

Etude des stratégies à adopter pour un « vol campagne » en planeur



Céleste Cordier

N°SCEI : 42080

Thème :

Jeu, sport



1. Introduction

2. Théorie et définition

3. Conception et construction d'un profil d'aile d'ASK13

4. Expérience en soufflerie du prototype d'aile

5. Comparaison de deux stratégies de vol

6. Conclusion et discussion

Introduction : le planeur

- Avion sans moteur
- En « plastique » ou en « bois »
- Biplace ou individuel
- 2 types de vol : campagne ou local



Planeur plastique

Comment assurer un bon vol « campagne » en planeur en comparant deux stratégies de vol pour atteindre un repère de la manière la plus propice à la poursuite du vol ?

Plan

1. Introduction

2. Théorie et définition

3. Conception et construction d'un profil d'aile d'ASK13

4. Expérience en soufflerie du prototype d'aile

5. Comparaison de deux stratégies de vol

6. Conclusion et discussion

Théorie 1

Définition :

Force de trainée :

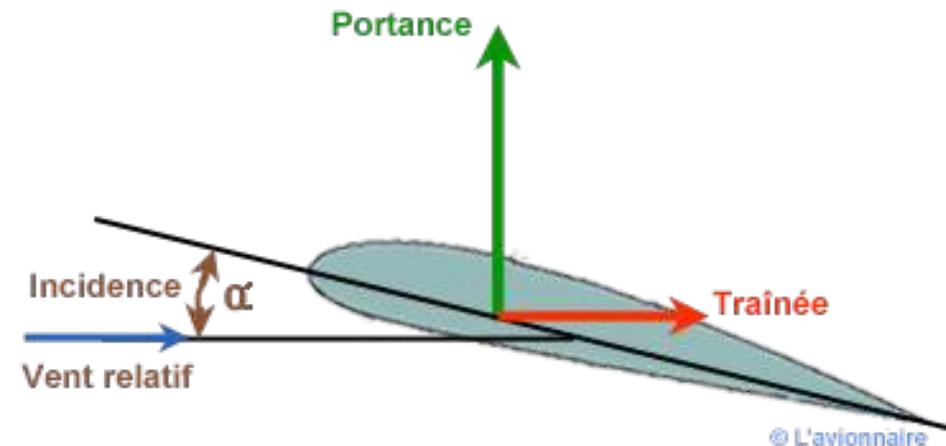
$$F_t = \frac{1}{2} C_x \rho S V^2$$

Force de portance :

$$F_p = \frac{1}{2} C_z \rho S V^2$$

Finesse :

$$F = \frac{\text{portance}}{\text{trainée}} = \frac{C_z}{C_x}$$



Avec :

C_x : coefficient de trainée (sans unité)

C_z : coefficient de portance (sans unité)

ρ : masse volumique de l'air ($1,2 \text{ kg/m}^3$)

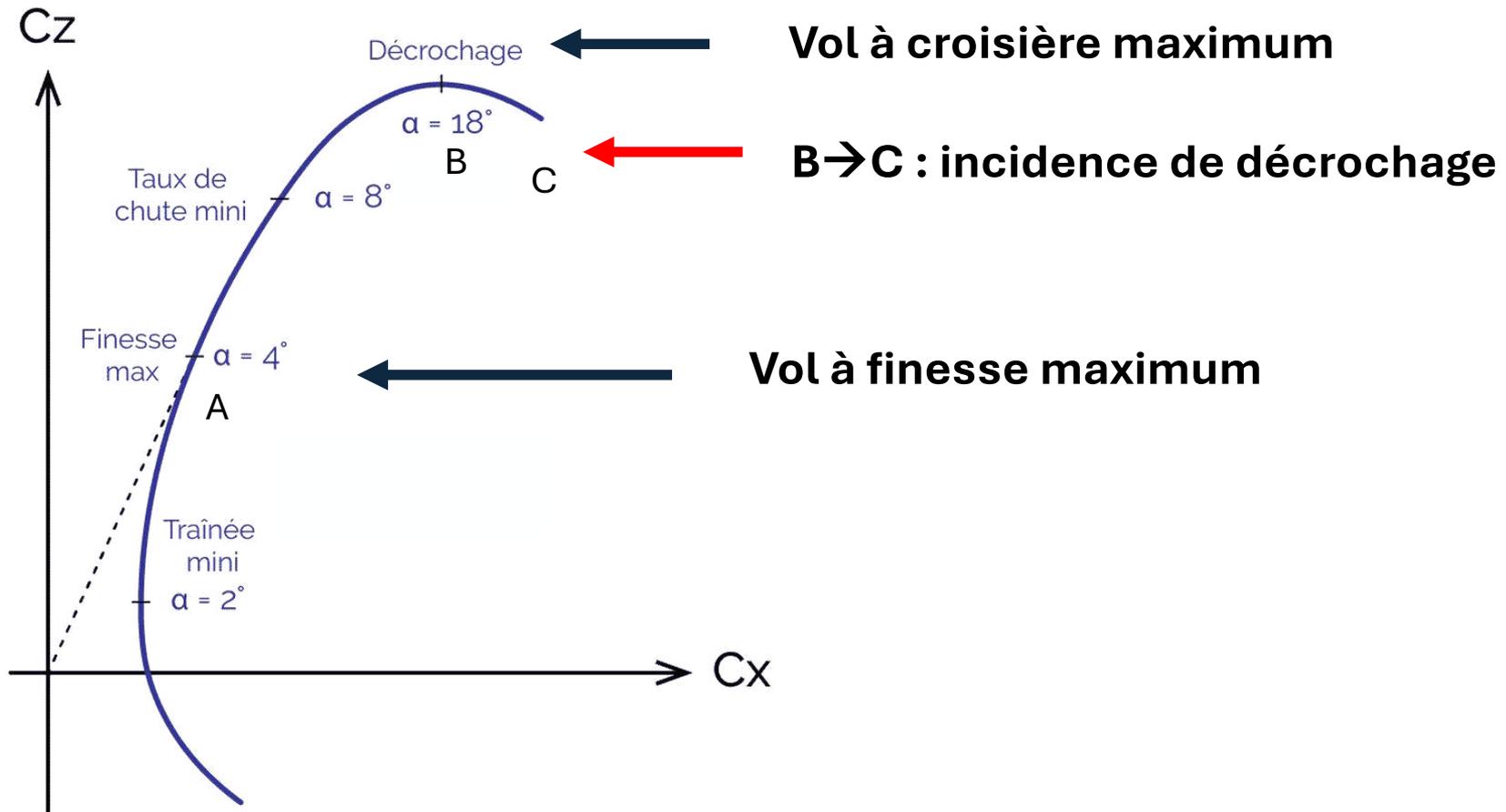
S : surface de référence (m^2)

BUT : obtenir la courbe des polaires

$$\alpha > 0 \text{ (ou } \alpha < 0)$$

Théorie 2

BUT : obtenir la courbe des polaires



Plan

1. Introduction

2. Théorie et définition

3. Conception et construction d'un profil d'aile d'ASK13

4. Expérience en soufflerie du prototype d'aile

5. Comparaison de deux stratégies de vol

6. Conclusion et discussion

Le planeur ASK13

- 1^{er} lâché : 1966
- Créateur : Rudolf Kaiser
- Constructeur : Alexander Schleicher
- Planeur biplace
- Aujourd'hui utilisé pour l'instruction
- « Mauvais planeur »

Bois +
Toile

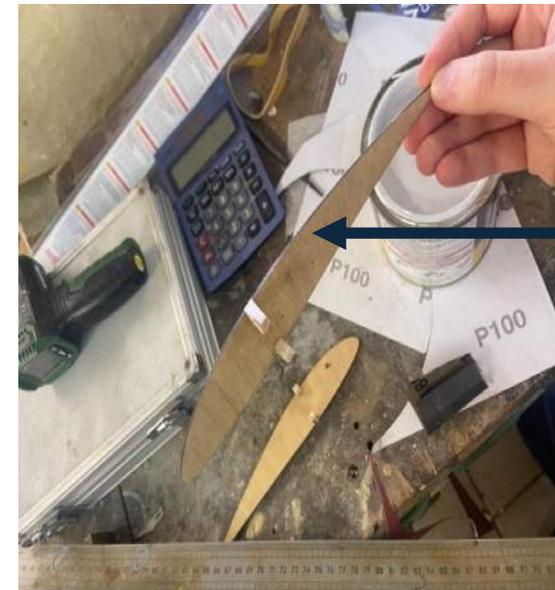
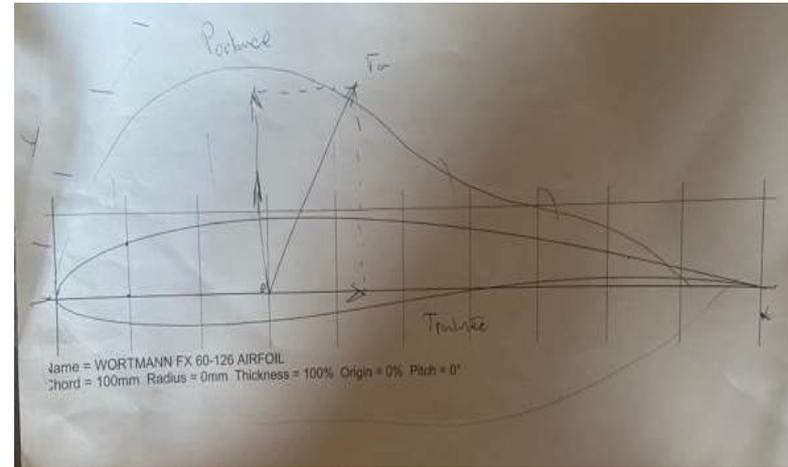


- Pourquoi choisir l'ASK13 ?

