Récapitulatif de mes valeurs

Attention : démonter le galet afin de limiter les frottements et mettre en garde les élèves sur la possibilité pour la lisse de taper si elle part à plus de 90°

|  |  |  |
| --- | --- | --- |
| Elément à modéliser | Forme de la formule | Constantes à déterminer |
| Couple de gravité |  |  |
| Couple ressort |  |  |
| Inertie de la lisse |  |  |
| Frottements |  |  |

Attention : il faut encore affiner

Dans le modèle SW fourni, il suffit pour faire varier X dans l’onglet « Equations » en cliquant droit sur le paramètre « X=… » dans l’arbre, puis éditer les équations, et changer la valeur…

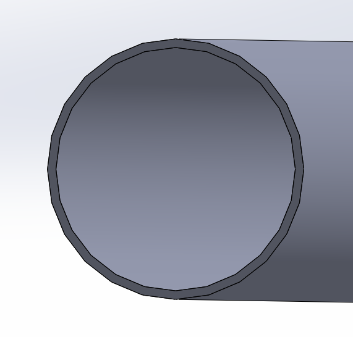
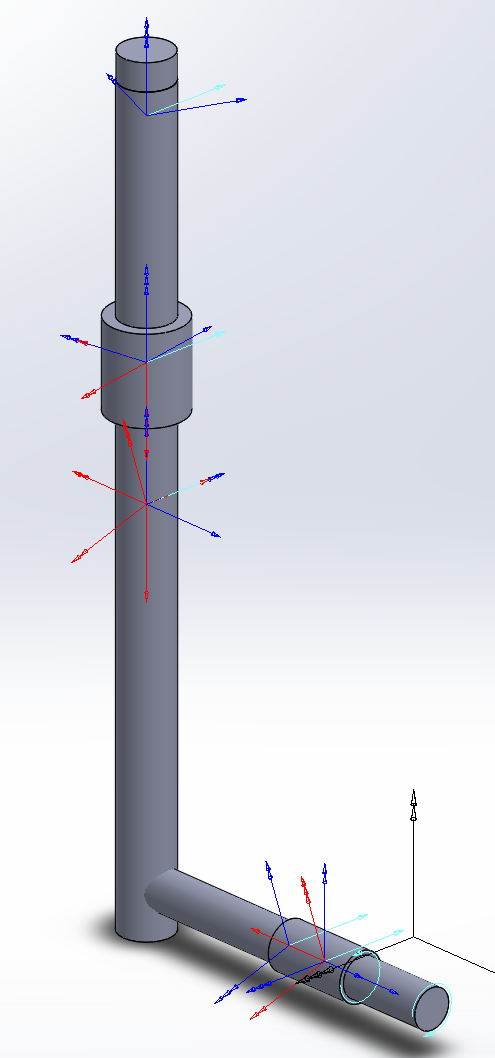
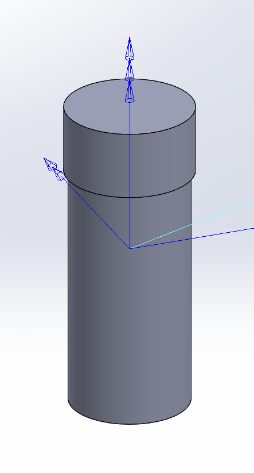


|  |  |
| --- | --- |
| Calage angulaire répondant au CDC |  |

Remarque : à partir de , la barrière remonte

Schéma de définition des variables

Dimensions en mm :



