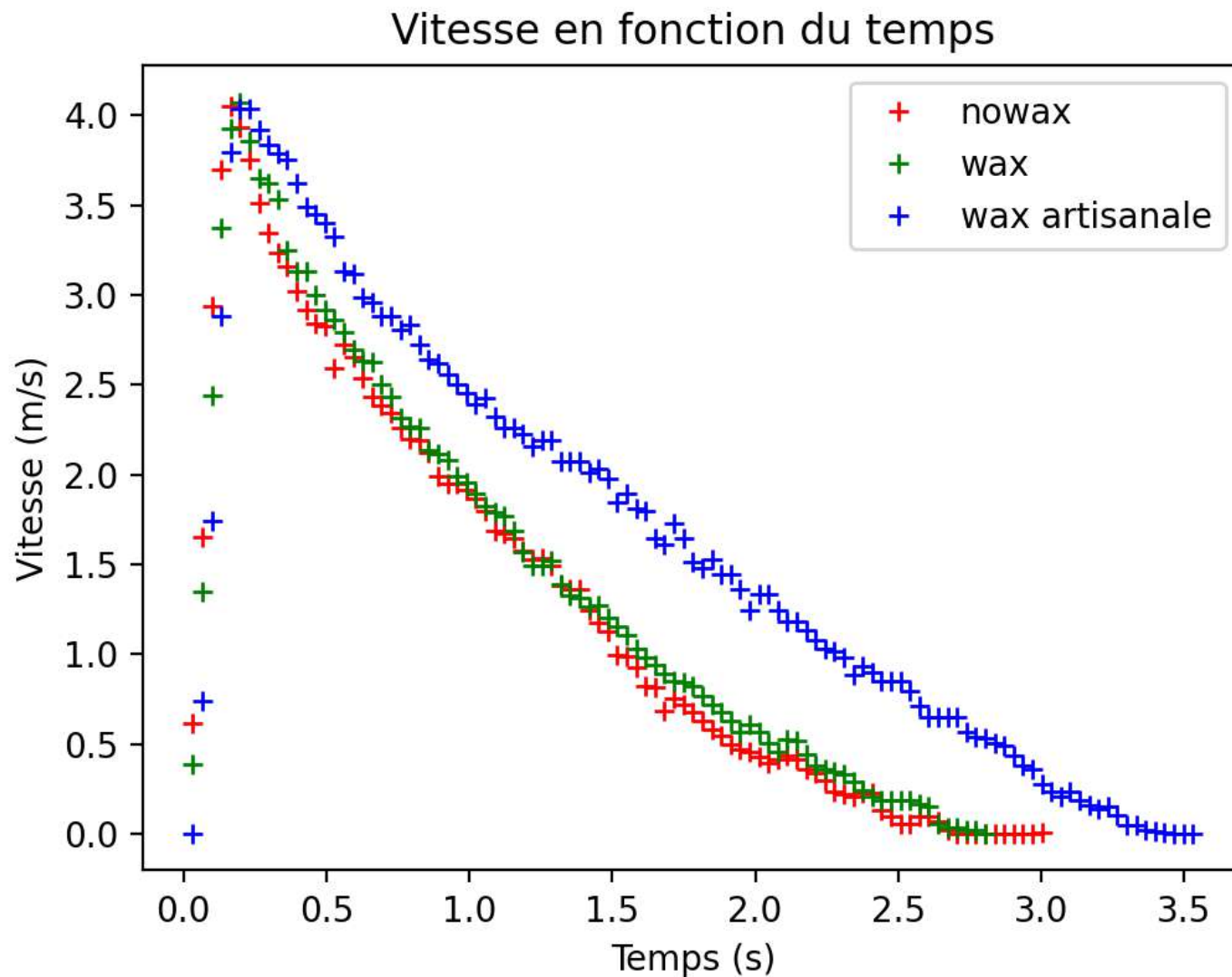
$$k = 0,41 \text{ kg} \cdot \text{m}^{-1}$$

Pas de wax

$$k = 0,37 \text{ kg} \cdot \text{m}^{-1}$$

~

L'impact de la wax artisanale

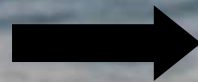


L'impact de la wax artisanale

- Valeurs des constantes

$$a = 0,95 \text{ N.m}^{-1}$$

$$b = 0,07 \text{ m}^{-1}$$



$$k = 0,15 \text{ kg.m}^{-1}$$

Wax artisanale

Pas de wax

$$k = 0,15 \text{ kg.m}^{-1}$$

<

$$k = 0,37 \text{ kg.m}^{-1}$$

Résultats

Pas de wax

$$k = 0,37 \text{ kg.m}^{-1}$$

Wax indus.

$$k = 0,41 \text{ kg.m}^{-1}$$

Wax artisanale

$$k = 0,15 \text{ kg.m}^{-1}$$

Conclusion 3

Ces résultats suggèrent que la **wax artisanale** réduit les frottements mais que l'effet de la **wax industrielle** est négligeable.

Cependant :

Les résultats sont-ils susceptibles d'être influencés par une éventuelle hétérogénéité des différentes trajectoires mesurées ?

⇒ analyse complémentaire

Modélisation des trajectoires individuelles :
3 courbes sans wax présentent des valeurs de k
nettement plus élevées que les autres.