- Planeur avec lequel j'ai commencé le pilotage
- Celui que je connais le mieux
- Construction réalisable

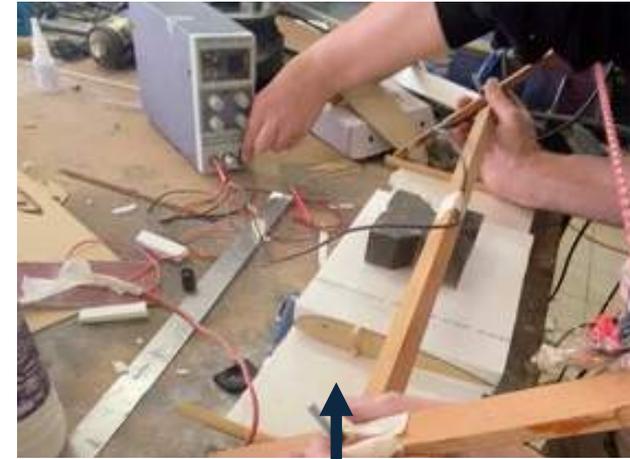
Construction de l'aile : MAI 2023

- Proposition du projet à Francis Durand à l'aérodrome de Champforgeuil (71)
- Conception informatique de l'aile de l'ASK13
- Création du support qui permettra de découper l'aile

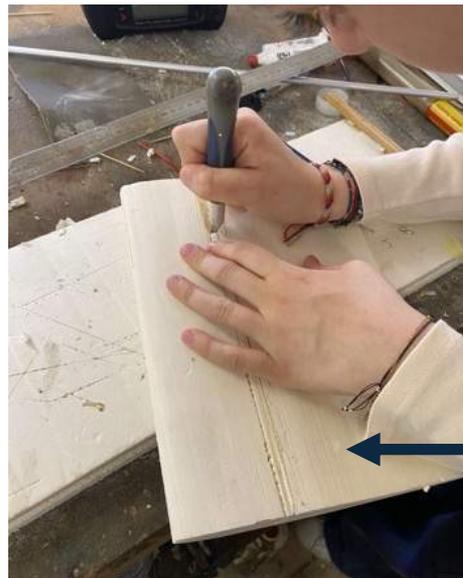
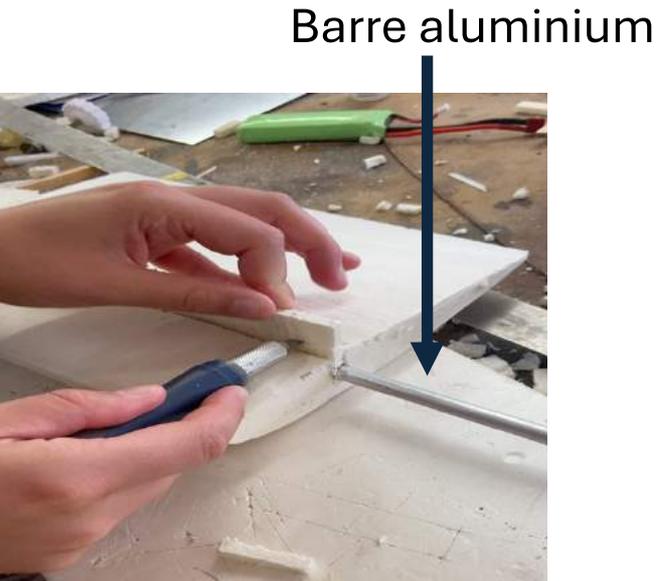


Construction de l'aile : AOUT 2023

- Découpage de la maquette avec le support + fil de fer chaud
- Découpage au 1/3 de l'aile afin d'insérer une barre très légère qui me permettra de créer une inclinaison



Fil de fer



Maquette en polyester



Construction de l'aile : SEPTEMBRE 2023

- Création mastic pour reboucher les trous
- Pose tissu de verre très fin → éviter toutes les forces de frottements
- 1 mois de pause



← Application
colle/ résine



↑ Pose mastic

Construction de l'aile : OCTOBRE-NOVEMBRE 2023

- Équilibrage maquette avec bille de plomb au milieu
- Recouvrement avec du mastic afin de lisser la maquette

→ **BUT**: lisser la maquette pour éliminer les forces de frottements indésirables



Bille

1)



Fait tous les
2-3 jours
pendant 2
mois

2)



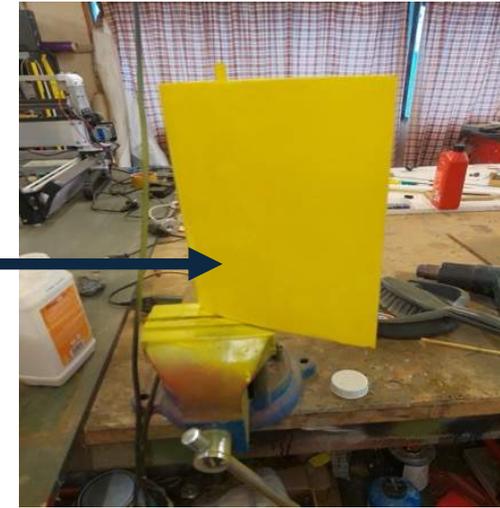
3)



Construction de l'aile : Décembre 2023

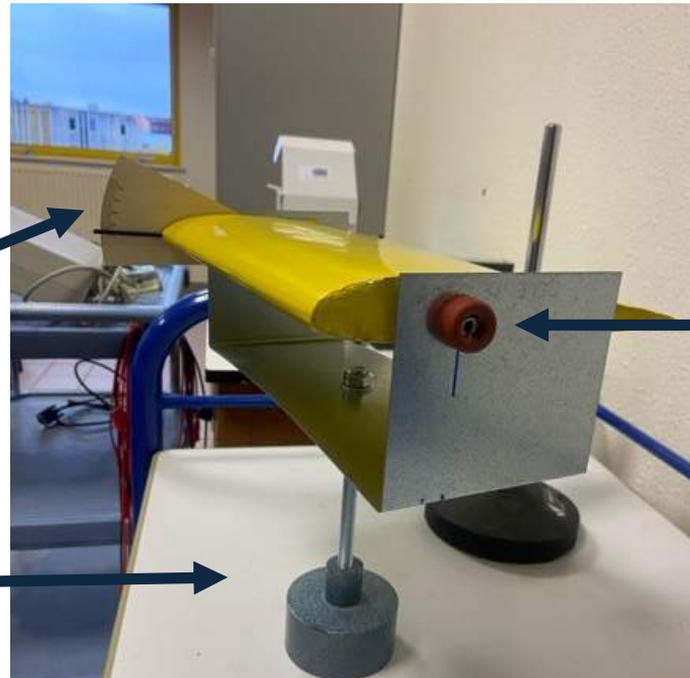
- Revêtement planeur/ avion
- Pose de 1 semaine
- Création indicateur angle + Support

Revêtement



Indicateur
angle

pied support



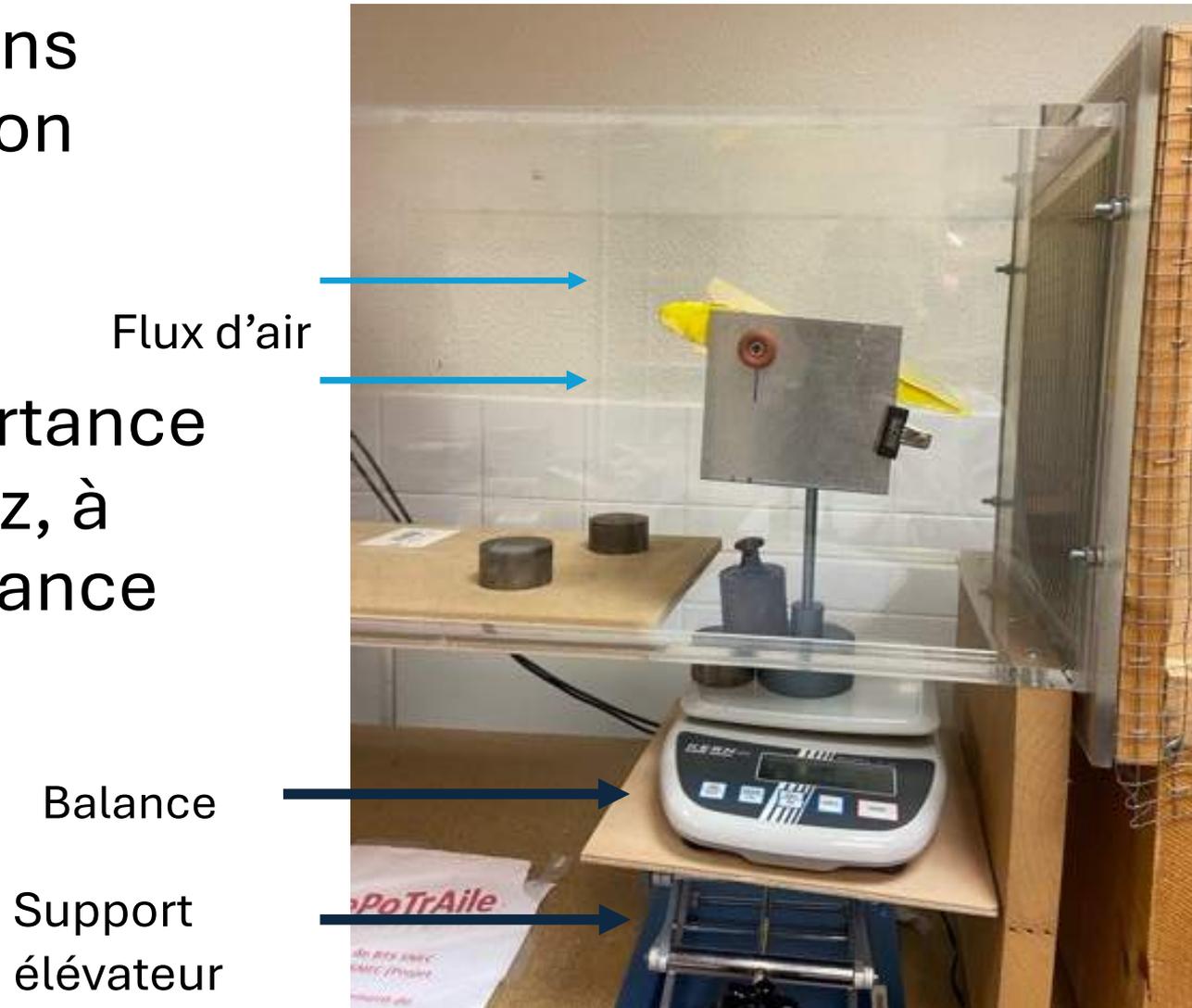
Fixation
caoutchouc

Plan

1. Introduction
2. Théorie et définition
3. Conception et construction d'un profil d'aile d'ASK13
- 4. Expérience en soufflerie du prototype d'aile**
5. Comparaison de deux stratégies de vol
6. Conclusion et discussion