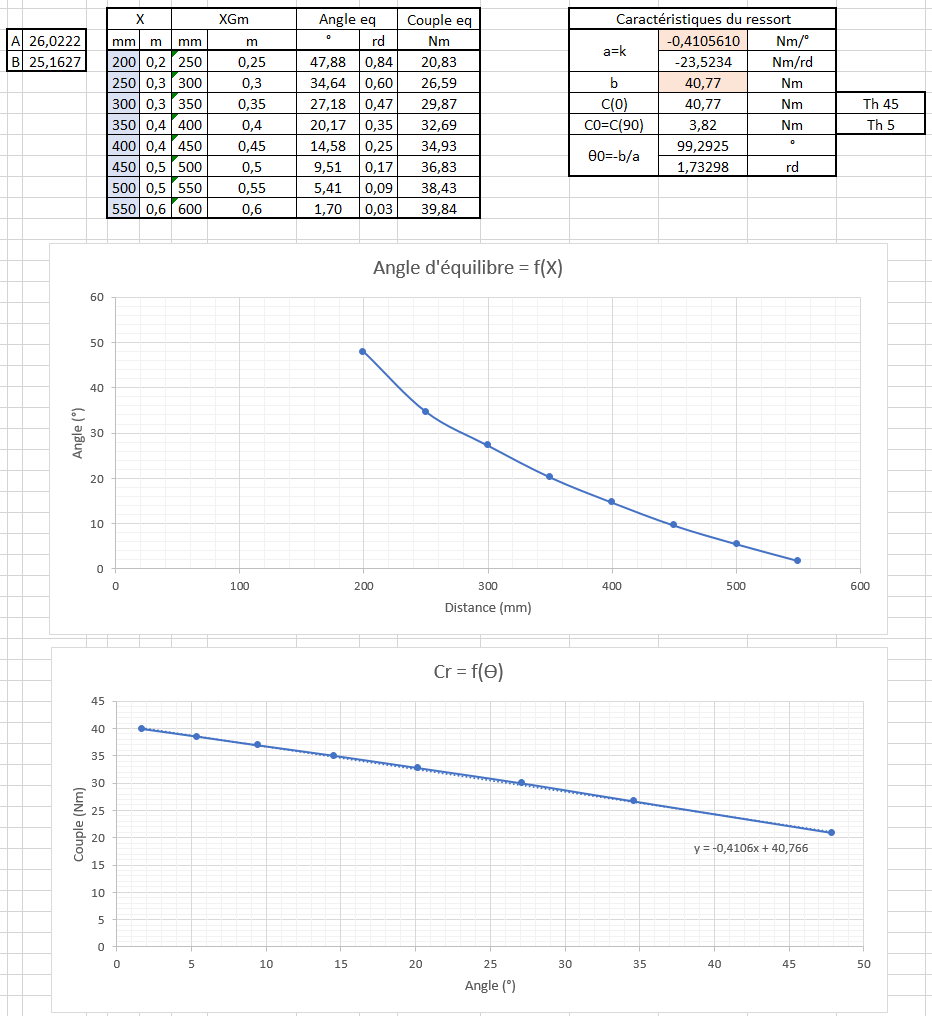
Détermination du couple de gravité

On pourrait valider ces coefficients avec SW en comparant par exemple l’angle d’équilibre moyen de la lisse réel à l’angle obtenu en simulation, je ne l’ai pas fait ni demandé dans le sujet.

Sinon, passer par la résultante de l’action de la gravité et la position du centre de gravité.

Identification de la formule du couple du ressort

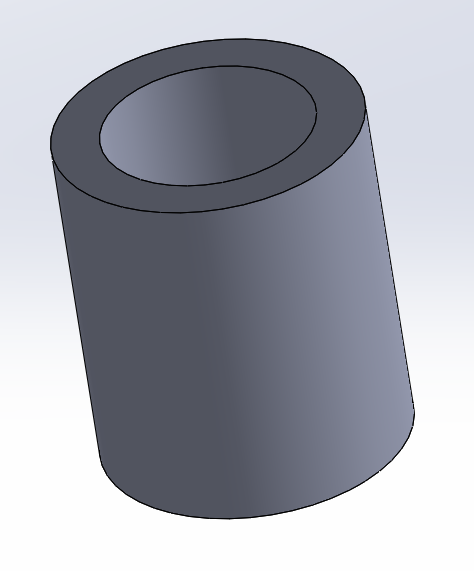
J’ai réalisé des expériences sur 10 secondes. J’ai essayé de faire tenir la lisse en équilibre pour différentes positions de la masse en me plaçant aux extrémités de la zone d’équilibre. J’ai alors pris comme angle d’équilibre le milieu entre les deux valeurs extrêmes. Tout est dans l’Excel fourni.



On considère que le ressort agit sur la barrière en suivant une loi linéaire du type :

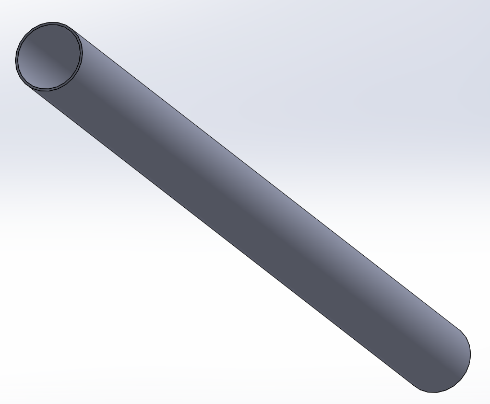
On définit comme le couple lorsque la barrière est verticale.

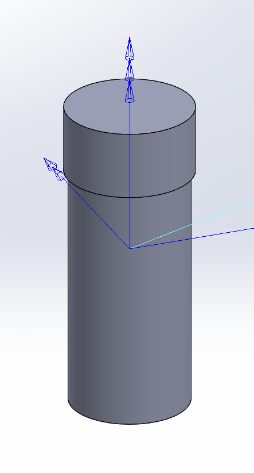
|  |  |
| --- | --- |
| Expérimenté | Constructeur |
|  |  |

Calcul de l’inertie

Masse mobile de révolution sur (longueur 100 mm):

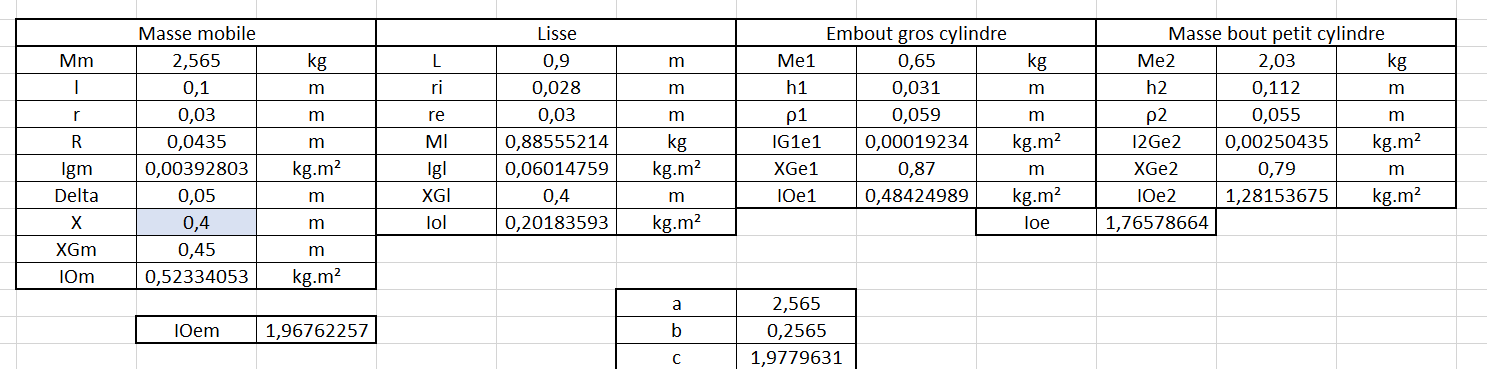
Avec les données :

