



GYMNASE AUGUSTE PICCARD

# SUR LA ROTATION DES SOLIDES

OPTION SPÉCIFIQUE PHYSIQUE, 3M8

---

Julien Bricka, Romain Blondel

19 mars 2024

## BUT

Cette expérience vise à étudier la rotation d'un solide, en faisant varier différents paramètres influant le moment de force le faisant tourner et étudier en quoi son mouvement est impacté.

## 1 INTRODUCTION

Le mouvement de rotation est l'un des deux mouvements fondamentaux dans l'étude des mouvements en physique (le second étant le mouvement rectiligne). Le moment d'inertie est une mesure de la résistance d'un objet au mouvement de rotation. En effet, la cinématique de rotation s'applique lorsque toutes parties d'un corps tournent autour d'un axe, mais nous ne nous concentrerons que sur celle effectuant une rotation angulaire constante ou uniformément accélérée autour d'un axe fixe. Un objet en rotation possède une inertie déterminée par la forme de celui-ci.

Ce phénomène peut se retrouver dans la physique d'un yoyo, d'une toupie à fil. Sans inertie, le yoyo tomberait à la même vitesse que s'il était lâché (sans le support du fil) ; tirer le fil de la toupie ne demanderait quasiment aucune énergie. Dans un cadre plus commun, la notion d'inertie est utile dans la conception de pneus, car cela déterminera la difficulté pour le mettre en mouvement ou alors l'arrêter.

Avant Galilée, la théorie du mouvement ne connaissait pas l'inertie : l'état d'un corps est immobile dans son "lieu naturel" et son "mouvement naturel" est de retourner dans son milieu. Ainsi, les corps lourds tel que l'eau et la terre devait tomber, et les corps légers comme l'air ou le feu devait monter. Les anciennes idées de conservation du mouvement d'Aristote seront reprises, celles-ci expliquait qu'en projetant un projectile, ce dernier peut conserver en lui de l'élan. Newton s'inspire des écrits de Galilée et Descartes afin de proposer sa vision de l'inertie, qui est sa première loi : tout corps persévère dans l'état de repos ou de mouvement uniforme en ligne droite dans lequel il se trouve, à moins que quelque force n'agisse sur lui, et ne le contraigne à changer d'état. Huygens va définir le moment d'inertie qui s'applique similairement aux objets en rotations<sup>1</sup>.

L'inertie découle de la deuxième loi de Newton :

$$\sum \vec{F} = m\vec{a}$$

Dans le cas où un objet tourne autour d'un axe, une formule similaire est utilisée :

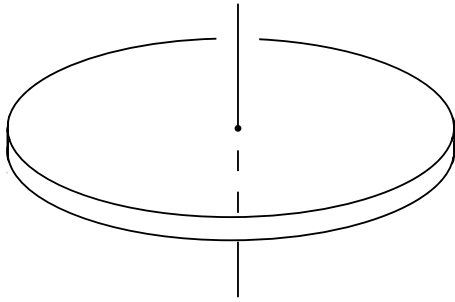
$$M = I\alpha$$

avec  $M [N \cdot m]$  le moment de force,  $\alpha [rad(1/s^2)]$  l'accélération angulaire et  $I [kg \cdot m^2]$  le moment d'inertie. (Notons que cette formule est un cas particulier que le solide tourne autour d'un axe, mais une expression générale existe également avec un tenseur d'inertie.)

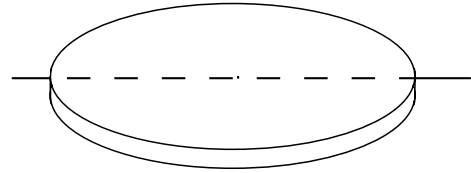
Chaque objet possède une inertie différente dépendant de sa masse, sa taille et le placement de l'axe auquel il tourne autour. Lors de l'expérience, nous utilisons un "disque" (en soit un empilement de disques) placé sur un coussin d'air, équivalant à un axe de rotation vertical.

---

1. Source : <https://fr.wikipedia.org/wiki/Inertie>

**FIGURE 1**

Un disque ayant un axe de rotation vertical  
a une inertie valant  $I = \frac{r^2}{2}$

**FIGURE 2**

Un disque ayant un axe de rotation hori-  
zontal a une inertie valant  $I = \frac{r^2}{4}$

Un point placé sur un disque en rotation peut être caractérisé par une trajectoire circulaire accéléré, ou non. Dans un mouvement circulaire uniforme (*MCU*), un point matériel à une position  $\theta_0$  [rad] de sa référence, subie une vitesse angulaire constante  $\omega_0$  [ $\frac{rad}{s}$ ] autour d'un axe :

$$\theta(t) = \omega_0 t + \theta_0$$

et dans le cas d'un mouvement uniformément accéléré (*MCUA*), nous prenons compte d'un  $\alpha$ , représentant la constante d'accélération angulaire du système :

$$\theta(t) = \frac{1}{2}\alpha t^2 + \omega_0 t + \theta_0$$

## 2 PRINCIPE DE MESURE ET DESCRIPTION

Nous allons enregistrer des vidéos d'un système muni de point de tracking tournant sous l'application d'une force constante, en faisant varier celle-ci et sa distance d'application. Les vidéos seront ensuite analysées afin d'obtenir l'accélération angulaire du système et observer le lien entre celle-ci et le moment de la force qui s'exerce sur le système.