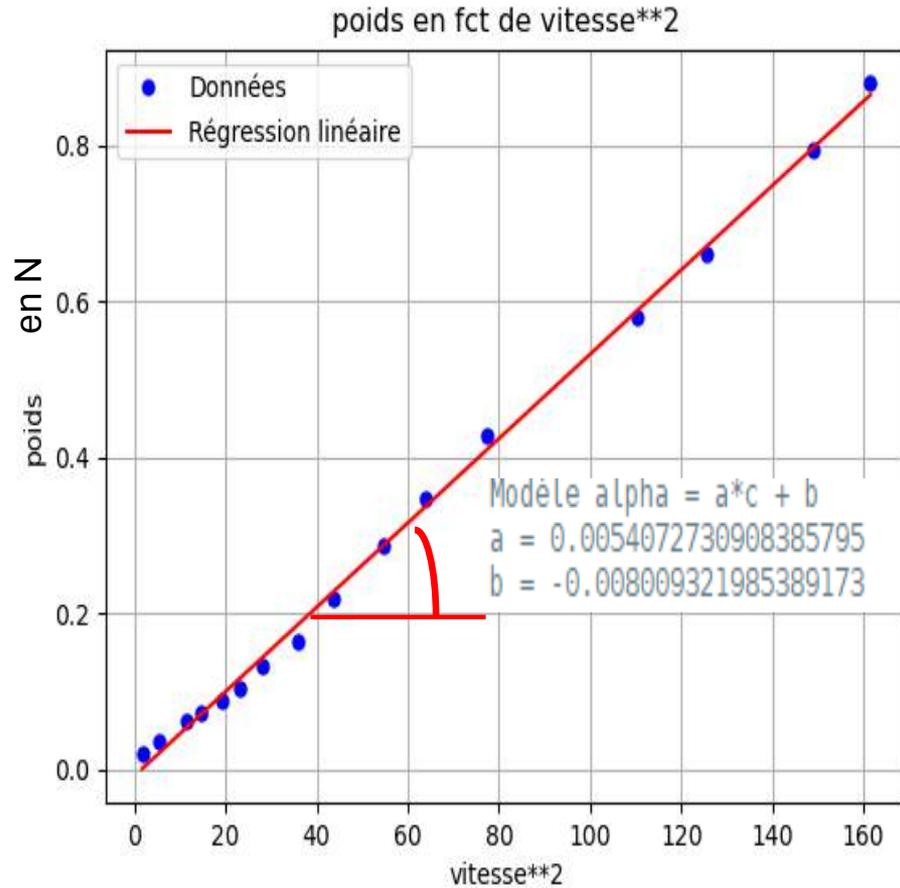
1^{re} série d'expériences en soufflerie

- Expériences dans soufflerie de mon lycée
- **BUT** : calcul portance et coefficient C_z , à l'aide d'une balance

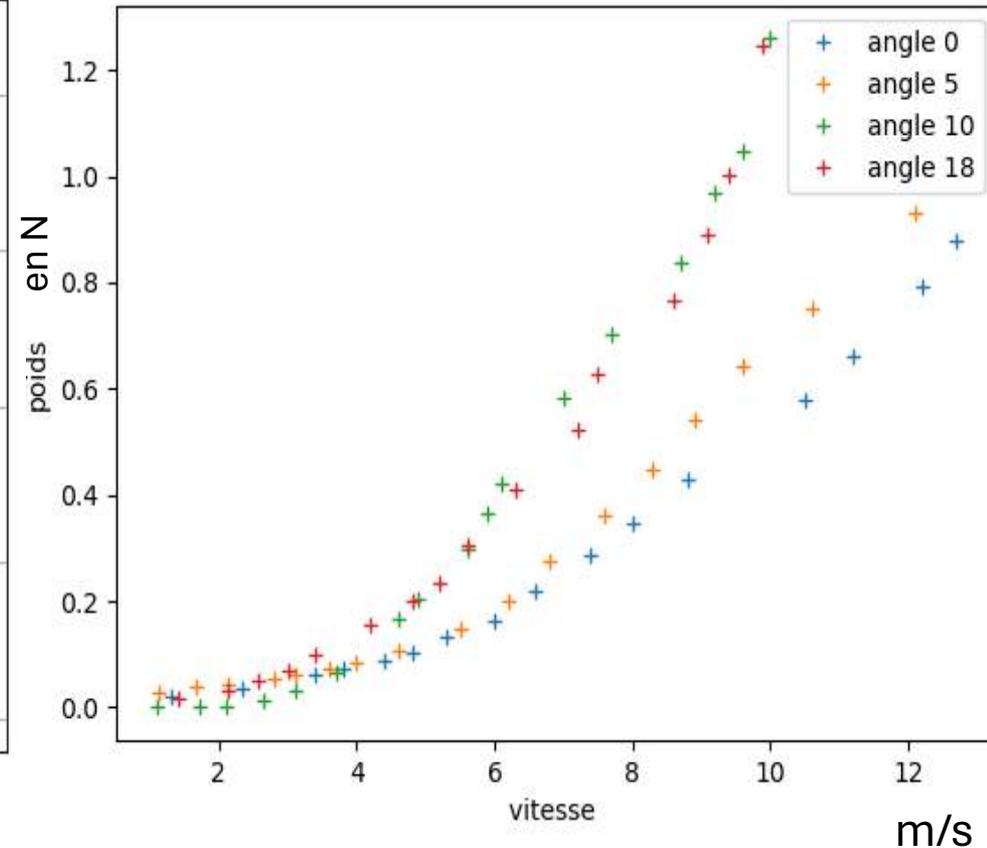


Résultats expérimentaux

- **BUT** : obtention du Coefficient Cz



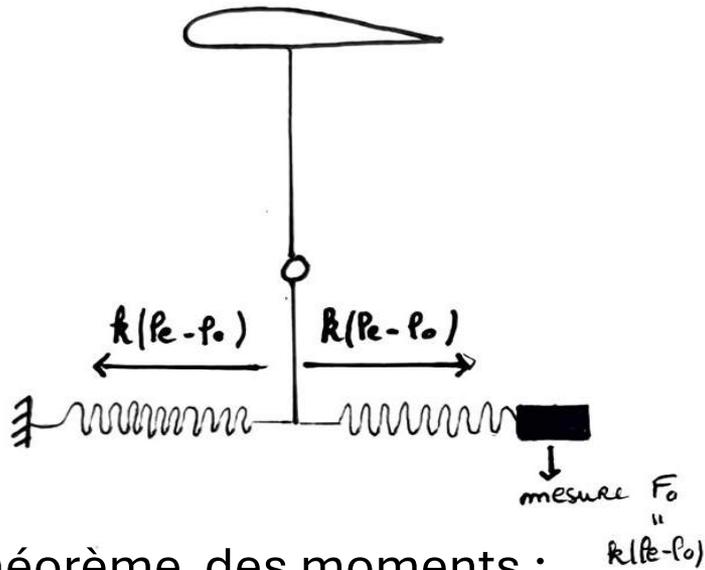
Pour angle : 0°



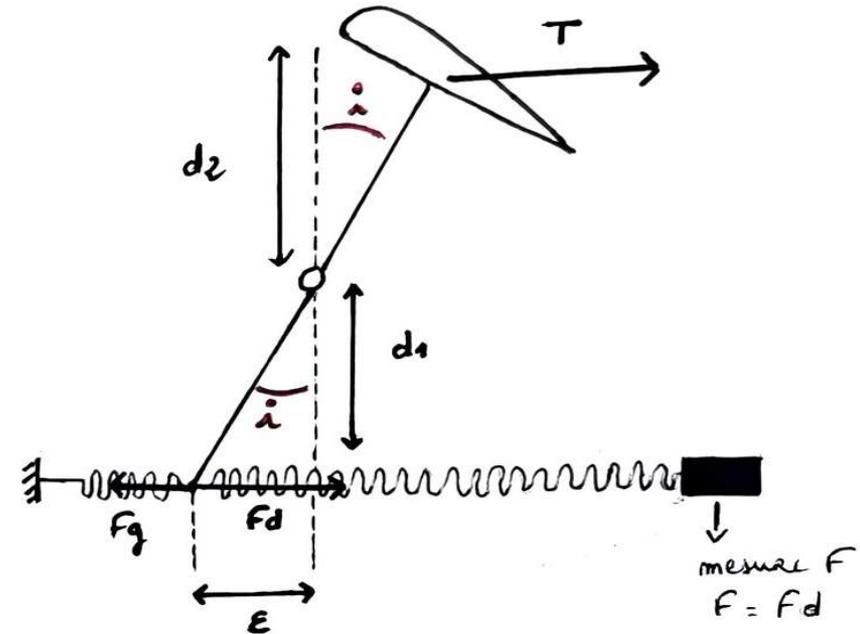
Portance en fonction de la vitesse,
pour un angle défini