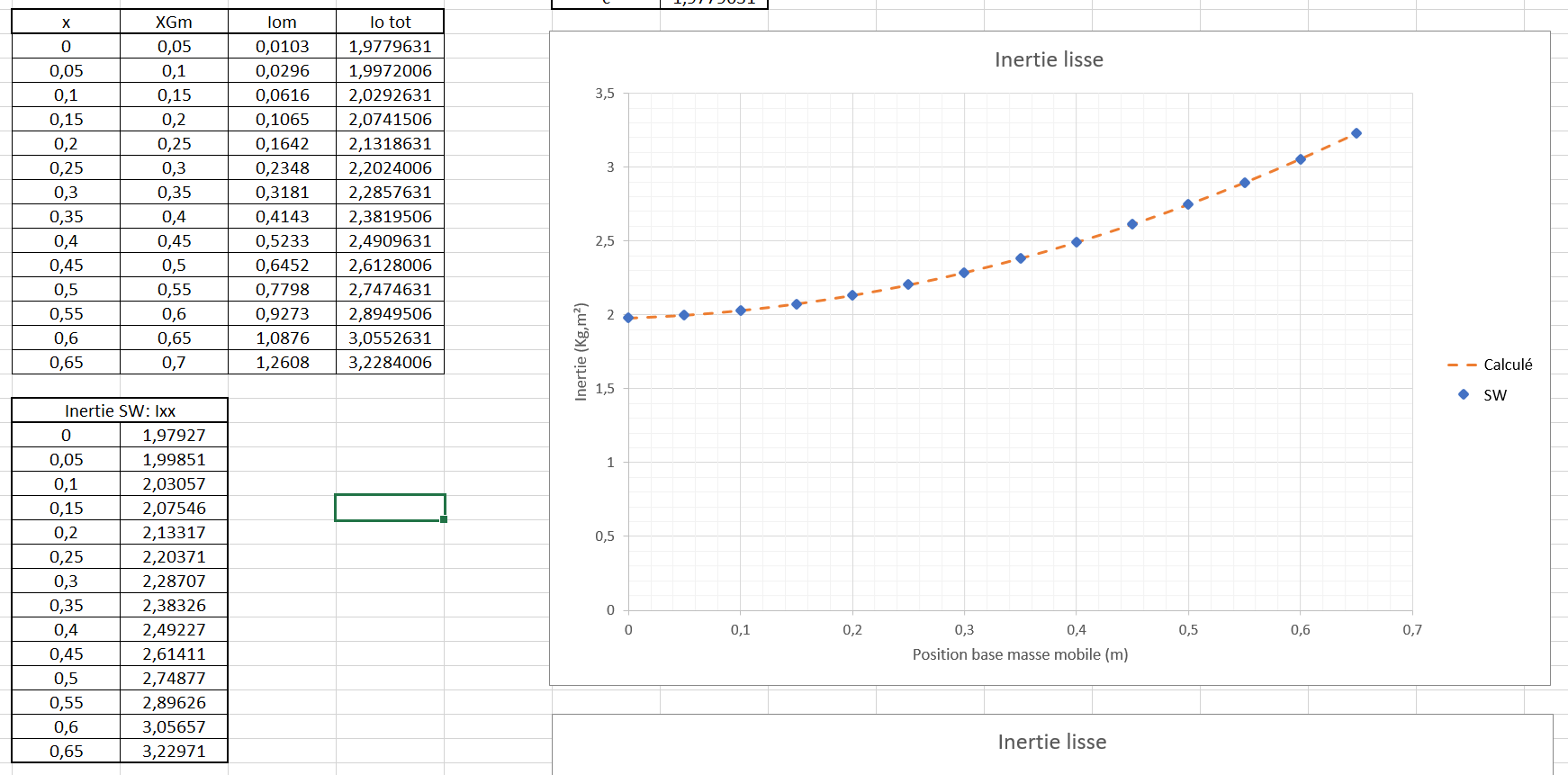
Lisse de révolution sur :