### 2.1 MATÉRIEL

Le matériel suivant est nécessaire afin de réaliser l'expérience :

- |   |                          |
|---|--------------------------|
| — Une soufflerie                                | — Du scotch              |
| — Un tube                                       | — Un stylo               |
| — Un grand disque métallique                    | — Du papier              |
| — 3 petits disques (diamètre de 3, 6 et 9 [cm]) | — Un téléphone ou caméra |
| — Une poulie                                    | — Un statif              |
| — Une ficelle                                   | — Une plaque métallique  |
| — Des poids (entre 20 et 90 [g])                | — Une noix de serrage    |
| — Un support de poids (10 [g])                  |                          |

## 2.2 DÉROULEMENT

Il est nécessaire d'assembler le dispositif d'expérimentation (voir figure 3) avant les manœuvres suivantes. Il consiste en le grand disque métallique avec dessus les trois petits, autour desquels on peut entourer la ficelle, le tout monté sur un axe. On peut y scotcher un bout de papier avec un point noir au centre et un autre sur le bord afin de faire le tracking. On le met alors sur la soufflerie afin de réduire au maximum le frottement. De plus, on fixe une plaque métallique sur un statif afin de tenir le téléphone. Finalement, on aligne la poulie avec le bord des petits disque et on peut la fixer par exemple avec du scotch.

Avant de commencer l'expérience, le téléphone doit être placé sur la planche métallique. L'entièreté du grand disque doit être incluse dans le cadrage et la soufflerie lancée (voir figure 4).

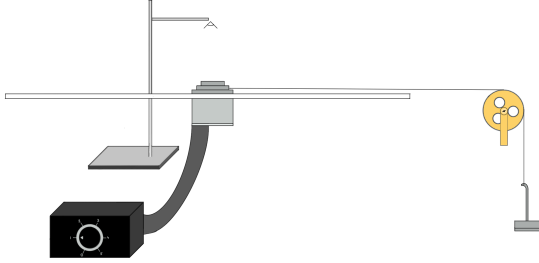
### 2.2.1 EXPÉRIENCE 1

1. Lier une extrémité de la ficelle aux poids (30 [g] en plus du support) et faire passer celui-ci par la poulie.
2. Enrouler l'autre extrémité de la ficelle autour l'un des trois petits disques.
3. Commencer l'enregistrement de la vidéo.
4. Donner un léger élan au grand disque pour enrouler le fil autour du disque. (Cela permet d'enregistrer l'accélération complète créée par la masse.)
5. Une fois la masse descendue, arrêter l'enregistrement vidéo.
6. Faire de même avec 50, 70 et 90 [g].

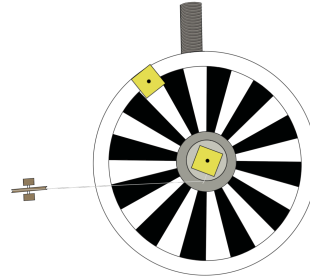
### 2.2.2 EXPÉRIENCE 2

1. Lier une extrémité de la ficelle aux poids (70 [g] avec le support) et faire passer celui-ci par la poulie.
2. Enrouler l'autre extrémité de la ficelle autour du disque ayant un diamètre de 3 [cm].
3. Commencer l'enregistrement de la vidéo.
4. Donner un léger élan au grand disque pour enrouler le fil autour du disque. (Cela permet d'enregistrer l'accélération complète créée par la masse.)
5. Une fois la masse descendue, arrêter l'enregistrement vidéo.
6. Faire de même avec les disques de diamètre de 6 et 9 [cm].

## 2.3 SCHÉMAS



**FIGURE 3**  
Schéma de du dispositif d'expérimentation



**FIGURE 4**  
Schéma des disques vus du dessus (point de vue de l'enregistrement vidéo)

## 3 RÉSULTATS ET CALCULS

Ayant toutes les vidéos prises pour les différentes masses et rayons, il a fallu effectuer le tracking du point qui tourne. Pour cela nous avons utilisé la librairie *Python* nommée *OpenCV*<sup>2</sup>.

*OpenCV* est la plus grande librairie d'analyse vidéo. C'est open source et contient plus de 2500 algorithmes. Elle est écrite en *C* et est supportée également en *C++*, *Java* et *Python*. On peut également noter un support de toute les plateformes principales (*Windows*, *Linux*, *MacOS*), les mobiles (*Android*, *iOS*) mais surtout les interfaces graphiques majeurs (*CUDA*, *OpenCL*).