Modélisation de la 2ème expérience

Sans Vent



Avec Vent



Théorème des moments :

$$T \times d_2 + Fd \times d_1 - Fgd_1 = 0$$

$$\text{Donc : } T = \frac{d_1}{d_2} (Fd - Fg)$$

$$\text{On a : } Fd = k(le + \varepsilon - lo) = F$$

$$\rightarrow k\varepsilon = F - k(le - lo) = F - F_0$$

$$Fg = k(le - \varepsilon - lo)$$

$$Fd - Fg = 2k\varepsilon = 2(F - F_0)$$

$$\text{Donc } T = \frac{d_1}{d_2} (F - F_0)$$

2ème série d'expériences en soufflerie

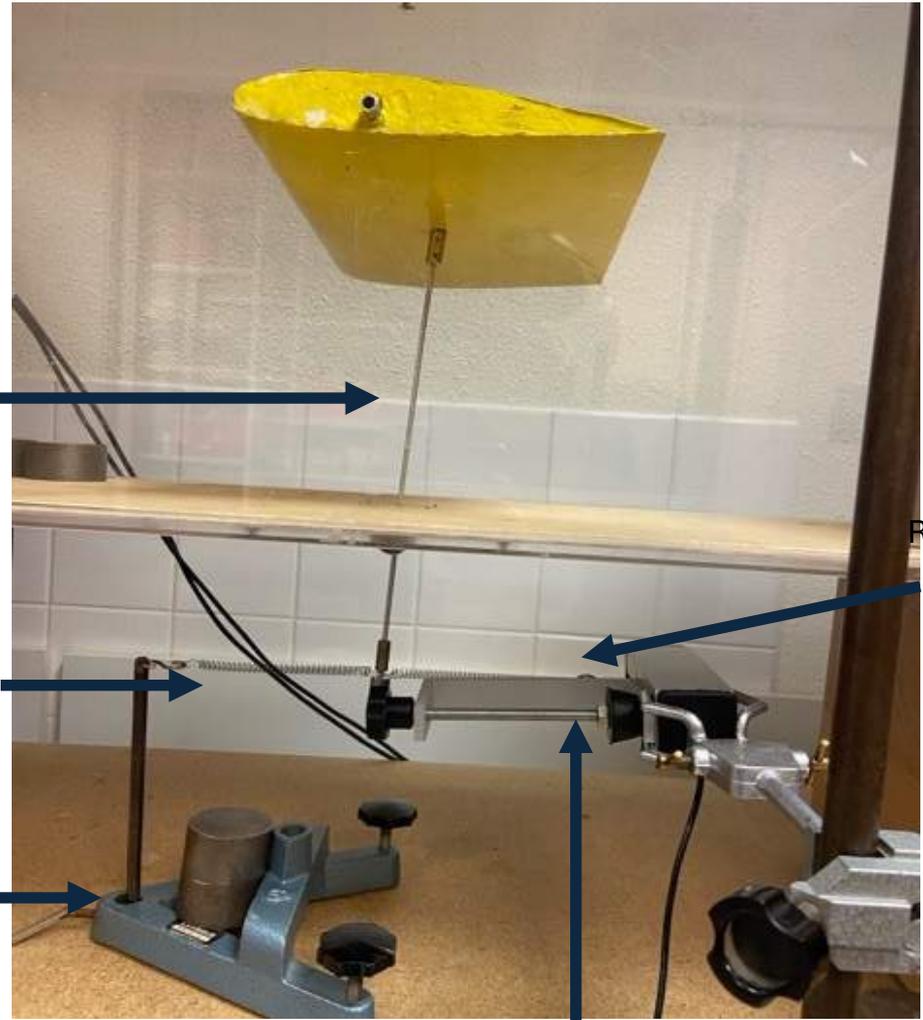
- **BUT** : Calculer force (en Newton) de l'aile → déterminer la traînée

Tige élévatrice de l'aile centrée au milieu

Ressort 2

Support maintenant le ressort

Anémomètre



Ressort 1

Capteur de force à crochet : dynamomètre



Résolution du 1^{er} point Cx

$$F_0 = 0,0983 \text{ N}$$

$$d_1 = 0,090 \text{ m}$$

$$d_2 = 0,149 \text{ m}$$

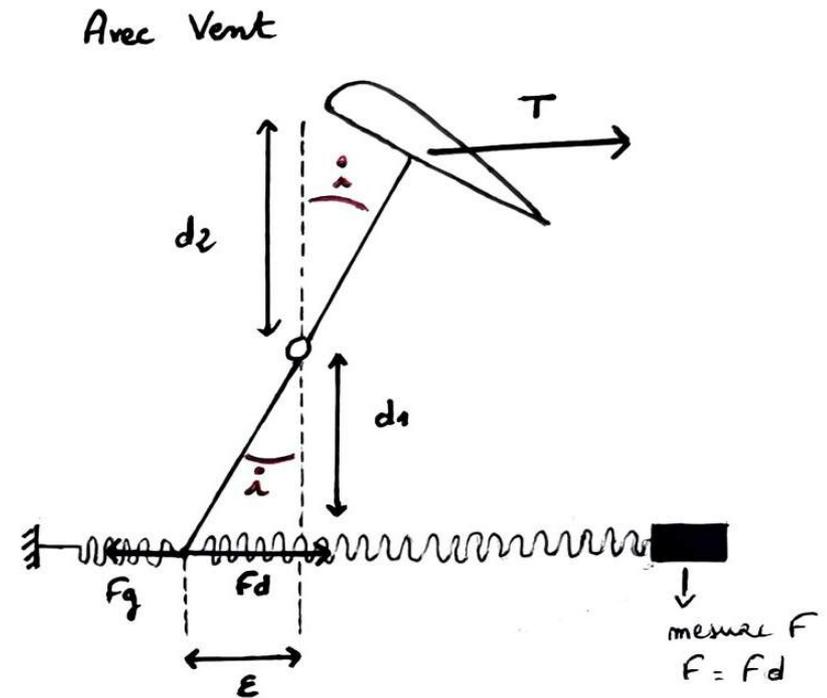
Valeur mesurée du 1er point :

$$V = 1,75 \text{ m/s}$$

$$F = 0,0993 \text{ N}$$

$$F_t = \frac{d_1}{d_2} (F - F_0) = 6,36 \times 10^{-4} \text{ N} \quad (\text{En valeur absolue})$$

$$C_x = \frac{2F_t}{\rho S V^2} = 0,0069$$



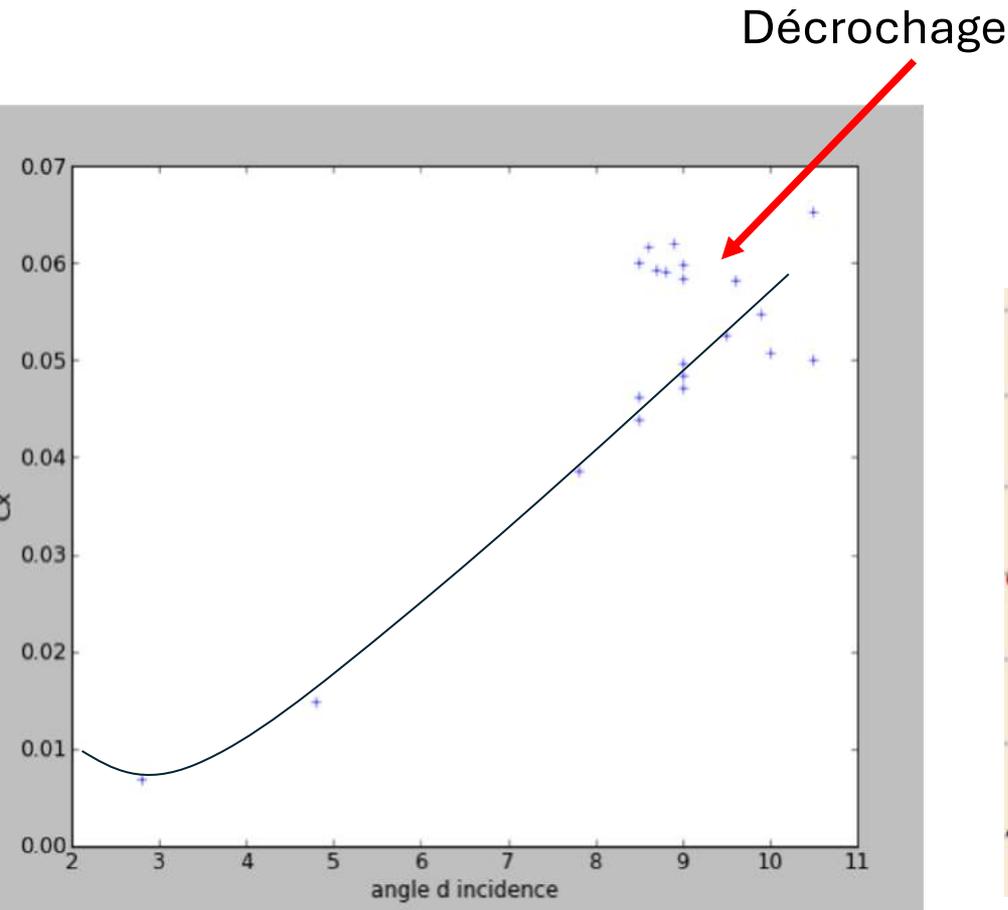
Rappel :

Force de trainée :