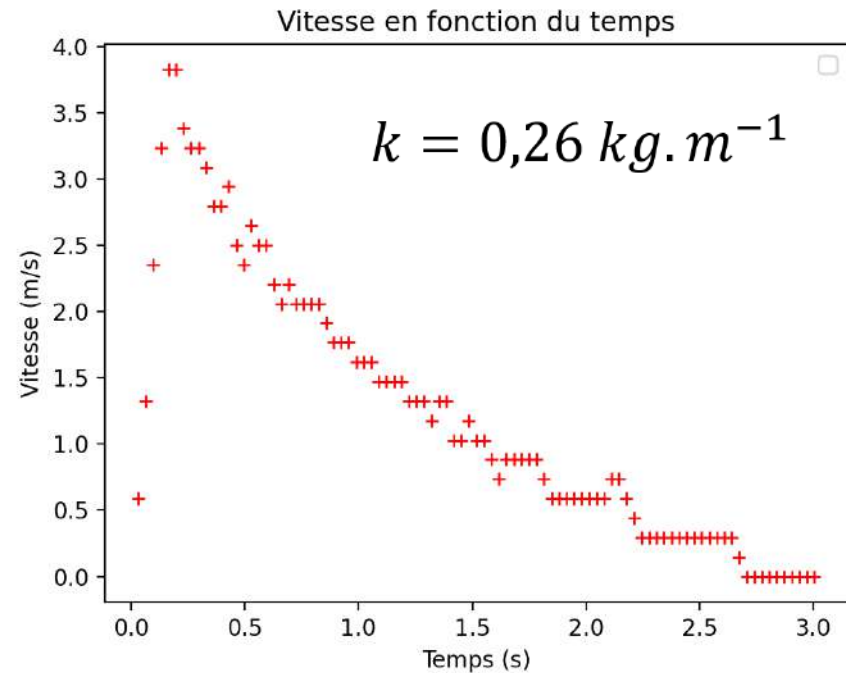
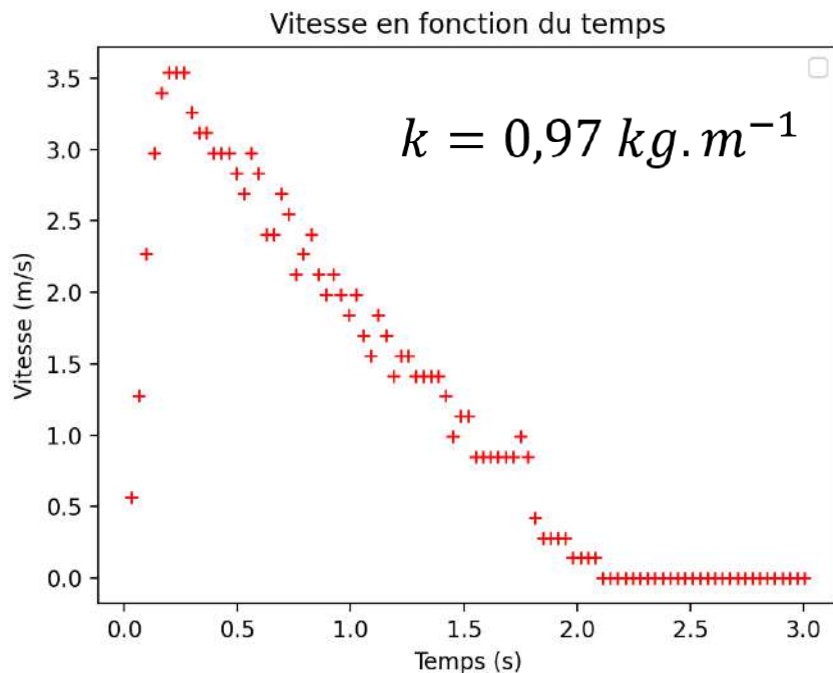
$$k = 1,07 \text{ kg.m}^{-1}$$

$$k = 1,03 \text{ kg.m}^{-1}$$

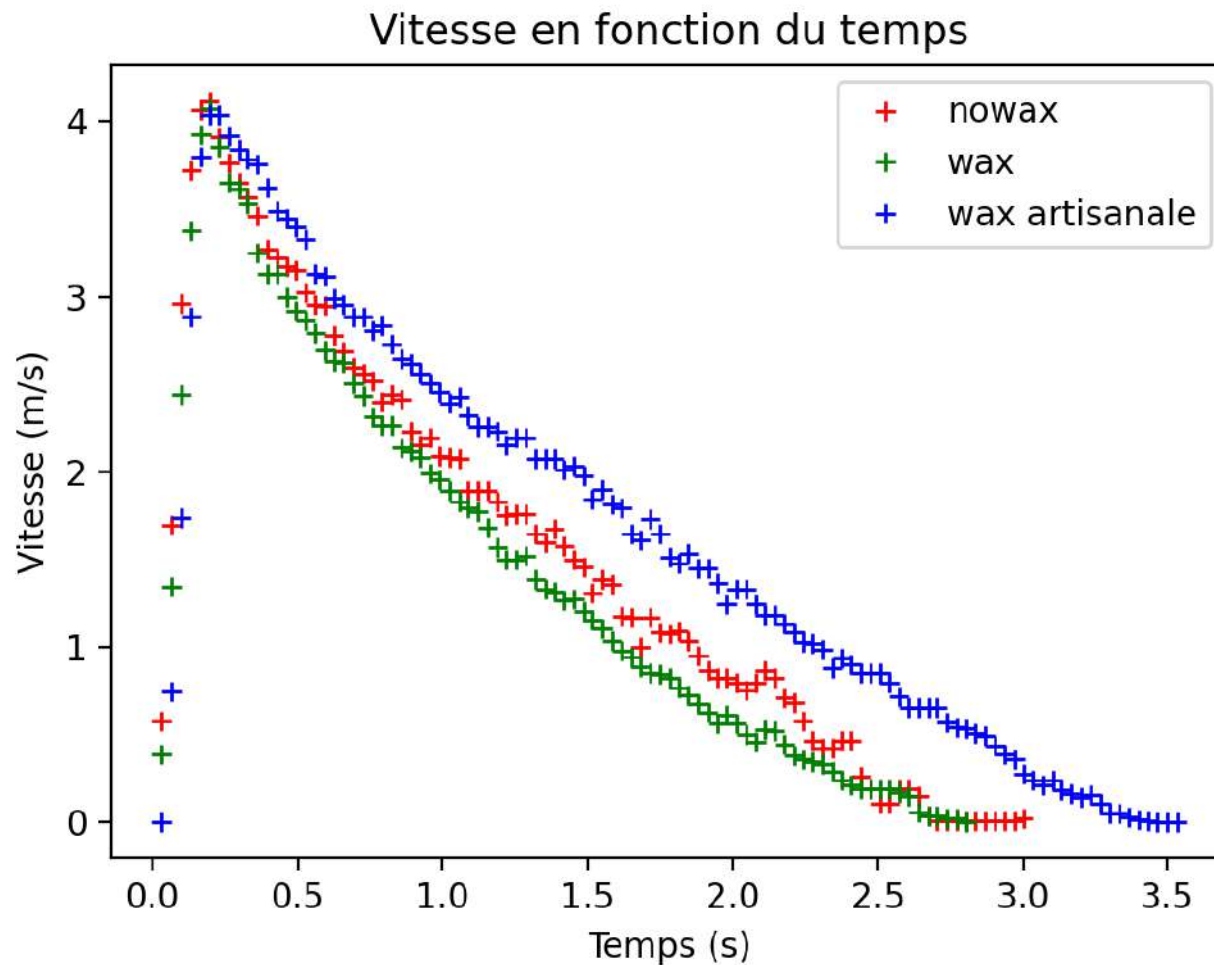
$$k = 0,97 \text{ kg.m}^{-1}$$

Autres courbes :

$$0,13 \text{ kg.m}^{-1} < k < 0,61 \text{ kg.m}^{-1}$$



Étude sans ces 3 courbes :