Nous avons donc importé les vidéos avec cette librairie afin de les convertir en liste de matrices de pixels pour chaque image, qui peuvent être visualisée par *matplotlib* et manipulée comme des *array* de *numpy*. Nous avons pris aussi le nombre d'images par seconde (*[fps]*). Pour les trois mesures à 70 *[g]*, nous avons essayé de l'augmenté en prenant des vidéos via le ralenti d'un *iPhone*, qui est selon le constructeur du 120 *[fps]*, mais sont encodée ensuite en 30 *[fps]*, et pas de manière régulière (vidéo qui ralenti au fur et à mesure puis accélère à la fin dans un but esthétique) ce qui explique le grand étalement des vitesse. Toutes les autres mesures sont en 60 *[fps]* standards.

Nous avons ensuite sélectionné la partie de la vidéo que l'on désire analyser, et sur la première image de celle-ci nous avons repérer dans un carré de 50 pixels la position du centre et du point de tracking sur le disque. Nous avons ensuite enregistrer les positions sur chaque image du point via un algorithme *CSRT*<sup>3</sup>. Comme la position est enregistrée comme un rectangle de coins supérieur gauche (*a*, *b*) et de dimension  $w \times h$ , on peut le transformer en un point  $(x, y) = (\frac{x+(x+w)}{2}, \frac{y+(y+h)}{2})$ .

L'algorithme *CSRT* (*Channel and Spatial Reliability Tracker*) est un algorithme de tracking assez lent, mais très précis et robuste. Il s'appuie sur la couleur pour une meilleur stabilité, ainsi que sur l'évolution spatiale et temporel de l'objet. Il utilise un filtre de corrélation pour estimer la position

2. Site internet : [opencv.org](http://opencv.org)

3. Fonctionnement de CSRT : <https://medium.com/@khwabkalra1/object-tracking-2fe4127e58bf>

de l'objet, et ce filtre est entraîné et mis à jour itérativement durant le tracking.

Connaissant donc la position du centre de rotation  $(x_c, y_c)$  ainsi que la position du point de tracking à chaque image  $(x_f, y_f)$ , ainsi que le temps entre deux images  $\Delta t$  qui est l'inverse du nombre d'image par seconde, nous pouvons calculer la position angulaire du point de tracking par rapport au centre et sa position initiale au court du temps. Pour cela, nous utilisons la formule  $\theta_{f+1} = \theta_f + \arctan(\frac{y_f}{x_f}) - \arctan(\frac{y_{f-1}}{x_{f-1}})$  avec  $\theta_f$  la position angulaire à chaque frame et l'arctan est celle prenant en compte le cadran. Afin de corriger les saut de  $2\pi$  après un tour complet, nous avons corrigé les valeurs quand  $|\arctan(\frac{y_f}{x_f}) - \arctan(\frac{y_{f-1}}{x_{f-1}})|$  est supérieur à  $\pi$  (comme nos angles sont assez petit c'est une borne suffisante) par  $\theta_f := \theta_f - 2\pi \cdot \text{sgn}(\arctan(\frac{y_f}{x_f}) - \arctan(\frac{y_{f-1}}{x_{f-1}}))$ , avec  $\text{sgn}(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$ . Nous avons alors à ce stade les couples angles-temps  $((\theta_0, 0), (\theta_1, \Delta t), \dots, (\theta_f, f \cdot \Delta t), \dots)$  qui forment notre fonction. Nous avons donc fait une courbe de tendance quadratique ( $ax^2 + bx + c$ ) ensuite par la méthode des moindres carrés. Finalement, par la théorie du mouvement circulaire uniformément accéléré (MCUA), nous pouvons calculer l'accélération angulaire  $\alpha$  en doublant le coefficient pour  $x^2$  dans notre courbe de tendance.

Pour la vitesse angulaire, nous avons fait une dérivée numérique :  $\omega_f = \frac{\theta_{f+1} - \theta_f}{\Delta t}$ , puis fait une courbe de tendance linéaire ( $ax + b$ ) avec les couples vitesse-temps obtenus. Le coefficient pour  $x$  est par le MCUA l'accélération angulaire. Nous avons refait cela afin d'obtenir accélération angulaire instantanée, mais les résultats sont trop aléatoires pour être exploités. Les figures suivantes présentent les trois graphes  $(\theta(t), \omega(t) \text{ et } \alpha t, \text{ selon le temps } t)$  pour chaque mesure en précisant la masse et le rayon considéré (celui dit petit est de 1.5 [cm], moyen 3 [cm] et grand 4.5 [cm]). Les mesures sont en bleu et les courbes de tendances en vert. Leurs paramètres sont disponibles dans le *notebook* de calculs fourni en annexe.