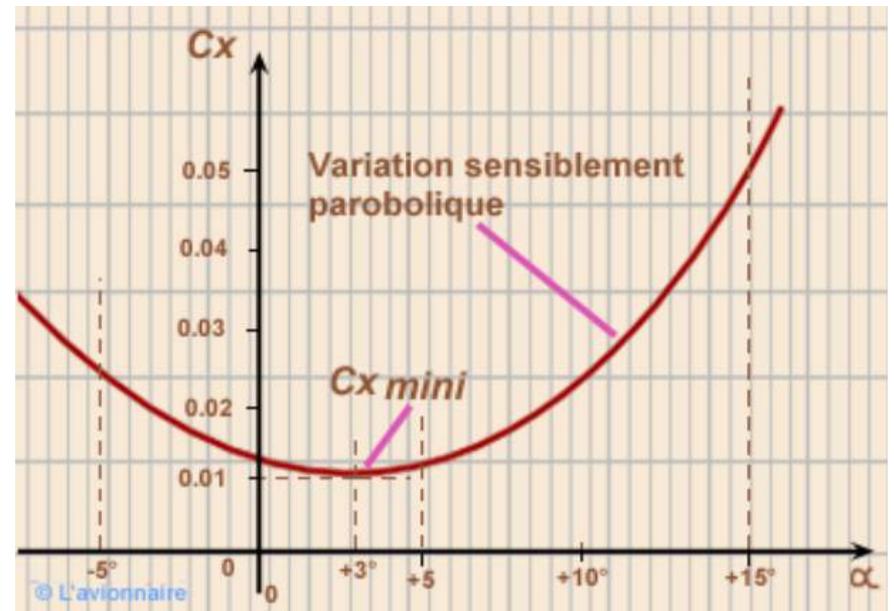
$$F_t = \frac{1}{2} C_x \rho S V^2$$

Résultats expérimentaux

Coefficient C_x en fonction de l'angle d'incidence



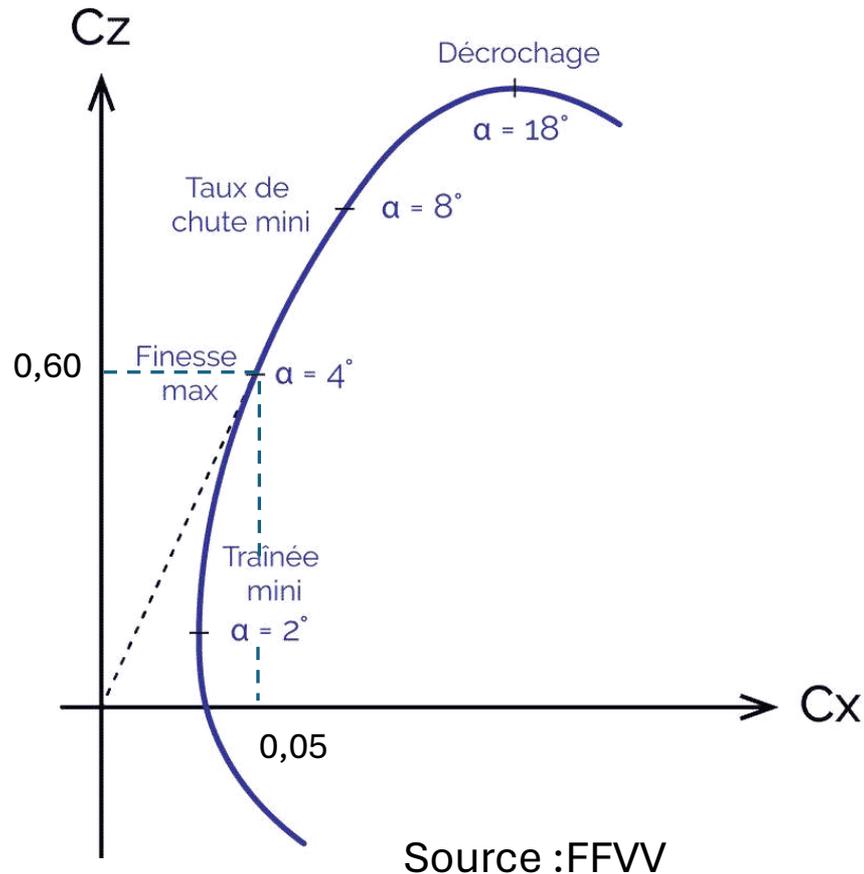
- Courbe à valeur positive \rightarrow traînée ne peut pas « changer de sens », opposée au déplacement de l'aile
- Courbe parabolique, croissance quadratique