Nouvelle valeur :

$$k = 0,24 \text{ kg.m}^{-1}$$

Résultats

Pas de wax

$$k = 0,24 \text{ kg.m}^{-1}$$

Wax indus.

$$k = 0,41 \text{ kg.m}^{-1}$$

Wax artisanale

$$k = 0,15 \text{ kg.m}^{-1}$$

Application numérique :

$$x_f = \frac{1}{2b} \ln \left(\frac{a + bv_0^2}{a + bv_f^2} \right)$$

Avec : $v_0 = 5 \text{ m.s}^{-1}$

Sans wax

$$x_f = 6,07 \text{ m}$$

Wax artisanale

$$x_f = 7,46 \text{ m}$$

Conclusion



- Modèle de glisse du skimboard : frottement fluide en v^2 et frottement solide.
- Effet hydrophobe de la wax artisanale > effet hydrophobe wax industrielle.
- Wax artisanale semble réduire les frottements et améliorer la glisse.
- Cependant : variabilité des trajectoires individuelles à explorer + augmenter les répétitions.

Travail du binôme

- Modélisation différente de la glisse : les forces de pressions exercées par l'eau sur la planche.
- Étude de la glisse avec un skimboard en matière epoxy.
- Quantification et réduction de la dégradation du skimboard liée à la tension superficielle.

Annexes



Annexe

Test des frottements en v :

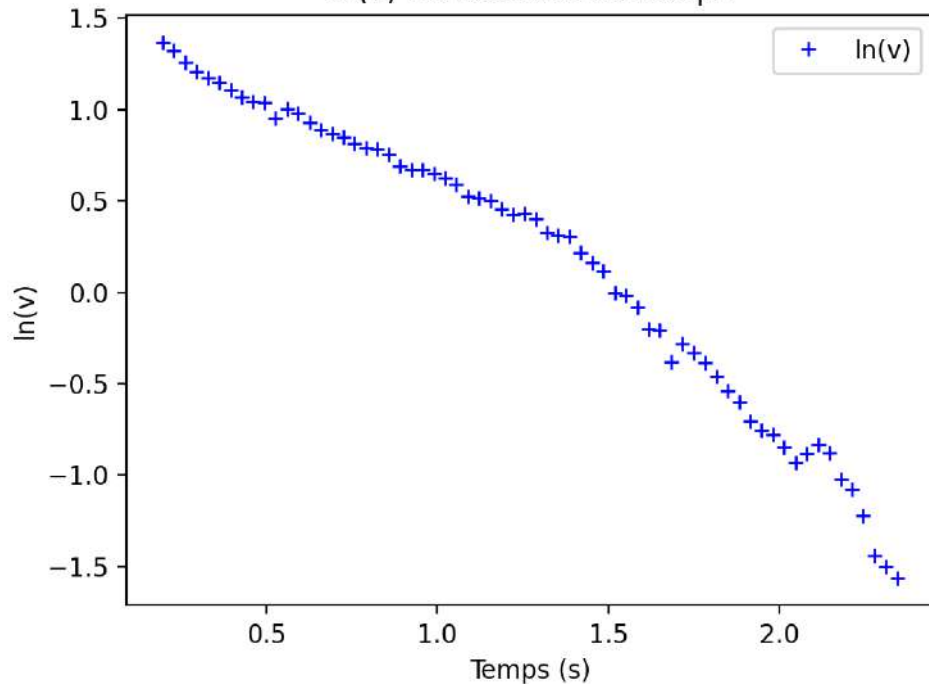
$$\vec{T} = -k\vec{v}$$

$$m \frac{d\vec{v}}{dt} = -k\vec{v} \Leftrightarrow \frac{dv}{v} = -\frac{k}{m} dt \Leftrightarrow \ln(v) = -\frac{k}{m} t + \ln(v_0)$$

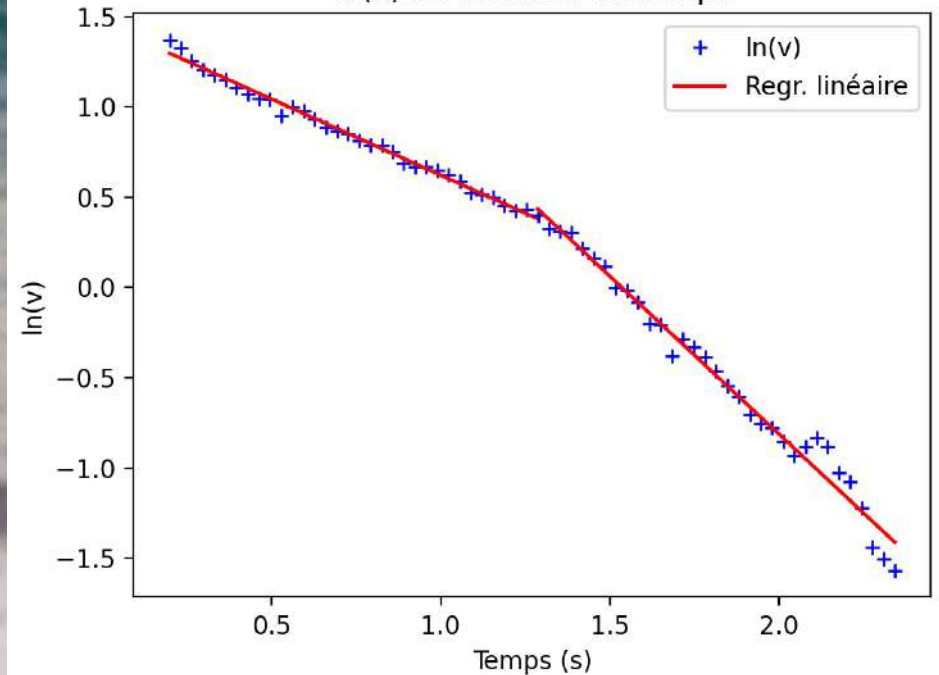
Annexe

$$\ln(v) = -\frac{k}{m}t + \ln(v_0)$$

ln(v) en fonction du temps



ln(v) en fonction du temps



Annexe

Sachant que :