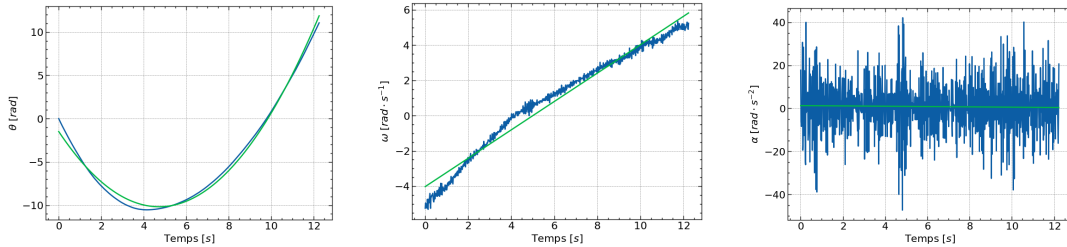


FIGURE 5

Angle, vitesse angulaire et accélération angulaire pour une masse de 40 [g] sur le rayon moyen

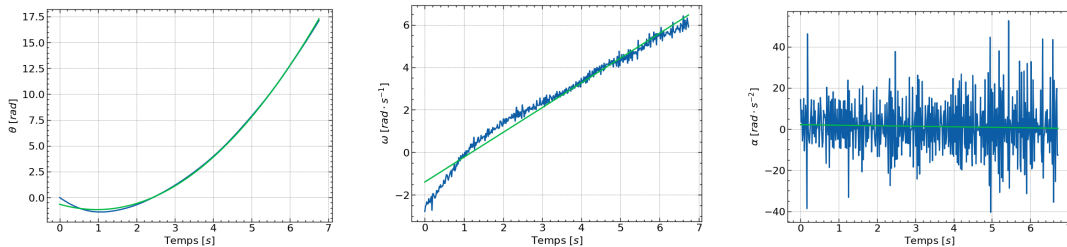
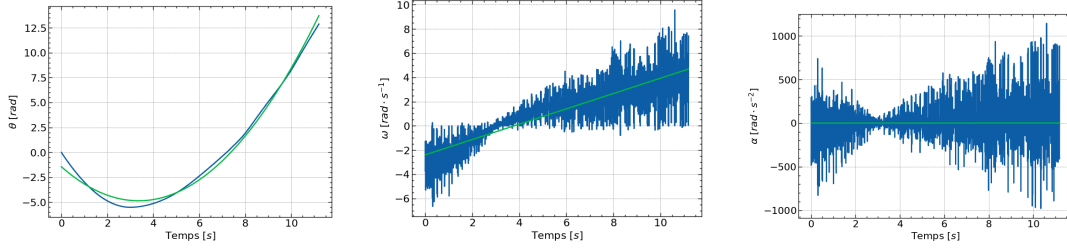


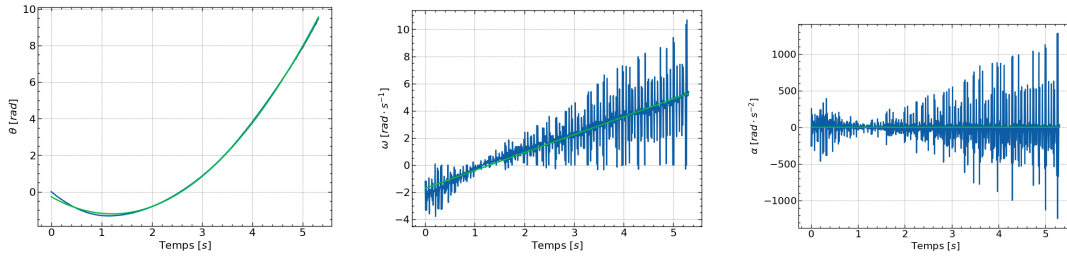
FIGURE 6

Angle, vitesse angulaire et accélération angulaire pour une masse de 60 [g] sur le rayon moyen



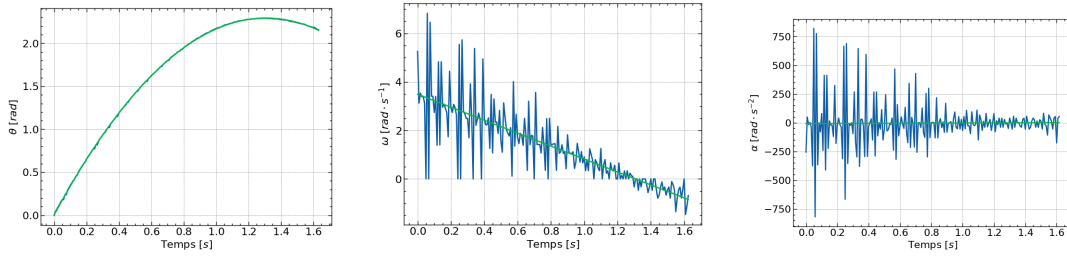
**FIGURE 7**

Angle, vitesse angulaire et accélération angulaire pour une masse de 70 [g] sur le rayon petit