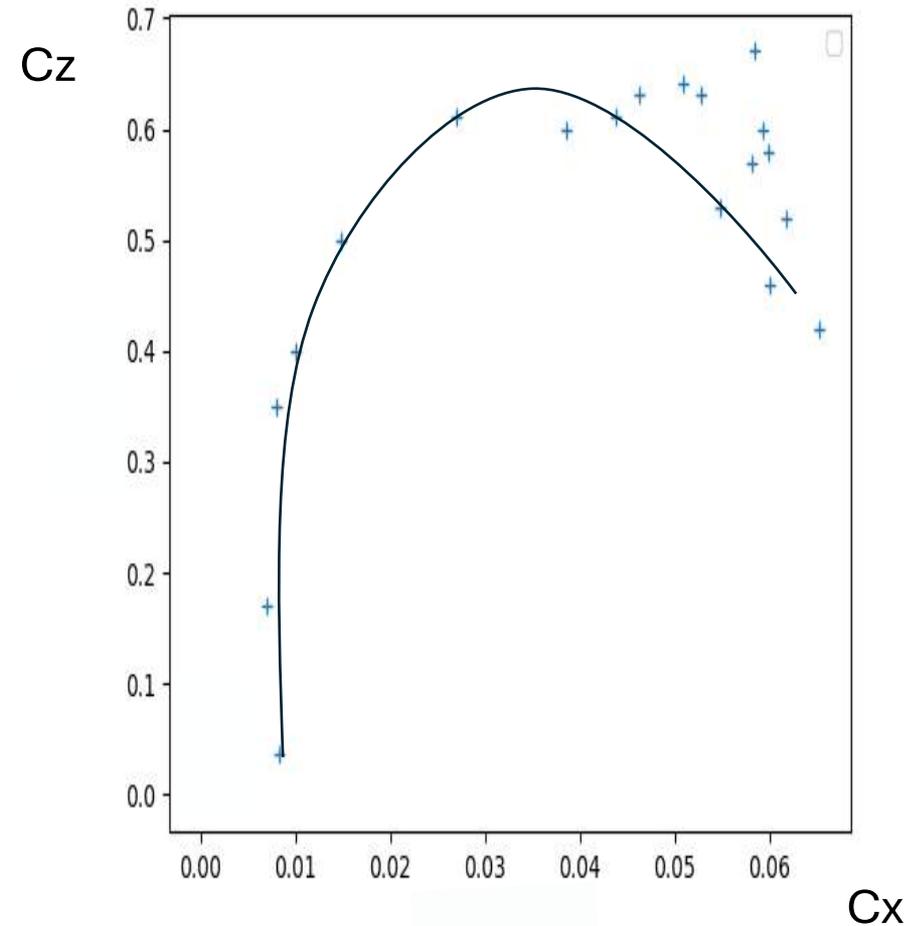
Courbe Théorique

Comparaison courbes des polaires

Courbe théorique



Courbe expérimentale

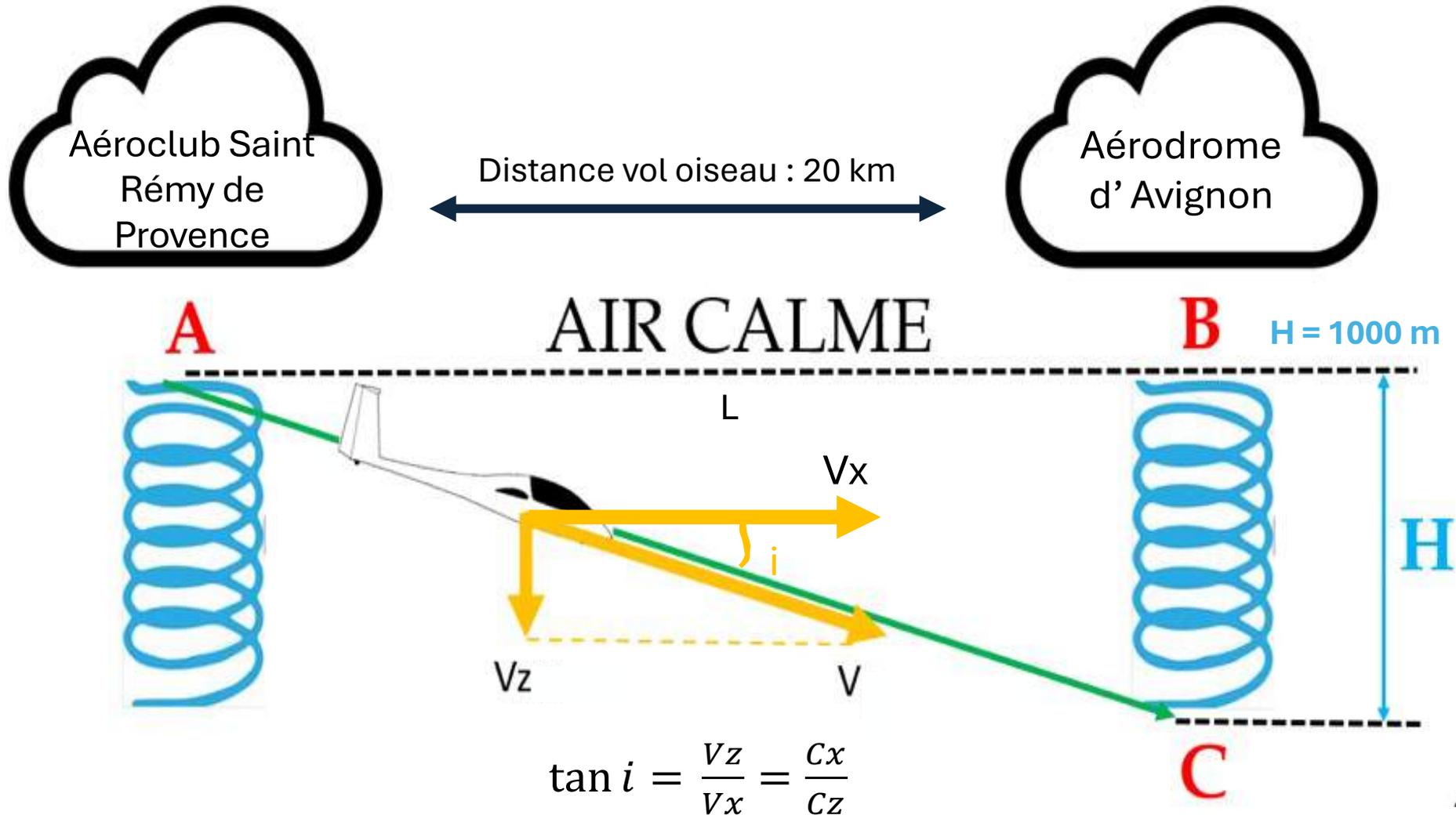


Plan

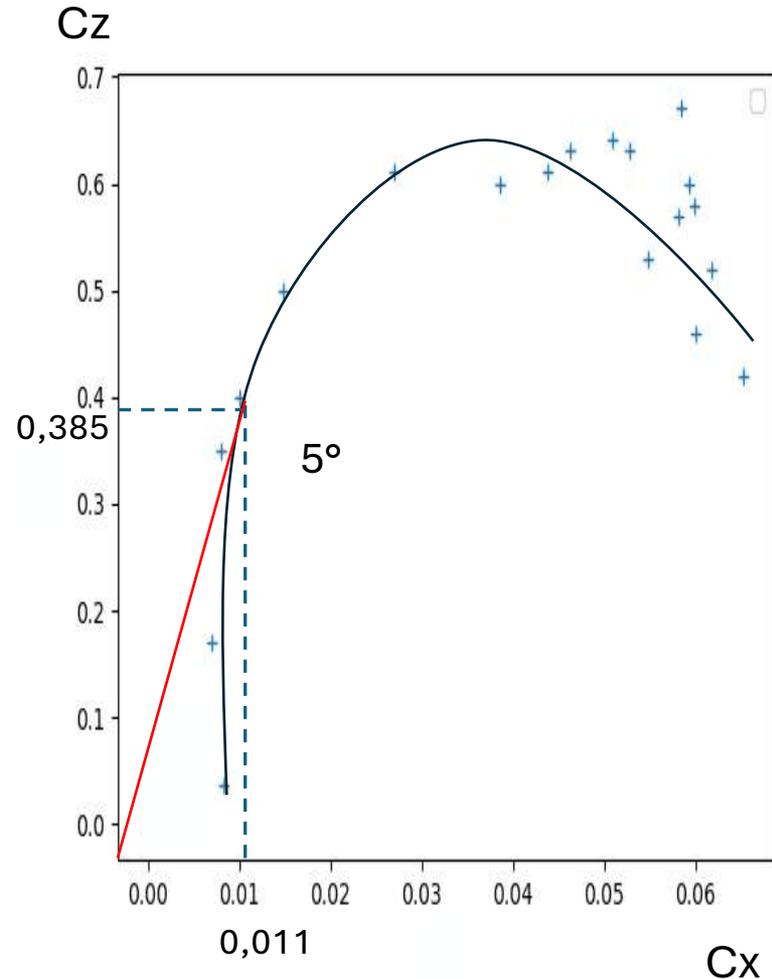
1. Introduction
2. Théorie et définition
3. Conception et construction d'un profil d'aile d'ASK13
4. Expérience en soufflerie du prototype d'aile
- 5. Comparaison de deux stratégies de vol**
6. Conclusion et discussion

Étude d'un cas

- Caractéristique :
- 300 kg (pilote + planeur)
 - Surface alaire : 15 m^2



1^{er} Cas : Vol à finesse maximum



- Recherche maximum du rapport \rightarrow tangente à la courbe

$$\text{Finesse max} = \frac{C_z}{C_x} = \frac{1}{\tan i} = \mathbf{31,9}$$

- Vitesse :

$$V = \sqrt{\frac{2P}{\rho S C_z}} = \sqrt{\frac{2 \times 300 \times 10}{1,2 \times 15 \times 0,385}} = 29,4 \text{ m/s}$$

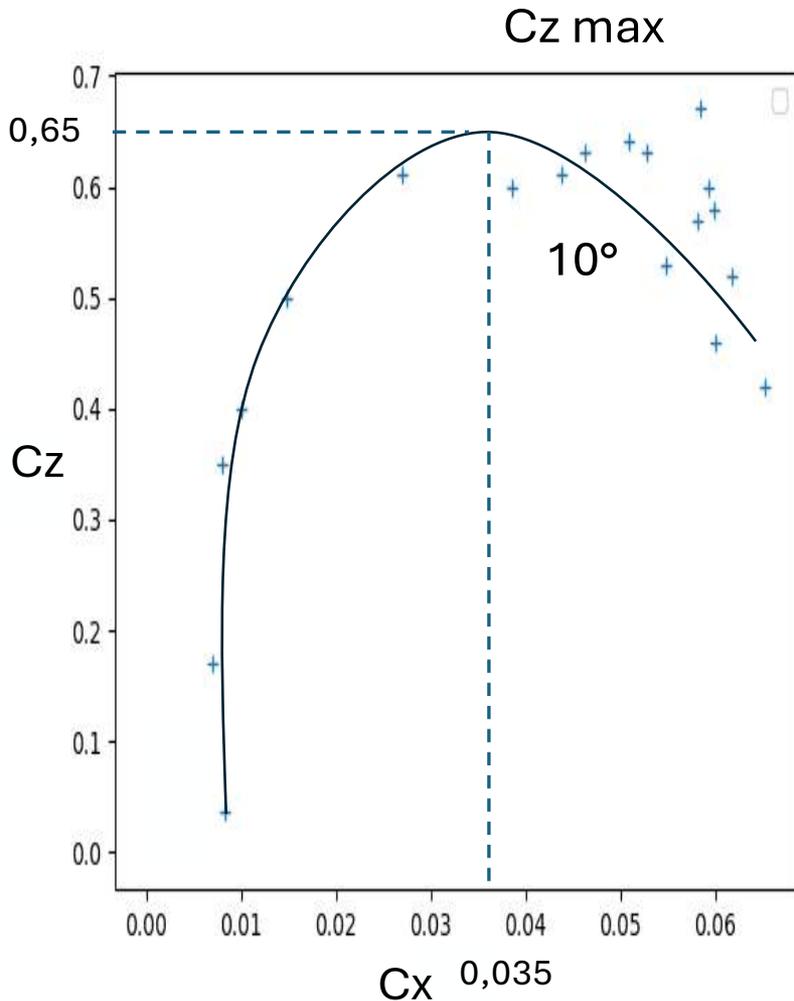
- Temps de vol :

$$T = \frac{H}{V_z} = \frac{H}{V_x \times \tan i} = \frac{H}{V_x \times \frac{C_x}{C_z}} = \frac{1000}{0,92} = \mathbf{1098 \text{ secondes}}$$

- Distance parcourue :

$$L = \frac{H}{\tan i} = \mathbf{31\,900 \text{ mètres}}$$

2ème Cas : Vol à croisière maximum



- Vitesse maximale \rightarrow fonctionner avec C_z maximal \rightarrow portance max

Finesse = 18,0

- Vitesse :

$$V = \sqrt{\frac{2P}{\rho S C_z}} = \sqrt{\frac{2 \times 300 \times 10}{1,2 \times 15 \times 0,65}} = 22,6 \text{ m/s}$$

- Temps de vol :

$$T = \frac{H}{V_z} = \frac{H}{V \times \tan i} = \frac{1000}{1,23} = \mathbf{813 \text{ s}}$$

- Distance parcourue :

$$L = \frac{H}{\tan i} = \mathbf{18 \text{ 000 mètres}}$$

Plan

1. Introduction
2. Théorie et définition
3. Conception et construction d'un profil d'aile d'ASK13
4. Expérience en soufflerie du prototype d'aile
5. Comparaison de deux stratégies de vol
- 6. Conclusion et discussion**

Conclusion

→ Finesse max ASK13 théorique : \approx **18-19**

→ Finesse max plastique théorique : \approx **22-25**

- Résultat supérieur aux attentes théoriques mais cohérents dans le domaine aéronautique

- Le profil d'aile n'est pas parfait

- Sources d'erreurs :

 - **Mesures** : forces, vitesse flux d'air, incertitudes C_x - C_z

 - **Effet de bord** : profil large, proche des 2 parois de la cage d'air

 - **Nombre de Reynolds** :

C_z et C_x sont fonction de l'angle d'attaque et du nombre de Reynolds → un profil donné a une infinité de polaires car infinité de nombres de Reynolds



Merci de votre attention

Annexe 1 - Limitations des résultats

- **Modèle réducteur/ simplifié**
- Variations des masses d'air ascendantes ou descendantes, air turbulent
- On ne prend pas en compte le décollage et l'atterrissage
- Les conditions pratiques en vol (sécurité)
- Autres perspectives de recherches



Annexe 2 - La soufflerie

Flux d'air allant jusqu'à 60 km/h (16,7 m/s)

Réalisée en 2012 par les BTS CIEL :
Cybersécurité, Informatique et réseaux
Électronique + PSI

Disposition pour les TP, TD et TIPE

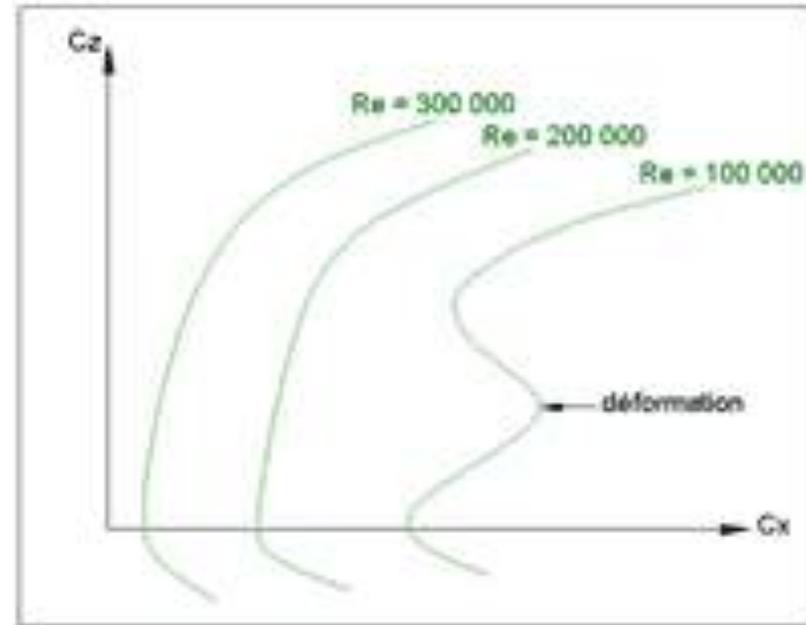
Construction :

- Convergent + tube : construit par un compagnon du devoir (ébéniste)
- Plexiglass : crée par entreprise locale de plastique
- Grille de protection + support Ellice créés par un chaudronnier
- PARTICULARITÉ : soufflerie fonctionne par inspiration et non par soufflement