$$\frac{dv}{dt} = \frac{dx}{dt} \frac{dv}{dx} = v \frac{dv}{dx}$$

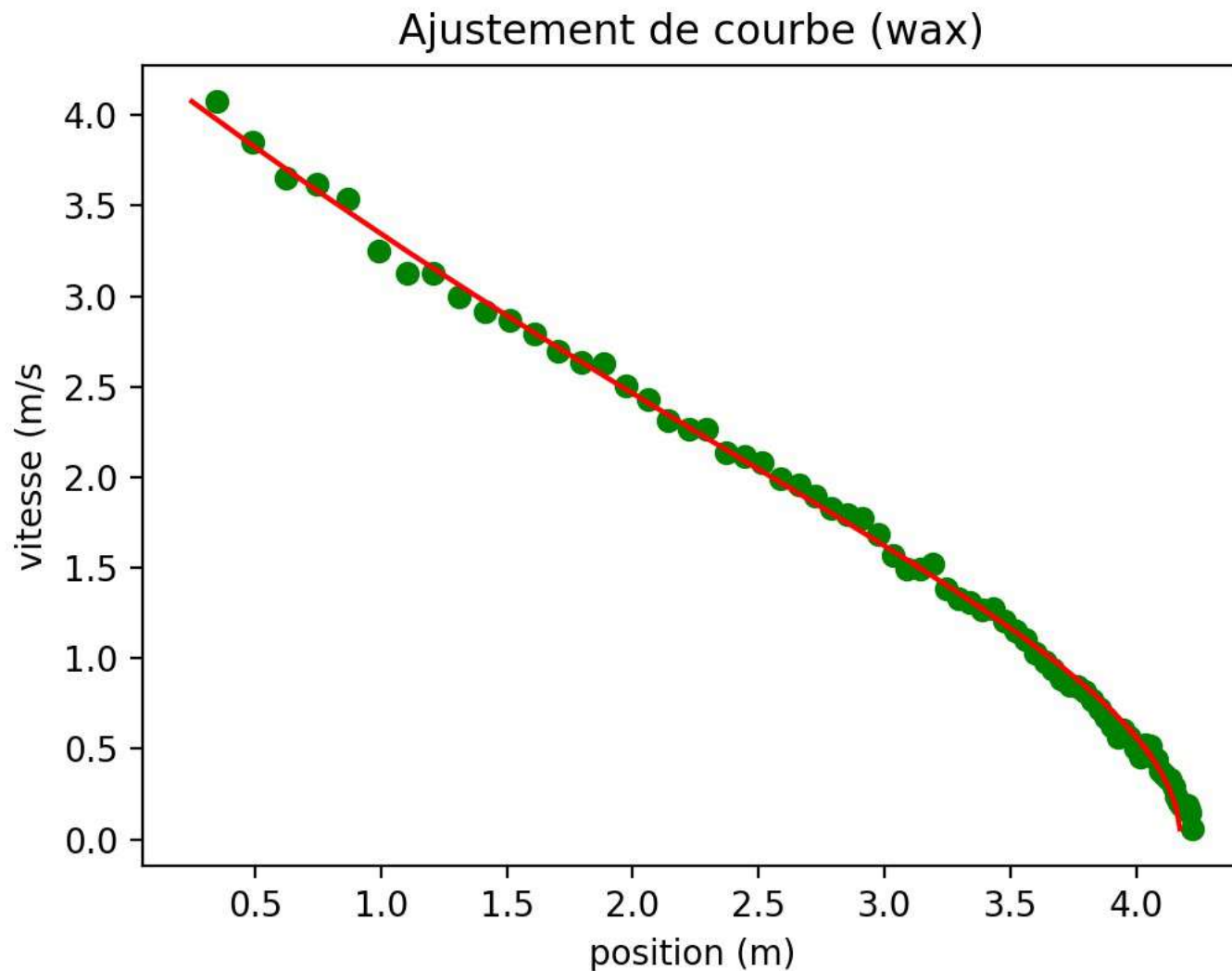
$$\frac{dv}{dt} = -\frac{k}{m}v^2 - fg \quad \Leftrightarrow \quad \frac{dv}{dt} = -a - bv^2 \quad \Leftrightarrow \quad v \frac{dv}{dx} = -a - bv^2$$

$$v \frac{dv}{dx} = -a - bv^2 \quad \Leftrightarrow \quad -dx = \frac{v dv}{(a + bv^2)} \quad \Leftrightarrow \quad [x]_0^x = \left[\frac{\ln(a + bv^2)}{2b} \right]_{v_0}^v$$

$$\Leftrightarrow \quad x = \frac{1}{2b} \ln \left(\frac{a + bv_0^2}{a + bv^2} \right)$$