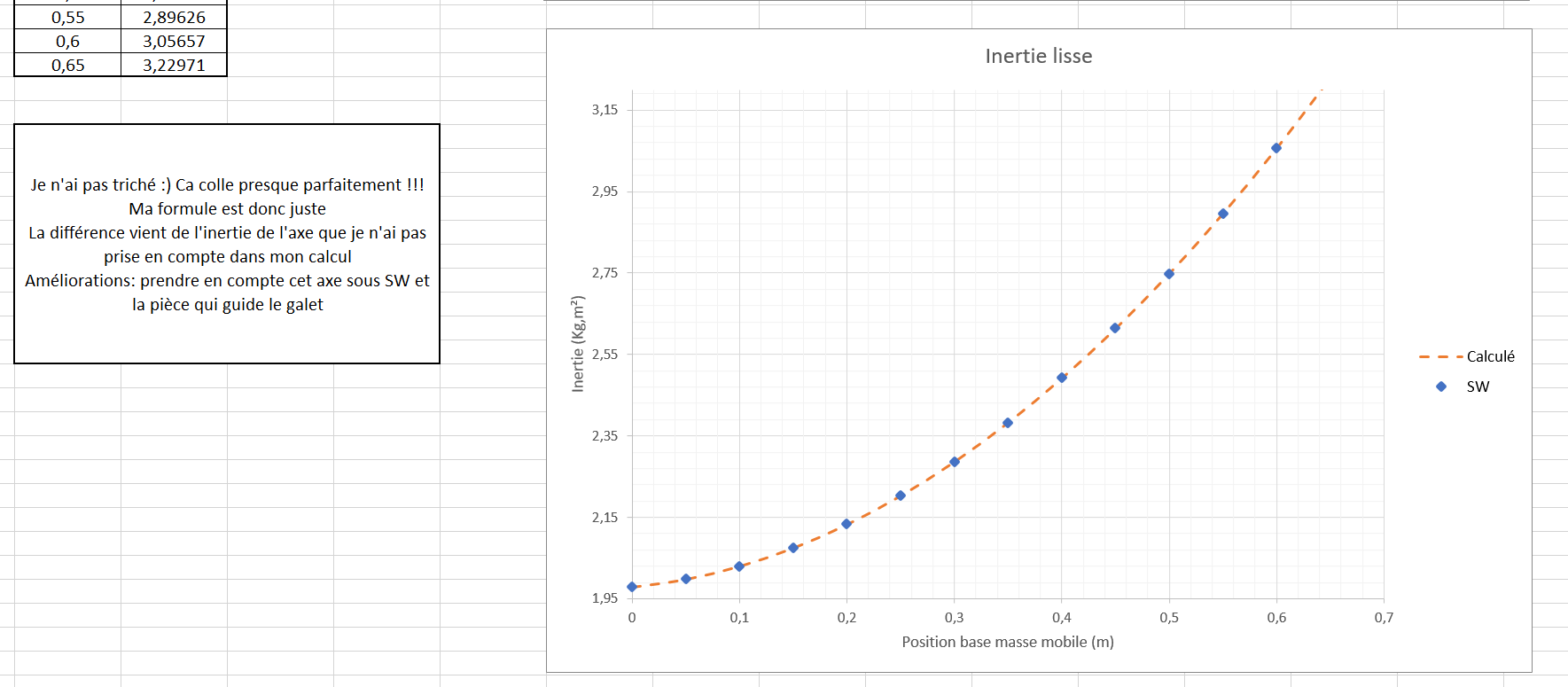
Gros cylindre poids fixe :

Petit cylindre poids fixe :

On néglige inertie axe + pièce de guidage du galet



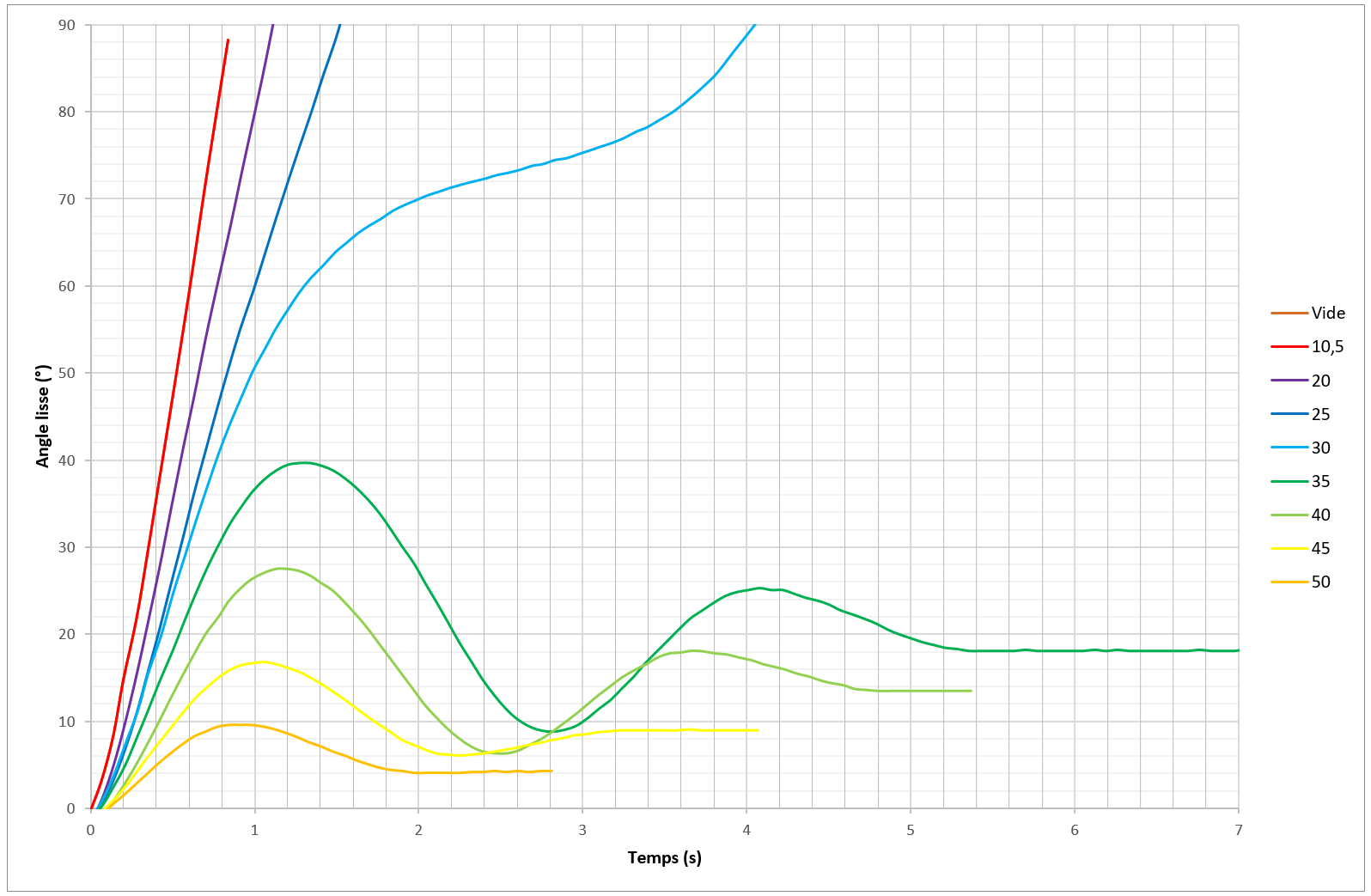




L’inertie calculée et l’inertie obtenue avec SW sont parfaitement les mêmes, à la différence près que sous SW, j’ai l’inertie de l’axe en plus, que je n’ai pas prise en compte dans mon calcul.