Annexe 3- Nombre de Reynolds

- Les effets de la vitesse et de la corde sur l'efficacité du profil sont similaires
- Pour le calcul des performances : Re qui se rapproche le plus des conditions de vol du futur modèle.
- Les profils creux et/ou épais supportent mal les faibles nombres de Reynolds
- Re compris entre 60 000 et 500 000, $\sim 1\,000\,000$ pour modèles très gros et rapides.



$$Re = \frac{\mu \times V \times L}{\eta}$$

Avec :

μ masse volumique de l'air : $1,2 \text{ Kg.m}^{-3}$

η coefficient de viscosité dynamique : $18,5 \times 10^{-6} \text{ Pa.s}$

L longueur de la corde : $19 \text{ cm} = 19 \times 10^{-2} \text{ m}$

V vitesse de l'écoulement d'air en m.s^{-1}

Annexe 4 - Incertitudes 1

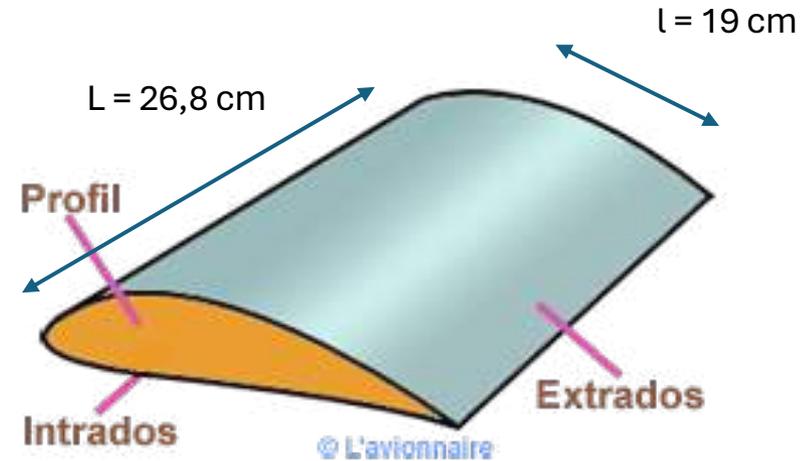
- **Surface de l'aile :**

Incertitude des demi-étendues : 0,05cm

$$u(L) = \frac{0,05}{\sqrt{3}} = 0,029 \text{ cm} = 2,9 \times 10^{-4} \text{ m}$$

$$u(l) = \frac{0,05}{\sqrt{3}} = 0,029 \text{ cm} = 2,9 \times 10^{-4} \text{ m}$$

$$S = 509,2 \text{ cm}^2 = 0,05 \text{ m}^2$$



$$\text{Incertitude sur } S : u(S) = S \times \sqrt{\left(\frac{u(L)}{L}\right)^2 + \left(\frac{u(l)}{l}\right)^2} = 0,95 \text{ cm}^2 = 9,5 \times 10^{-5} \text{ m}^2$$

- **Masse mesurée expérimentalement :**

Incertitude sur les mesures : 0,1 g

$$u(m) = \frac{0,1}{\sqrt{3}} = 0,05 \text{ g} = 5 \times 10^{-5} \text{ Kg}$$



Balance

Annexe 5 - Incertitudes 2

- **Force permettant de trouver la traînée :**

Incertitudes sur les mesures : 0,005 N

$$u(\text{mesure}) = \frac{0,005}{\sqrt{3}} = 2,89 \times 10^{-03} \text{ N}$$



Dynamomètre

- **Vitesse mesurée :**

Incertitudes sur les mesures : 0,01 m/s

$$u(\text{mesure}) = \frac{0,01}{\sqrt{3}} = 5,77 \times 10^{-03} \text{ m/s}$$



Anémomètre

Annexe 6 - Incertitudes 3

• Incertitude coefficient Cx

$$\frac{u(Cx)}{Cx} = \sqrt{\left(\frac{u(Ft)}{Ft}\right)^2 + \left(\frac{u(S)}{S}\right)^2 + 2^2 \left(\frac{u(V)}{V}\right)^2}$$

	U(Cx)		U(Cx)
1	$4,33 \times 10^{-5}$	8	$1,95 \times 10^{-4}$
2	$4,53 \times 10^{-5}$	9	$2,89 \times 10^{-4}$
3	$5,88 \times 10^{-4}$	10	$3,46 \times 10^{-4}$
4	$5,88 \times 10^{-5}$	11	$3,79 \times 10^{-4}$
5	$5,82 \times 10^{-5}$	12	$5,82 \times 10^{-5}$
6	$7,08 \times 10^{-5}$	13	$5,52 \times 10^{-4}$
7	$1,02 \times 10^{-4}$	14	$5,16 \times 10^{-4}$

15	$5,51 \times 10^{-4}$
16	$9,45 \times 10^{-4}$
17	$5,56 \times 10^{-4}$
18	$5,65 \times 10^{-4}$
19	$5,86 \times 10^{-5}$
20	$5,62 \times 10^{-4}$

• Incertitude coefficient Cz

$$\frac{u(Cz)}{Cz} = \sqrt{\left(\frac{u(Fp)}{Fp}\right)^2 + \left(\frac{u(S)}{S}\right)^2 + 2^2 \left(\frac{u(V)}{V}\right)^2}$$

	U(Cz)		U(Cz)
1	$8,78 \times 10^{-4}$	8	$4,38 \times 10^{-3}$
2	$9,07 \times 10^{-4}$	9	$4,69 \times 10^{-3}$
3	$2,02 \times 10^{-3}$	10	$5,06 \times 10^{-3}$
4	$2,17 \times 10^{-4}$	11	$3,49 \times 10^{-3}$
5	$2,52 \times 10^{-3}$	12	$3,88 \times 10^{-3}$
6	$3,28 \times 10^{-3}$	13	$4,57 \times 10^{-3}$
7	$4,25 \times 10^{-3}$	14	$5,22 \times 10^{-3}$

15	$5,57 \times 10^{-3}$
16	$9,40 \times 10^{-3}$
17	$5,58 \times 10^{-3}$
18	$5,39 \times 10^{-3}$
19	$6,54 \times 10^{-3}$
20	$6,78 \times 10^{-3}$

Annexe 7 - Incertitudes 4

Calcul des incertitudes pour chaque point Cx et Cz :

```
import numpy as np

Cx=np.array([ 0.0069522,0.0080,0.0083,0.010 , 0.01480834,0.027, 0.03862334, 0.0438564, 0.04630166, 0.06527692, 0.06005605,0.06177002, 0.05983239, 0.05924879, 0.05844608,
0.05817882, 0.0547573, 0.05269348, 0.05083798])

Cz=np.array([ 0.17,0.35,0.036,0.4,0.5,0.61,0.6,0.61,0.63,0.42,0.46,0.52,0.58,0.60,0.670,0.57,0.53,0.63,0.64])

T = np.array([0.0006081786125050719, 0.0024327144500206487, 0.007602232656314124, 0.009730857800081637, 0.011251304331344312, 0.01569699295706425, 0.01860769342978088,
0.021590340743931253, 0.022198519356436686, 0.024935323112709632, 0.025847591031467488, 0.026455769643972317, 0.028280305481488008, 0.02949666270649827, 0.02949666270649827,
0.0301048413190031, 0.03162528785026578, 0.03223346646277121, 0.0337539129940339, 0.033145734381528465])

vitesse = np.array([5.4,7.4,8.1,8.6,9,9.4,10,10.5,11.1,11.6,12,12.2,12.7,13,13.4,13.8,14.4,14.7,15,15.3])

UmesureT=2.89*10**(-4)

US = 9.5*10**(-5)

UV = 5.77*10**(-3)

incertitudeCx = Cx*((UmesureT/T)**2 + (US/0.05)**2 + (2**2)*(UV/vitesse)**2 )**1/2
##

UmesureP = 4.9*10**(-4)

P= np.array([0.0294 , 0.03724, 0.04116, 0.05488, 0.06272, 0.07448, 0.08232, 0.10486, 0.147 , 0.19894, 0.2744 ,0.3056, 0.3626 ,0.44786, 0.54292,0.5980,0.64288, 0.75068,0.8609,
0.93002])

incertitudeCz = Cz*((UmesureP/P)**2 + (US/0.05)**2 + (2**2)*(UV/vitesse)**2 )**1/2
```

Annexe 8 - Incertitudes 5

Tracé des barres Incertitudes de la polaire des vitesses

```
Cx=np.array([0.0069522,0.0080,0.0083,0.010 , 0.01480834,0.027579,  
0.03862334, 0.0438564, 0.04630166, 0.06527692, 0.06005605,0.06177002,  
0.05983239, 0.05924879, 0.05844608,0.05817882, 0.0547573,0.05269348,  
0.05083798])
```