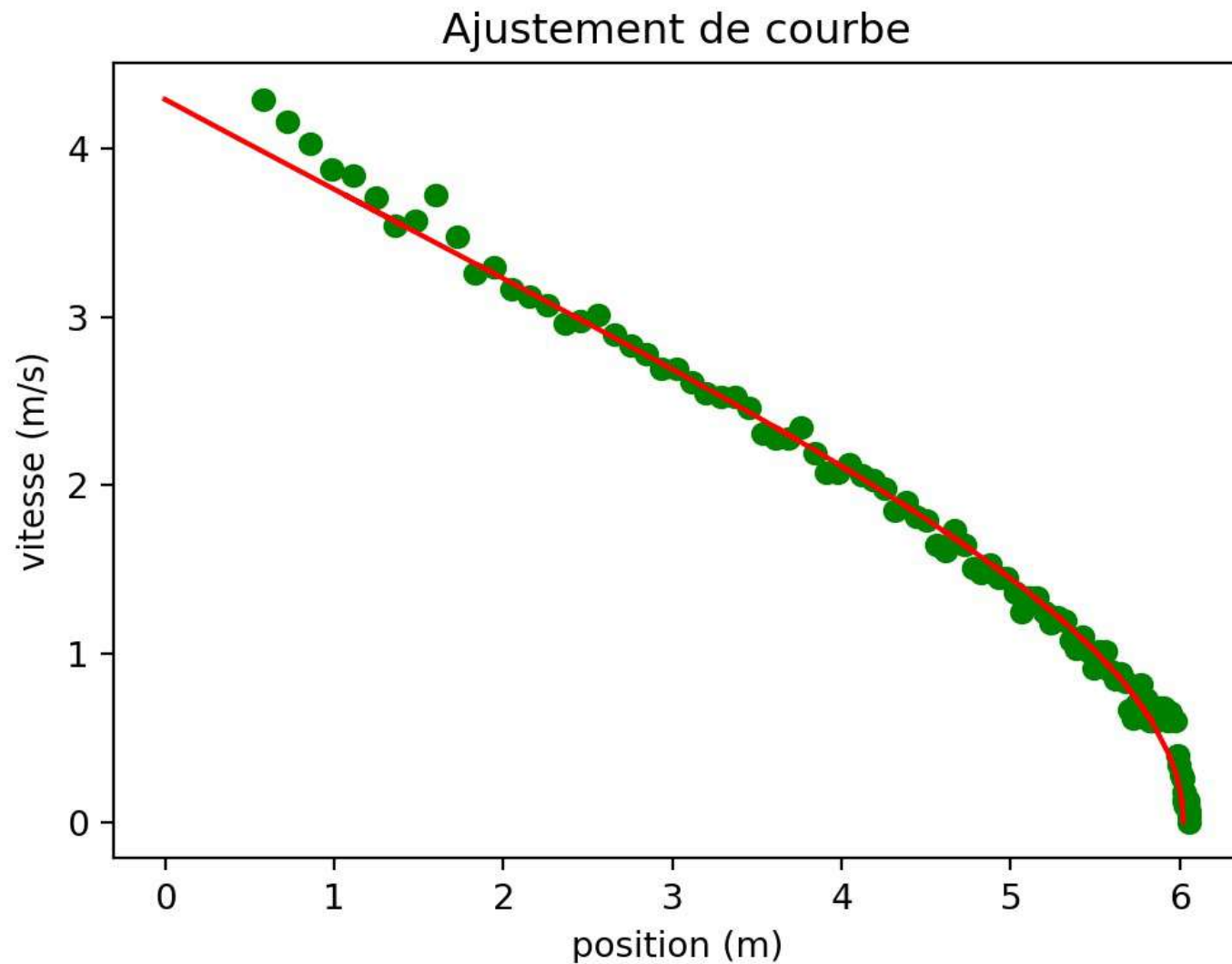
Annexe

Wax indus. :



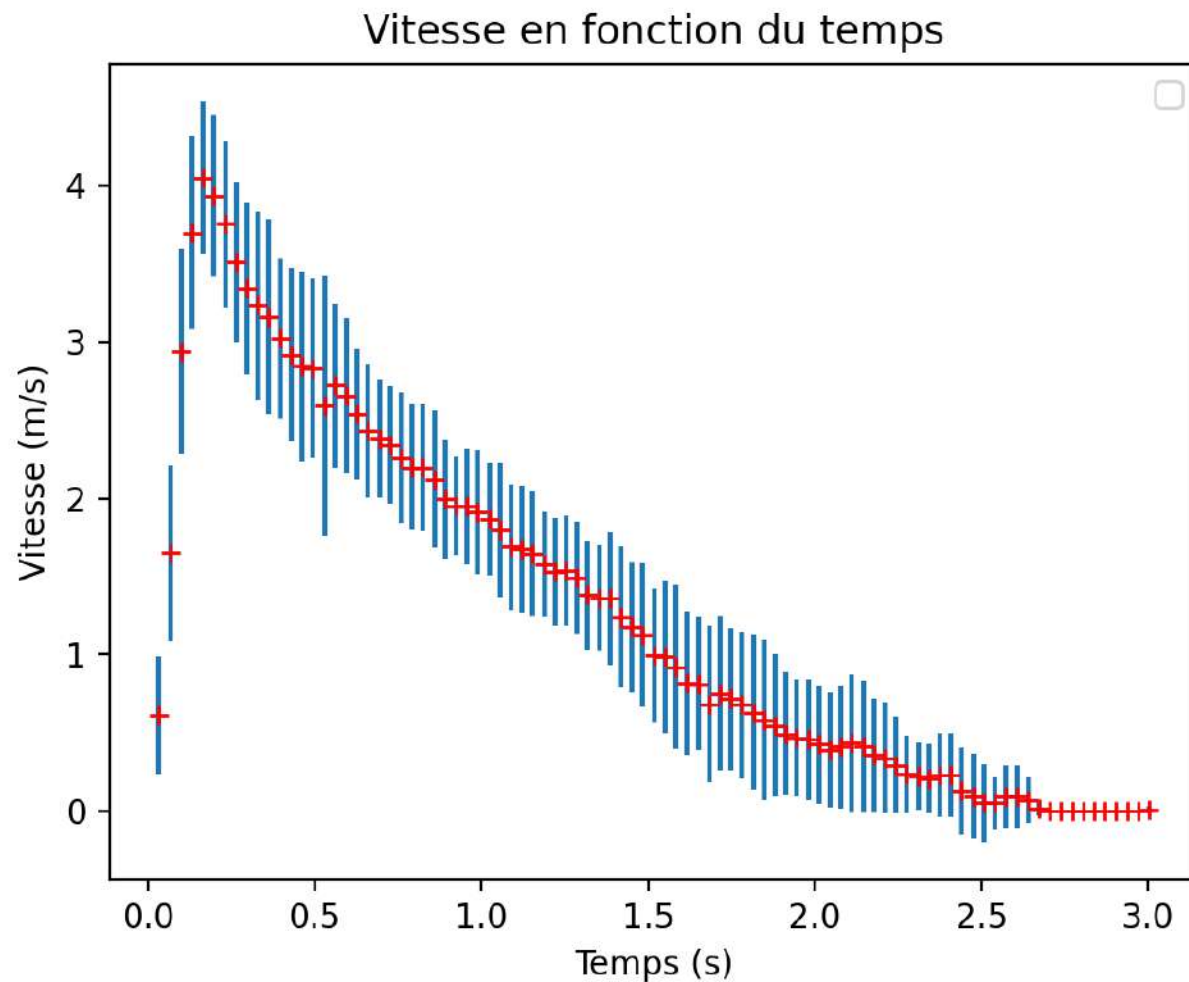
Annexe

Wax artisanale :