Tracé des réponses expérimentales de la lisse

Après avoir fait évoluer la lisse pour différentes positions de la masse, j’ai exporté les mesures sur Excel et recalé chaque courbe puis tracé :



Inversion pour résolution numérique

Identification des frottements secs

En exploitant mes mesures initiales d’équilibre qui balayaient la zone d’équilibre, j’ai calculé la somme des couples qui en théorie n’est nulle qu’à l’équilibre.

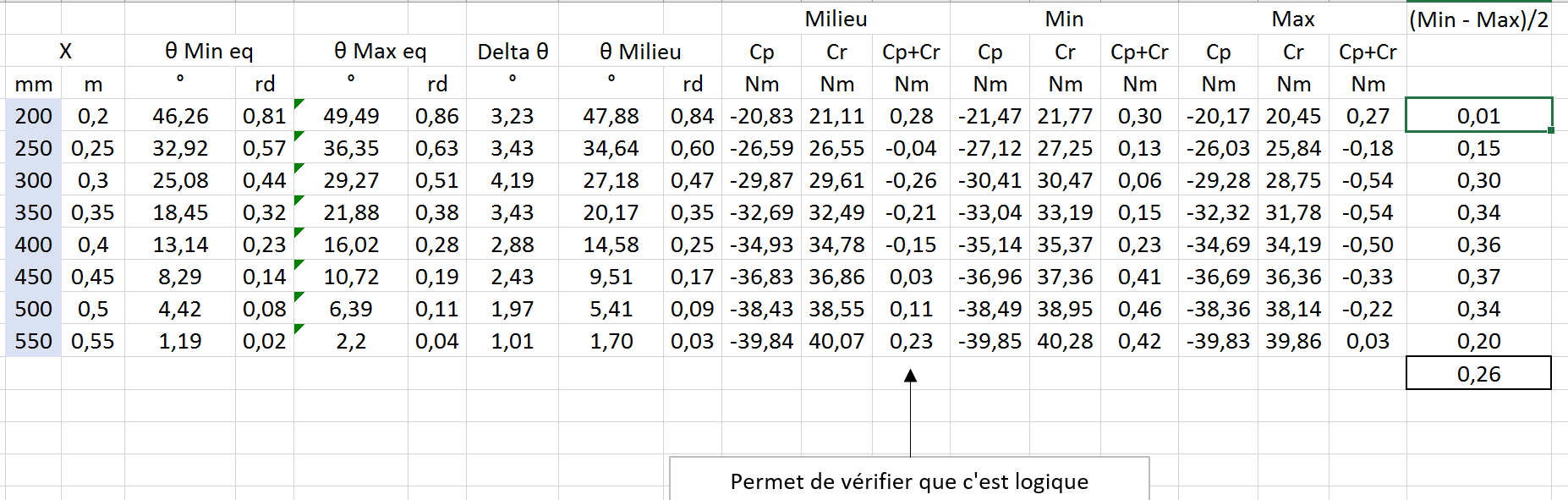
J’ai donc calcul cette somme sur les deux extrémités de l’intervalle d’équilibre pour différentes positions de la masse.

J’espérais avoir un couple positif, l’autre négatif, mais en fait non et c’est normal.

En effet, est une approximation obtenue à l’aide de mesures. Et comme est assez petit, les erreurs se cumulent.

Finalement, j’ai choisi de mesurer la différence des couples en position d’équilibre basse et haute et de diviser cette plage par 2.

J’ai ensuite fait la moyenne des valeurs pour chaque position de masse :