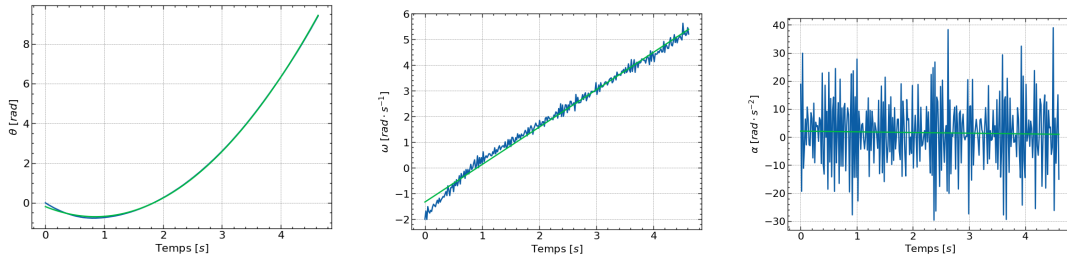
**FIGURE 8**

Angle, vitesse angulaire et accélération angulaire pour une masse de 70 [g] sur le rayon moyen



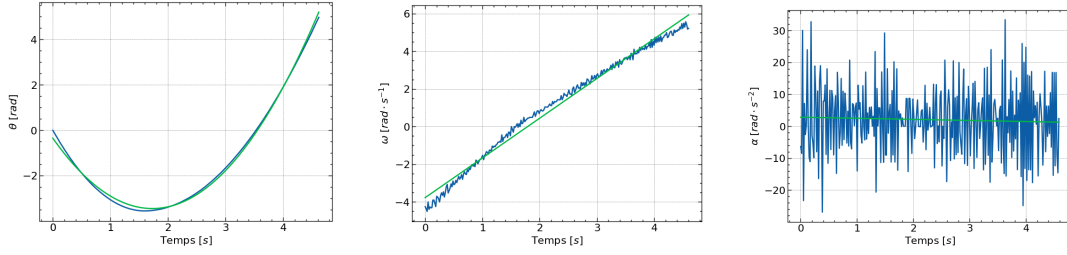
**FIGURE 9**

Angle, vitesse angulaire et accélération angulaire pour une masse de 70 [g] sur le rayon grand



**FIGURE 10**

Angle, vitesse angulaire et accélération angulaire pour une masse de 80 [g] sur le rayon moyen

**FIGURE 11**

Angle, vitesse angulaire et accélération angulaire pour une masse de 100 [g] sur le rayon moyen

Nous avons finalement consigné les principaux résultats dans la table suivante. Nous avons utilisé comme accélération angulaire la moyenne de celle obtenue par  $\theta$  et celle par  $\omega$  sans consigné outre mesure les valeurs particulières comme elles sont très proches (autour de 1 à 5 [%] d'écart, avec une autour de 10 [%]), mais elles sont également visibles dans le *notebook*. Le moment d'inertie du disque à été calculé dans chaque cas en considérant que la seule force  $F$  créant un moment est celle de pesanteur du contrepoids de masse  $m$  parfaitement redirigée par une poulie sans frottement et s'appliquant perpendiculairement au rayon. Pour le rayon  $r$ , on aura donc le moment  $M = F \cdot r = mgr$  avec  $g = 9.81 [m \cdot s^{-2}]$  d'où la relation  $M = I\alpha \Leftrightarrow I = \frac{M}{\alpha}$ .

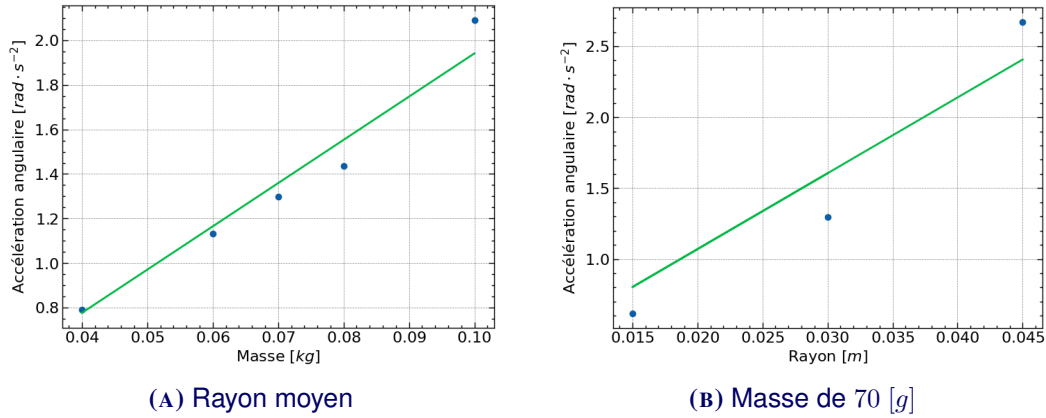
$m [kg]$	$r [m]$	$I [kg \cdot m^2]$	$\alpha [rad \cdot s^{-2}]$
0.040	0.030	0.015	0.791
0.060	0.030	0.016	1.133
0.070	0.015	0.017	0.618
0.070	0.030	0.016	1.297
0.070	0.045	0.012	2.673
0.080	0.030	0.016	1.435
0.100	0.030	0.014	2.092