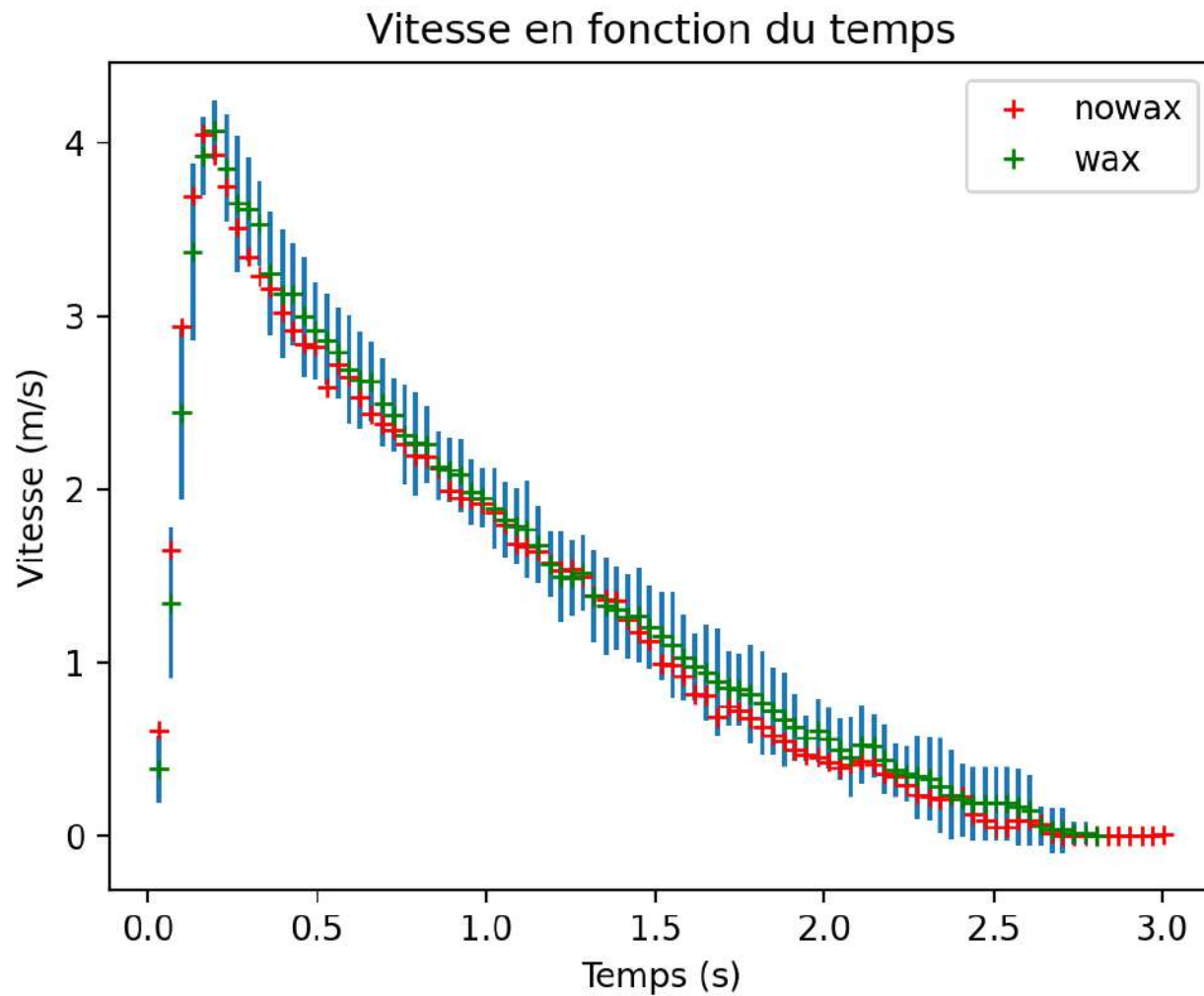
Annexe

Écart-type sans wax :