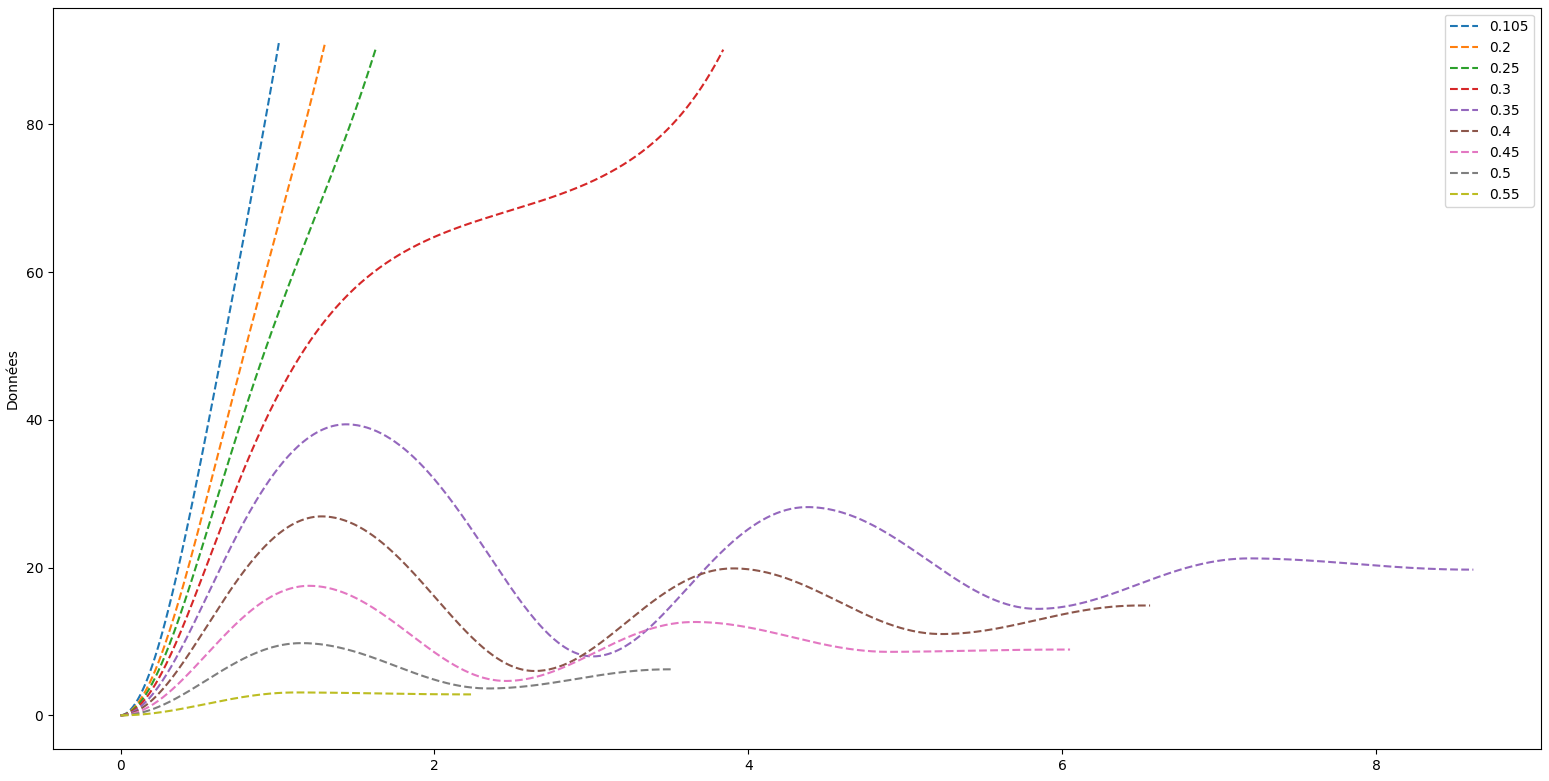
Du fait des explications précédentes, une légère incertitude plane sur ce coefficient.

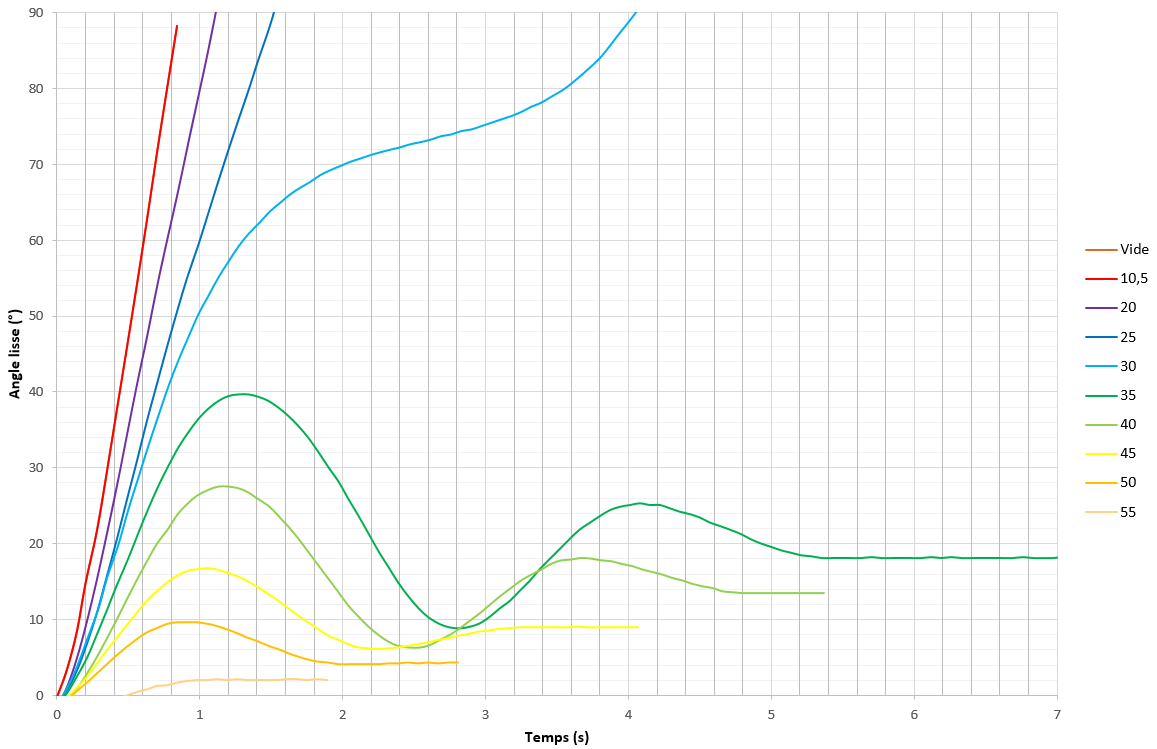
Résolution Python & Frottements fluides

La résolution a lieu tant que l’angle ne dépasse pas 90°, que le temps ne dépasse pas 10 secondes, et surtout, il a fallu trouver un critère d’arrêt lié aux frottements secs. En effet, quand la lisse s’arrête dans la réalité, sous python, le couple de frottement sec opposé au mouvement crée des oscillations. J’étudie les changements de signe de la vitesse, dès qu’il y en a 3 d’affilée, on arrête.