**TABLE 1**

Récapitulatif des principaux résultats

Comme le disque est toujours le même avec le même axe de rotation, le moment d'inertie est censé être constant, ce qui se voit par une moyenne de  $I \approx 0.015 [kg \cdot m^2]$  avec un écart-type de 0.0016 qui correspond à moins de 11 [%]. Par les relations précédentes, nous voyons que  $\alpha$  dépend linéairement de la masse et du rayon car le moment de force l'est que  $I$  est constant, ce qui est illustré par la figure suivante.





**FIGURE 12**  
Accélération angulaire selon un paramètre fixé

## 4 DISCUSSION DES RÉSULTATS

Les résultats obtenus sont très satisfaisants. Comme on le voit en table 1, le facteur de proportionnalité est constant au cours des mesures ce qui nous permet d'affirmer que le moment d'inertie de notre disque autour de son axe centrale est autour de  $0.015 \text{ [kg} \cdot \text{m}^2]$  avec une bonne confiance. De plus, comme le moment de force est selon notre approximation seulement dépend de la force exercée par le contrepoids, il va augmenter seulement grâce à la masse (car la force pesante d'une masse est proportionnelle à celle-ci) et la distance entre le point d'application de la force et le pivot, *id est* le rayon du disque. Comme le moment de force est proportionnel à l'accélération angulaire selon la théorie est que il est lui même proportionnel à ces facteurs, l'accélération angulaire devrait l'être. C'est ce que montrent les graphiques en figure 12.

Dans cette figure, nous observons que la proportionnalité au rayon fonctionne bien, comme on le voit dans la corrélation entre la courbe de tendance en vert et les points de résultats en bleus. Mais pour la masse fixe cela semble plus compliqué. Le problème ne vient néanmoins pas de la loi physique mais de la prise de mesure. Observons en effet les vidéos des figures 7, 8 et 9 qui sont utilisées pour vérifier cette partie de la loi. Les graphes les plus intéressants sont pour la vitesse. En effet, nous voyons une grande divergence aux extrêmes, mais quand la vitesse change de direction et donc que le disque ralentit et repart dans l'autre sens, cela est moindre. Ce problème se voit également sur l'accélération, où dans les autres mesures elle est régulière dans le bruit mais pas dans celles-ci. Cela s'explique néanmoins assez simplement. Afin d'essayer d'avoir des mesures plus précises, nous avons pris les vidéos via le mode ralenti d'un *iPhone*, ce qui nous offrait en théorie une prise d'image à  $120 \text{ [fps]}$ , en théorie donc meilleure que les  $60 \text{ [fps]}$  des autres vidéos. Mais lors du traitement de celles-ci, un autre problème fut soulevé. En effet, le software *iOS* ne fait pas un rendu linéaire de cette vidéo juste en réduisant le nombre d'image par seconde à l'encodage. Il va faire un rendu non-linéaire à  $30 \text{ [fps]}$ , afin d'avoir un aspect esthétique de décélération au début de la vidéo jusqu'au ralenti puis accélération à nouveau à la fin. De fait, l'utilisation naïve de ces vidéos comme du  $120 \text{ [fps]}$  sans corriger ces effets car cela est trop coûteux pour le bénéfice, ce qui explique les graphes étranges pour ces trois mesures. On voit que malgré cela, les résultats demeurent cohérents avec les autres. Une autre explication peut être que le plus grand nombre d'image par seconde fait que la précision de la méthode de tracking, ou dans ce cas son imprécision, est plus marquante car les résultats demandent dans ces cas plus de précision pour quantifier la vitesse précisément. Cette hypothèse semble plus probable au vu du fait que les résultats sont cohérents au globale avec les autres ce qui laisse à penser que l'imprécision des mesures instantanées ne se répercutent pas globalement.