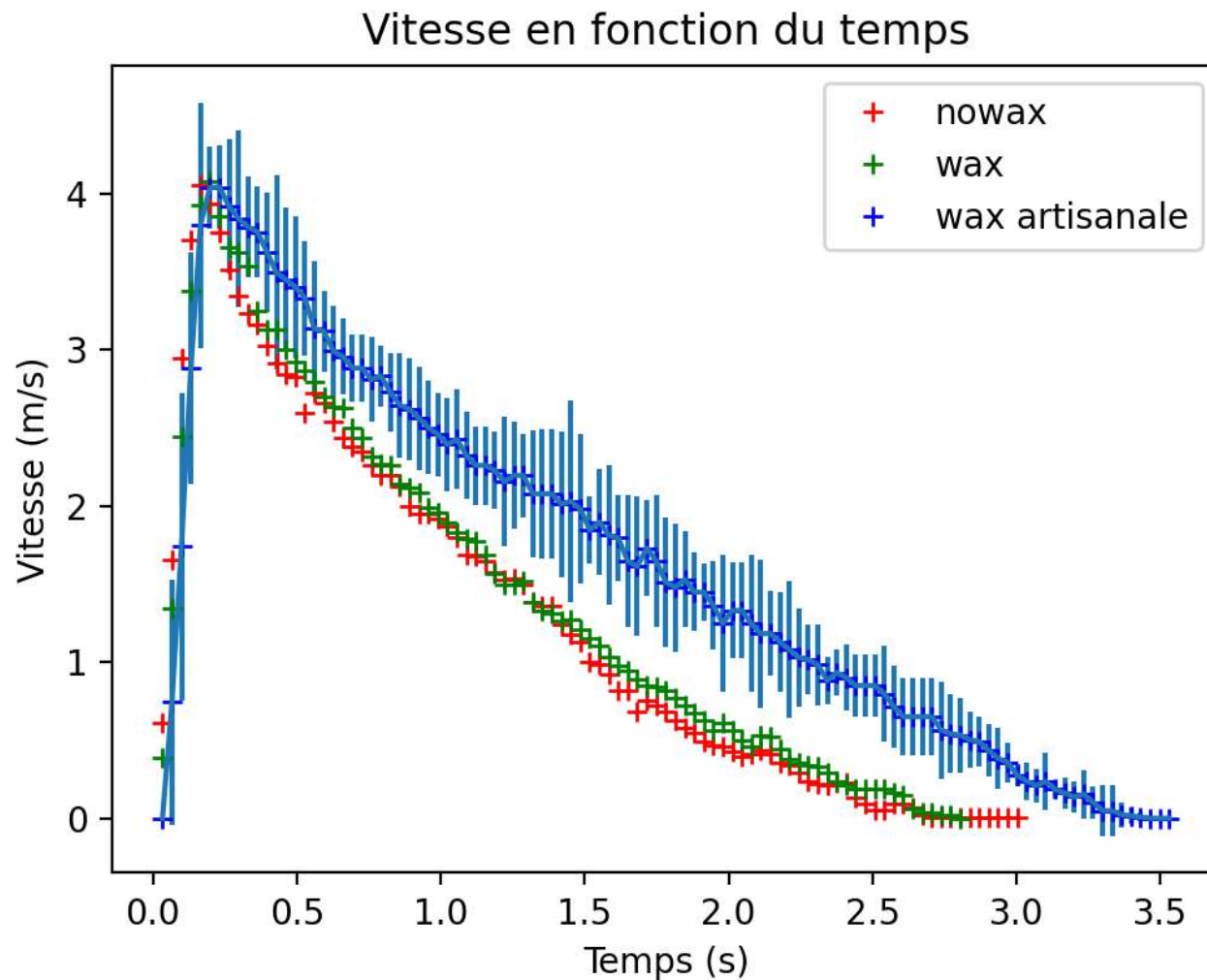
Annexe

Écart-type wax indus. :



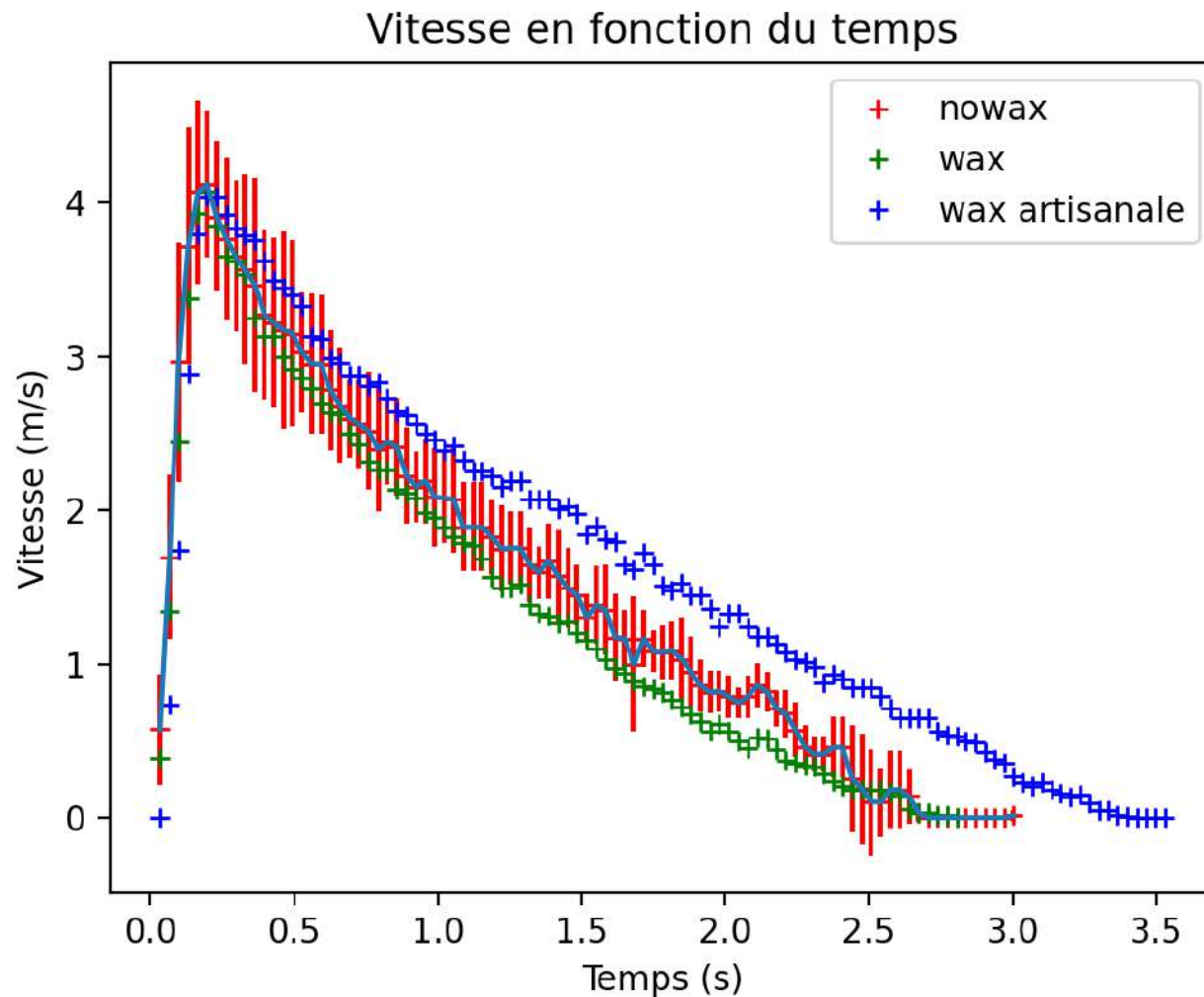
Annexe

Écart-type wax artisanale :



Annexe

Écart-type sans wax (nouvelle courbe) :