En recalant à tâtons le coefficient de frottements visqueux, je trouve :

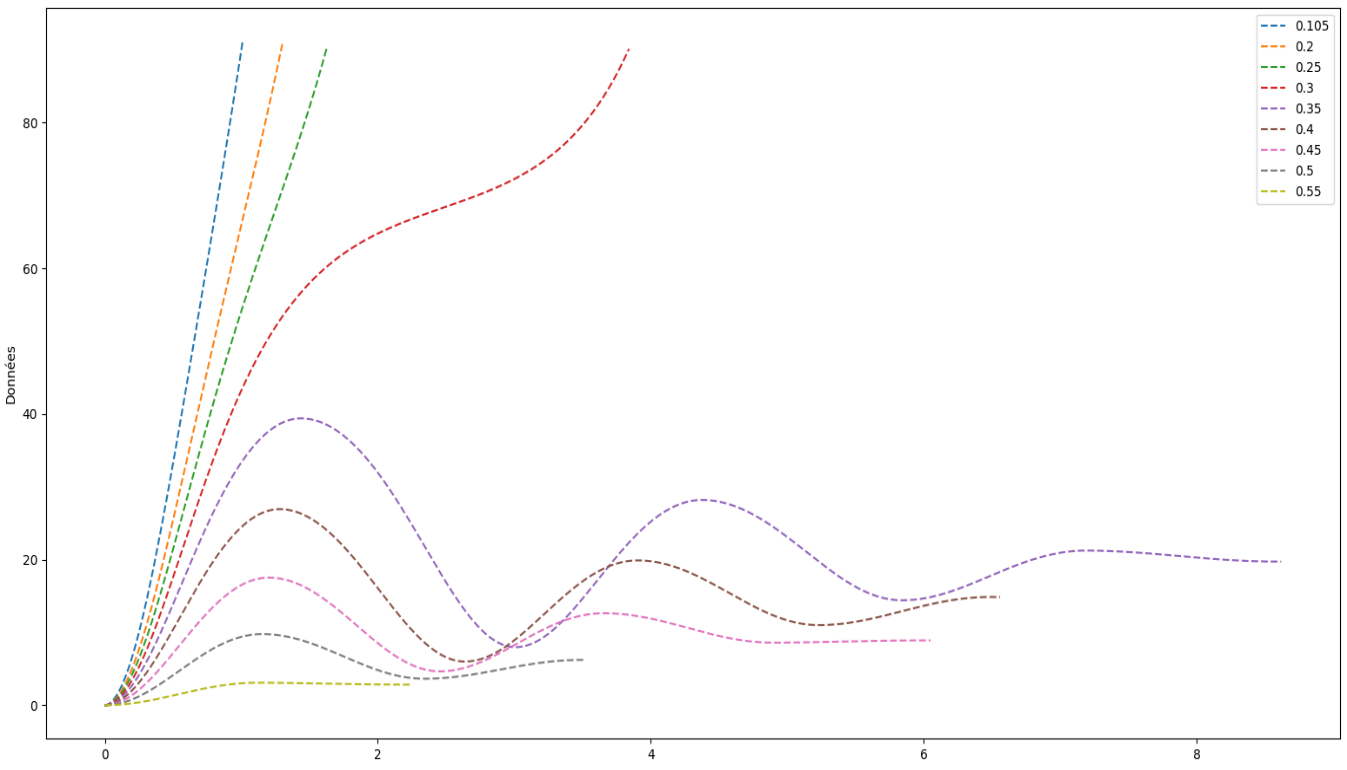
Précisons que ce coefficient a une grande influence sur le comportement transitoire observé pour la masse à 35 cm. J’ai donc réglé ce coefficient pour qu’à 35 cm sous Python, on observe le même comportement qu’en réel.