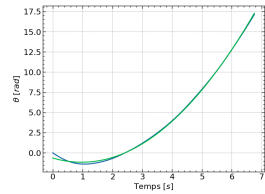
Il y a finalement un dernier point majeur à discuter, qui est visible principalement sur les graphes de la vitesse, mais explique également le léger décalage entre la courbe de tendance parabolique et l'angle en fonction du temps. En effet, la vitesse semble en deux parties, une avant qu'elle soit nulle où elle va diminuer plus rapidement qu'elle accélère ensuite. En théorie, si la seule force exerçant un moment était la force pesante  $F_p$ , cela ne devrait être qu'une droite. La séparation en deux cas s'explique néanmoins si l'on inclut une force de frottement  $F_f$  s'opposant au mouvement par définition. De fait, quand on lance le disque dans l'autre direction que celle où tire le contrepoids,  $F_p$  et  $F_f$  vont donc s'exercer les deux contre le mouvement et le moment de force va donc en réalité plus dépendre de  $F_{tot} = F_p + F_f$  alors que quand le disque s'arrête et repart dans la direction du contrepoids, les deux forces seront opposées et la force totale sera plutôt  $F_{tot} = F_p - F_f$ , ce qui explique les deux régimes de vitesse et donc de position. Néanmoins, les traités comme un seul régime pour les courbes de tendances n'est pas dérangeant pour observer les phénomènes désirés.

Un graphique peut encore interpeller, la figure 9. En effet, la courbe et la vitesse ont un signe opposé à toutes les autres. Cela s'explique simplement par un point de départ pour les angles différents qui fait qu'avec la méthode de calcul il y a eu ce changement de signe. On a donc pris dans la suite des calculs l'accélération en valeur absolue.

## 5 CONCLUSION

Les résultats sont très satisfaisants et confirment bien la théorie de rotation des solides. Nous avons considérés un des modèles les plus simples possibles, ce qui a mené à des erreurs tels que la non-prise en compte des forces de frottement, mais qui a permis quand même de prendre des mesures d'assez bonne qualité. L'essai de prise au ralenti à néanmoins fait qu'une partie des mesures étaient dégradées mais quand même exploitables et donnant des résultats au final cohérents. L'algorithme de tracking fut de plus bien choisi, comme en témoigne les bonnes courbes de mesures ayant pu être obtenues sans intervention de notre part dans le processus de tracking. On pourrait améliorer l'expérience en prenant les frottements en compte ou en essayant de les éliminer, par exemple avec un meilleur dispositif de coussin d'air ou en essayant de faire par un système magnétique, voir même avec un supraconducteur pour enlever tout frottement. On pourrait aussi imaginer un tracking matériel plutôt que par image qui serait plus précise. Finalement, une méthode de variation des paramètres plus large pourrait offrir encore plus de données. Malgré cela, cette expérience est très bonne pour confirmer la validité du modèle.





```

% omega = []

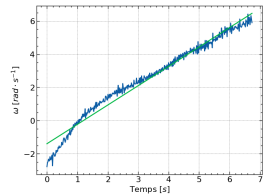
for i = 1:nrangs(len(theta)-1);
    omega.append((theta[i+1]-theta[i])/frame_rate))
end

% time_2 = np.arange(len(omega))/frame_rate
% point_2 = _np.optimize_curve_fit(lin, time_2, omega)
% point_2

% array([ 1.16055327, -1.39179803])

% plt.plot(time_2, omega)
% plt.plot(time_2, lin(time_2, *point_2))
% plt.xlabel("omega")
% plt.ylabel("t")
% plt.savefig("omega_0.png")

```

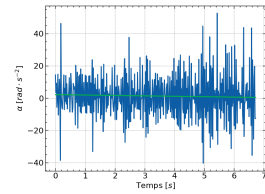


```
[3]: alpha = []

for i in range(len(omega)):
    alpha.append((omega[i]+1)*omega[i]/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = mp.optimize_curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel('Time [s]')
plt.ylabel('F*lambda [Rad /codet a^(-2)]')
plt.savefig('m05_a.png')
```



```

3 acc_quad = prog1[i0]*2
4 print(acc_quad, acc_lin)
5 acc_lin+=acc_quad/2
6 s=0
7 s_600 = s + s_600 + s_3
8
9 print(s, s_600)
10
11 1.014342337778434 1.3650832741707E33
12 0.01064314181497874
13
14 1.3 Vídeo p70
15
16 video_path = "p70.avi"
17 cap = cv2.VideoCapture(video_path)
18
19 frame = cap.get(cv2.CAP_PROP_FPS)
20 print("Frame rate: ", frame_rate)
21
22
23 Frame rate: 30.0