Annexe

```
#Créer une nouvelle liste avec pour chaque composante, la moyenne de toutes les listes.
def moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10):
    moy=[]
    for i in range(len(L1)):
        moy.append((L1[i]+L2[i]+L3[i]+L4[i]+L5[i]+L6[i]+L7[i]+L8[i]+L9[i]+L10[i])/10)
    return moy

def moyenneavecwx(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12):
    moy=[]
    for i in range(len(L1)):
        moy.append((L1[i]+L2[i]+L3[i]+L4[i]+L5[i]+L6[i]+L7[i]+L8[i]+L9[i]+L10[i]+L11[i]+L12[i])/12)
    return moy
print(moyennenowax(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8,v_nowax9,v_nowax10))

#fct pour l'étude en 1/v sur toute la courbe
def fctA(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10):
    Anowax=[]
    for i in range (5,len(moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10))-20):
        Anowax.append(1/moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10)[i])
    return Anowax

#fct pour l'étude en 1/v sur le début de la courbe
def fctA2(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10):
    Anowax2=[]
    for i in range (5,len(moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10))-52):
        Anowax2.append(1/moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10)[i])
    return Anowax2
```

Annexe

```
#fct pour ln(v)
def fctln(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10):
    lnnowax=[]
    for i in range (5,len(moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10))-20):
        lnnowax.append(m.log(moyennenowax(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10)[i]))
    return lnnowax

#intervalle de temps sur toute la courbe
def nouveau (L):
    nouveautemps=[]
    for i in range (5,len(L)-20):
        nouveautemps.append(L[i])
    return nouveautemps

#intervalle de temps sur le debut de la courbe
def nouveau2 (L):
    nouveautemps=[]
    for i in range (5,len(L)-52):
        nouveautemps.append(L[i])
    return nouveautemps

moy=moyennenowax(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8,v_nowax9,v_nowax10)

def incertitude (L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,moy):
    s=[]
    for i in range (len(L1)):
        s.append(m.sqrt(((L1[i]-moy[i])**2+(L2[i]-moy[i])**2+(L3[i]-moy[i])**2+(L4[i]-moy[i])**2+(L5[i]-moy[i])**2+(L6[i]-moy[i])**2+(L7[i]-moy[i])**2+(L8[i]-moy[i])**2+(L9[i]-moy[i])**2+(L10[i]-moy[i])**2)/10))
    return s
```

Annexe

```
# Afficher les courbes de vitesse
plt.plot(temps, (moyennenowax(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8,v
plt.plot(temps2,(moyenneavecwax(v_wax1,v_wax2,v_wax3,v_wax4,v_wax5,v_wax6,v_wax7,v_wax8,v_wax9,v_wax10,v
plt.errorbar(temps,(moyennenowax(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8
plt.legend()
plt.xlabel("Temps (s)")
plt.ylabel("Vitesse (m/s)")
plt.title("Vitesse en fonction du temps")
plt.show()
```

Annexe

```
#raccourcis de fonction
Anowax=np.array(fctA(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8))
Anowax2=np.array(fctA2(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8))
lnnowax=np.array(fctlN(v_nowax1,v_nowax2,v_nowax3,v_nowax4,v_nowax5,v_nowax6,v_nowax7,v_nowax8))
nouveautemps=np.array(nouveau(temps))
nouveautemps2=np.array(nouveau2(temps))

#valeurs pour les régressions
modeleln=np.polyfit(nouveautemps,lnnowax,1)
modeleA=np.polyfit(nouveautemps,Anowax,1)
modeleA2=np.polyfit(nouveautemps2,Anowax2,1)

#récupération du R² et de k
R1=np.corrcoef(nouveautemps,lnnowax)
Rln=R1[0][1]
print (Rln)
print('k1/m =',-modeleln[0])

plt.plot(nouveautemps,lnnowax,'b+',label='ln(v)')
#plt.plot(nouveautemps, modeleln[0]*nouveautemps+modeleln[1],'r',label='Regr. linéaire')
plt.legend()
plt.xlabel("Temps (s)")
plt.ylabel("ln(v)")
plt.title("ln(v) en fonction du temps")
plt.savefig('Courbe ln(v) nowax ')
plt.show()
```

Annexe

```
#récupération du R² et de k
R2=np.corrcoef(nouveautemps,Anowax)
RA=R2[0][1]
print (RA)
print('k2/m =',modeleA[0])

plt.plot(nouveautemps,Anowax,'b+',label='1/v')
#plt.plot(nouveautemps, modeleA[0]*nouveautemps+modeleA[1],'r',label='Regr. linéaire')
plt.legend()
plt.xlabel("Temps (s)")
plt.ylabel("1/v")
plt.title("1/v en fonction du temps")
plt.show()

#récupération du R² et de k
R3=np.corrcoef(nouveautemps2,Anowax2)
RA=R3[0][1]
print (RA)
print('k2/m =',modeleA2[0])

plt.plot(nouveautemps2,Anowax2,"b+",label='1/v')
plt.plot(nouveautemps2, modeleA2[0]*nouveautemps2+modeleA2[1],'r',label='Regr. linéaire')
plt.legend()
plt.xlabel("Temps (s)")
plt.ylabel("1/v")
plt.title("1/v en fonction du temps (début de la courbe)")
plt.show()
```

Annexe

```
#valeurs de vitesse et de position
v = np.array([4.049589733391448, 3.934481559978196, 3.7528261513946455, 3.512471197818889, 3.343782
s = np.array([0.27380885996666665, 0.39861171856666666, 0.5235014952166667, 0.6461065321666667, 0.7
t = np.array([0.165,0.198,0.231,0.264,0.297,0.33,0.363,0.396,0.429,0.462,0.495,0.528,0.561,0.594,0.

#equation qui lie v à s
def function(vitesse,a,b) :
    return 1/(2*b)*np.log((a+b*vitesse[0]**2)/(a+b*vitesse**2))

#récupération des coeff a et b
parametres , cov = scipy.optimize.curve_fit(function, v, s)
a , b = parametres

print(np.diag(cov))
print(np.linalg.cond(cov))
print(cov)

#tracer points expérimentaux
plt.scatter(s, v, color="green")
#tracer s=f(v)
plt.plot(1/(2*b)*np.log((a+b*v[0]**2)/(a+b*v**2)), v, color="red", )
plt.xlabel("position (m)")
plt.ylabel("vitesse (m/s)")
plt.title("Ajustement de courbe")
print("Valeur estimee de a : " + str(a))
print("Valeur estimee de b : " + str(b))
plt.savefig('Courbe Curvfit v² moyenne nowax')
plt.show()
```