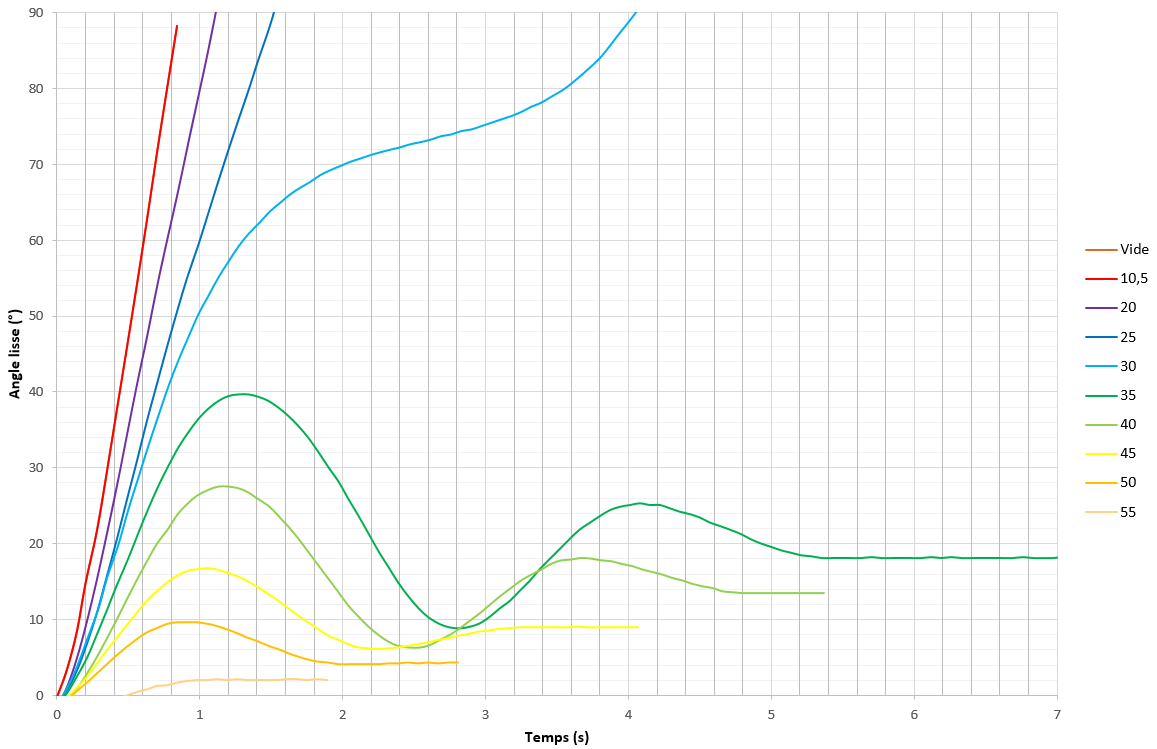




Ça colle pas trop mal ☺

Je ne suis pas exactement parti de 0 à chaque fois… Dommage

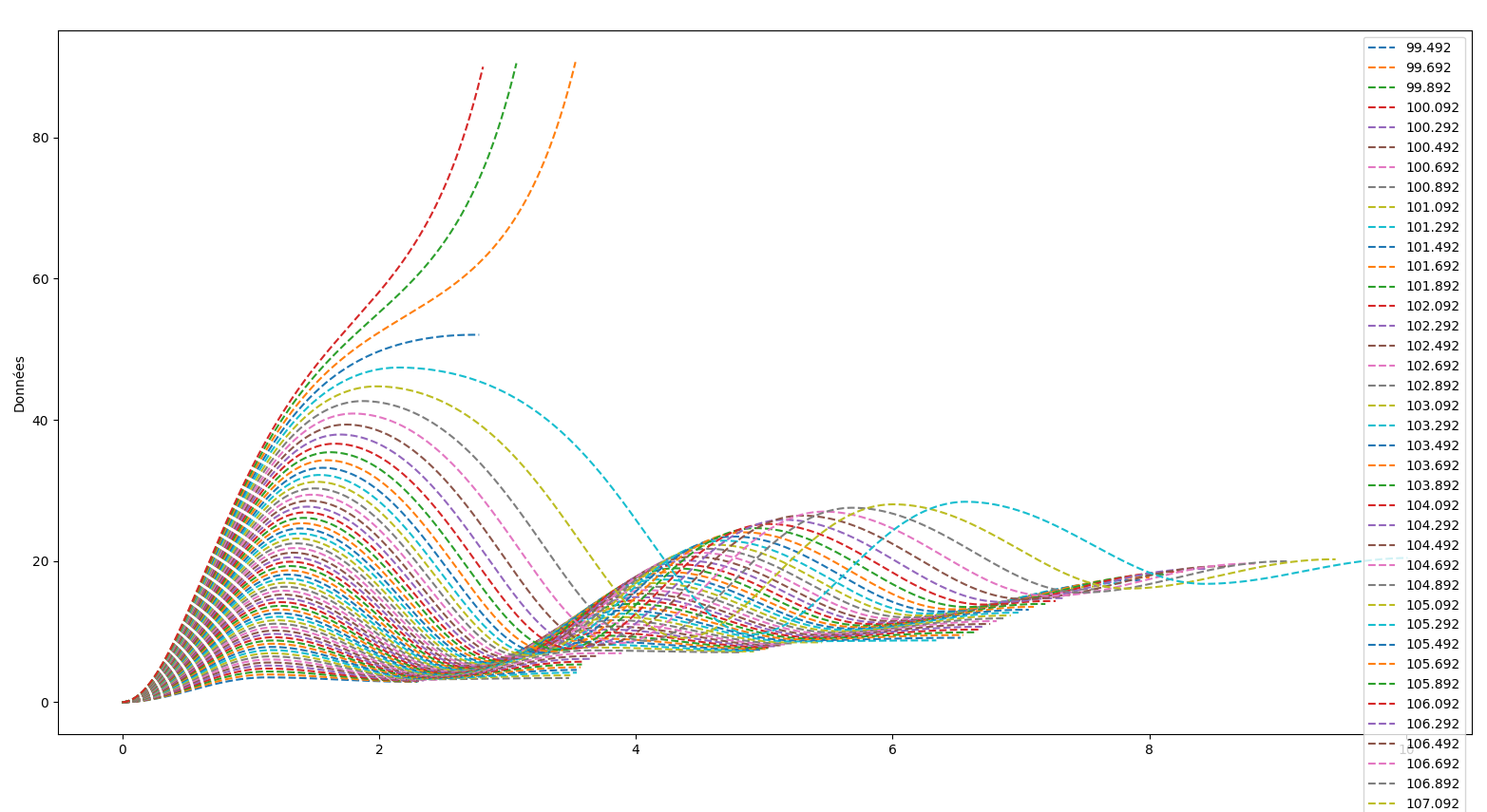
En les mettant l’une sur l’autre par transparence :



Réponse à la problématique

On ajoute une boucle while au code qui permet de récupérer, pour différents calages incrémentés depuis le calage actuel, le temps de fin et l’angle de fin.

Il faut à la fois que l’angle dépasse 90 et que le temps soit inférieur au critère de 3 secondes.



Remarque : à partir de , la barrière remonte

Le calage répondant au cahier des charges est :

Pour avoir cette précision, l’incrément d’angle est de

Comportement final de la lisse :