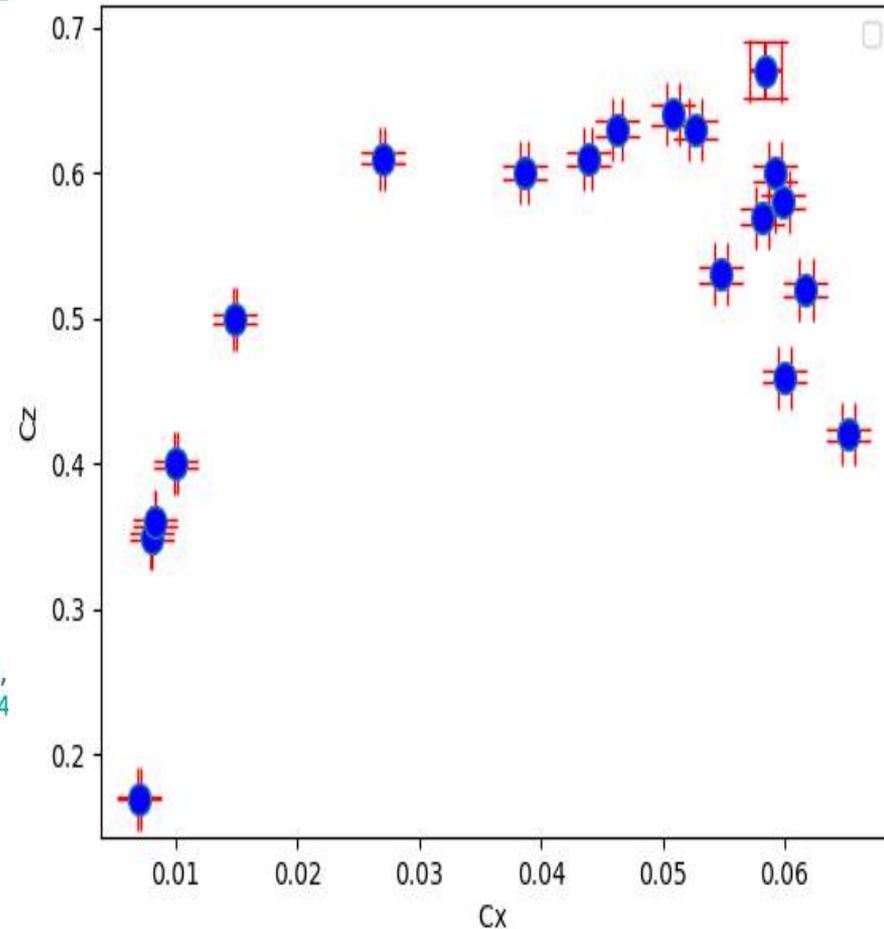
```
Cz=np.array([0.17,0.35,0.036,0.4,0.5,0.61,0.6,0.61,0.63,0.42,0.46,  
0.52,0.58,0.60,0.67,0.57,0.53,0.63,0.64])
```

```
incertitudeCx = np.array([ 1.53389593e-04, 5.87756302e-05,  
5.82446902e-05,7.02765122e-05,1.02468639e-04, 1.94880443e-04,  
2.89271113e-04,3.46174427e-04, 3.79551620e-04,5.52448217e-04, 5.16170323e-04,  
5.51181834e-04, 5.45714856e-04, 5.56801545e-04, 9.65175876e-04,5.86157504e-04  
5.62854420e-04, 5.52138134e-04, 5.43421333e-04])
```

```
incertitudeCz= np.array([ 0.00090709, 0.00202399, 0.00021797, 0.00252367,  
0.00328242,0.00425222, 0.00438044,0.00469902, 0.00506737, 0.00349312,  
0.00388916, 0.004576, 0.00522404,0.00557008, 0.01940537, 0.00568609,  
0.00539706, 0.00654615, 0.00678297])
```

```
pyplot.errorbar(Cx,Cz, xerr = incertitudeCx ,yerr = incertitudeCz ,  
capsize = 10 , ecolor = 'red', marker = 'o' , markersize = 10 ,  
markerfacecolor = 'blue' ,linestyle = 'none')
```

```
plt.show()
```



Barres d'erreurs de Cz en fonction de Cx

Code python 1

Tracé de la portance en fonction de la vitesse, pour un angle d'incidence défini

```
###poids
poids0=np.array([0.02156, 0.03528, 0.06272, 0.07154, 0.0882 , 0.10388, 0.1323 ,0.16366, 0.2205 , 0.28616, 0.34692, 0.42924, 0.58016, 0.66052,0.7938 , 0.88004])
vitesse0 =np.array([1.3,2.35,3.4,3.8,4.4,4.8,5.3,6,6.6,7.4,8,8.8,10.5,11.2,12.2,12.71])
vitesse1=np.array([1.14,1.66,2.14,2.80,3.1,3.6,4.0,4.6,5.5,6.2,6.8,7.6,8.3,8.9,9.6,10.6,12.1])
poids10 =np.array([0.0294 , 0.03724, 0.04116, 0.05488, 0.06272, 0.07448, 0.08232, 0.10486, 0.147 , 0.19894, 0.2744 , 0.3626 ,0.44786, 0.54292,0.64288, 0.75068, 0.93002])
vitesse2=np.array([1.09, 1.71, 2.10, 2.65, 3.1, 3.7, 4.6, 4.9, 5.6, 5.9, 6.1, 7.0, 7.7, 8.7, 9.2, 9.6, 10])
poids20= np.array([0. , 0. , 0. , 0.01274, 0.03038, 0.06566, 0.16464,0.20384, 0.29596, 0.36554, 0.42042, 0.5831 , 0.70364, 0.83594,0.96824, 1.04566, 1.2593 ])
vitesse3=np.array([1.4, 2.12, 2.57, 3.0, 3.4, 4.2, 4.8, 5.2, 5.6, 6.3, 7.2, 7.5, 8.6, 9.1, 9.4, 9.9])
poids30=np.array([0.01568, 0.03136, 0.049 , 0.06958, 0.09996, 0.15386, 0.19894,0.2352 , 0.3038 , 0.40964, 0.5243 , 0.6272 , 0.76538, 0.88886,1.00058, 1.2446 ])

plt.close('all')
plt.figure('titre')
plt.plot(vitesse0, poids00, '+' , label = 'angle 0')
plt.plot(vitesse1, poids10, '+' , label = 'angle 5')
plt.plot(vitesse2, poids20, '+' , label = 'angle 10')
plt.plot(vitesse3, poids30, '+' , label = 'angle 18')
plt.legend()
plt.plot()
plt.xlabel('vitesse ')
plt.ylabel('poids')
plt.show()
```

Calcul exécuté préalablement :

$$P = m(liste) \times 10^{-3} \times g$$

Code python 2

Régression linéaire portance en fonction de V^2 :

```
#vitesse0 =np.array([1.3,2.35,3.4,3.8,4.4,4.8,5.3,6,6.6,7.4,8,8.8,10.5,11.2,12.2,12.71])
#m0= np.array([2.2,3.6,6.4,7.3,9.0,10.6,13.5,16.7,22.5,29.2,35.4,43.8,59.2,67.4,81.0,89.8])
#vitesse1=np.array([1.14,1.66,2.14,2.80,3.1,3.6,4.0,4.6,5.5,6.2,6.8,7.6,8.3,8.9,9.6,10.6,12.1])
#m1=np.array([3.0,3.8,4.2,5.6,6.4,7.6,8.4, 10.7, 15.0, 20.3, 28.0, 37.0, 45.7, 55.4, 65.6, 76.6, 94.9])
#vitesse2=np.array([1.09, 1.71, 2.10, 2.65, 3.1, 3.7, 4.6, 4.9, 5.6, 5.9, 6.1, 7.0, 7.7, 8.7, 9.2, 9.6, 10])
#m2=np.array([0, 0, 0, 1.3, 3.1, 6.7, 16.8, 20.8, 30.2, 37.3, 42.9, 59.5, 71.8, 85.3, 98.8, 106.7, 128.5])
vitesse3=np.array([1.4, 2.12, 2.57, 3.0, 3.4, 4.2, 4.8, 5.2, 5.6, 6.3, 7.2, 7.5, 8.6, 9.1, 9.4, 9.9])
m3=np.array([1.6, 3.2, 5.0, 7.1, 10.2, 15.7, 20.3, 24.0, 31.0, 41.8, 53.5, 64.0, 78.1, 90.7, 102.1, 127.0])

g=9.8
poids0 = ((m3)*10**(-3))*g
c = vitesse3**2 # Valeurs en abscisse : il faut un array (tableau) de numpy pour pouvoir ensuite tracer la droite de régression linéaire
alpha = poids0 # Valeurs en ordonnée

# Régression linéaire
a,b = np.polyfit(c,alpha,1) # détermination des coefficients du polynôme de régression

print("Modèle alpha = a*c + b")
print("a =",a )
print("b =", b )

# Tracé des graphes
plt.figure(1)
plt.plot(c,alpha,"o",color="b",label ="Données",markersize=5) # Graphe des données expérimentales
plt.plot(c,a*c+b,color="r",label="Régression linéaire") # Droite de régression linéaire

# Axes, titre et légendes
plt.xlabel("vitesse**2")
plt.ylabel("poids")
plt.title("poids en fct de vitesse**2")
plt.legend()
plt.grid()
plt.show()
```

Code python 3

Recherche de la trainée :

```
##2eme series d'experience : Pour la trainée
## calcul force T
#d1 = 0.0705
#d2 = 0.1495

d1 = 0.0905
d2 = 0.1495

F0 = 0.0983069672249258
X=[]
F=(0.0993116379715502,0.102325650211424,0.110865351557732,0.114381699170917,0.116893376037478,0.12423735890538,0.1290456431116909,0.133972778730094,0.134977449476719,0.139498467836529,0.141005473956466,0.14201014470309,0.145024156942964,0.147033498436213,0.147033498436213,0.148038169182837,0.150549846049398,0.151554516796023,0.154066193662584,0.153061522915959,0.156075535155833)

for i in range(len(F)) :
    T=(d1*(F[i]-F0)/d2)
    X.append(T)

print(X)
```

Code python 4

Tracé de Cx en fonction de angle d'incidence :

Tracé de Cx en fonction de Cz → polaires des vitesses :

```
## tracé de Cx en fonction de l'angle d'incidence

angle_incidence=np.array([2.8,4.8,7.8,8.5,8.5,8.8,8.9,10.5,8.5,8.6,9,8.7,9,9.6,9.9,9.5,10,9,10.5,9,9])

plt.close('all')
plt.figure('titre')
plt.plot(angle_incidence,Cx,'+ ')
plt.legend()
plt.plot()
plt.xlabel('angle d incidence')
plt.ylabel('Cx')
plt.show()

## tracé de Cx en fonction de Cz

Cx=np.array([0.0069522,0.00800967,0.00830945 ,0.01050723 , 0.01480834,0.027, 0.03862334, 0.0438564, 0.04630166, 0.06527692, 0.06005605,0.06177002, 0.05983239, 0.05924879,
0.05844608, 0.05817882, 0.0547573, 0.05269348, 0.05083798])

Cz=np.array([0.17,0.35,0.036,0.4,0.5,0.61,0.6,0.61,0.63,0.42,0.46,0.52,0.58,0.60,0.670,0.57,0.53,0.63,0.64])

plt.close('all')
plt.figure('titre')
plt.plot(Cx,Cz,'+ ')
plt.legend()
plt.plot()
plt.xlabel('Cx')
plt.ylabel('Cz')
plt.show()
##
```