```

[illegible]

```

results = []

from tracker import *
for i in range(1, len(track_frames)):
    # Get the next frame
    frame = track_frames[i]

    # Update the tracker
    success, res = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        x, y, w, h = tuple(map(int, res))
        results.append([x, y, w, h])
        cv.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        frame_tracker.append(frame)

def disp_img(frame):
    plt.imshow(frame_tracker[num])

# widget interface
display = widgets.interact(
    disp_img=widgets.interact(disp_img, num=widgets.IntSlider(
        value=0, min=0, max=len(track_frames)-1, step=1),
        Output(), _dom_classes=["widget-interact"]
    )
)

@function_widget._display_interact
def good_results = results

center_of_rot = (722, 530, 50, 50)

def get_pos_rot(center, frame):
    x, y, w, h = center
    return (x + w/2), (y + h/2)

pos_c = pos_rot(center, center_of_rot)
cur = frame_tracker[pos_c]

while cur:
    cur_image = cur

    prev_rot = [pos_c[0]-1, pos_c[1], pos_c[0]+1, pos_c[1]]
    for i in range(1, len(prev_rot)):
        next_rot = [prev_rot[i-1][0], pos_c[1], prev_rot[i-1][0]+1, pos_c[1]]
        cur_image = np.concatenate([cur_image, next_rot[i]]).astype(np.uint8)

```

[illegible][illegible][illegible]

-5.0878048160044	-0.5033327662747	-0.4878133601335	-0.0787462644211
-5.0795913274474	-0.5033327662747	-0.0507334442315	-0.0787462644211
-5.0713778000000	-0.5033327662747	-0.0226834442315	-0.0787462644211
-5.0631652744744	-0.5033327662747	-0.0070511864235	-0.0982781340676
-5.0549527447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-5.0467402744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-5.0385277447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-5.0303152744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-5.0221027447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-5.0138902744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-5.0056777447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9974652744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9892527447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9810402744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9728277447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9646152744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9564027447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9481902744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9399777447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9317652744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9235527447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9153402744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.9071277447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8989152744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8907027447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8824902744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8742777447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8660652744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8578527447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8496402744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8414277447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8332152744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8250027447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8167902744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8085777447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.8003652744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7921527447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7839402744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7757277447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7675152744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7593027447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7510902744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7428777447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7346652744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7264527447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7182402744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7100277447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.7018152744744	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.6936027447444	-0.5033327662747	-0.00134442315	-0.0982781340676
-4.685390274			

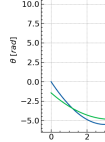


3	23599878844131534	7	2444489740009	12	230098431413154
4	230098431413154	8	230098431413154	13	230098431413154
5	230098431413154	9	230098431413154	14	230098431413154
6	230098431413154	10	230098431413154	15	230098431413154
7	230098431413154	11	230098431413154	16	230098431413154
8	230098431413154	12	230098431413154	17	230098431413154
9	230098431413154	13	230098431413154	18	230098431413154
10	230098431413154	14	230098431413154	19	230098431413154
11	230098431413154	15	230098431413154	20	230098431413154
12	230098431413154	16	230098431413154	21	230098431413154
13	230098431413154	17	230098431413154	22	230098431413154
14	230098431413154	18	230098431413154	23	230098431413154
15	230098431413154	19	230098431413154	24	230098431413154
16	230098431413154	20	230098431413154	25	230098431413154
17	230098431413154	21	230098431413154	26	230098431413154
18	230098431413154	22	230098431413154	27	230098431413154
19	230098431413154	23	230098431413154	28	230098431413154
20	230098431413154	24	230098431413154	29	230098431413154
21	230098431413154	25	230098431413154	30	230098431413154
22	230098431413154	26	230098431413154	31	230098431413154
23	230098431413154	27	230098431413154	32	230098431413154
24	230098431413154	28	230098431413154	33	230098431413154
25	230098431413154	29	230098431413154	34	230098431413154
26	230098431413154	30	230098431413154	35	230098431413154
27	230098431413154	31	230098431413154	36	230098431413154
28	230098431413154	32	230098431413154	37	230098431413154
29	230098431413154	33	230098431413154	38	230098431413154
30	230098431413154	34	230098431413154	39	230098431413154
31	230098431413154	35	230098431413154	40	230098431413154
32	230098431413154	36	230098431413154	41	230098431413154
33	230098431413154	37	230098431413154	42	230098431413154
34	230098431413154	38	230098431413154	43	230098431413154
35	230098431413154	39	230098431413154	44	230098431413154
36	230098431413154	40	230098431413154	45	230098431413154
37	230098431413154	41	230098431413154	46	230098431413154
38	230098431413154	42	230098431413154	47	230098431413154
39	230098431413154	43	230098431413154	48	230098431413154
40	230098431413154	44	230098431413154	49	230098431413154
41	230098431413154	45	230098431413154	50	230098431413154
42	230098431413154	46	230098431413154	51	230098431413154
43	230098431413154	47	230098431413154	52	230098431413154
44	230098431413154	48	230098431413154	53	230098431413154
45	230098431413154	49	230098431413154	54	230098431413154
46	230098431413154	50	230098431413154	55	230098431413154
47	230098431413154	51	230098431413154	56	230098431413154
48	230098431413154	52	230098431413154	57	230098431413154
49	230098431413154	53	230098431413154	58	230098431413154
50	230098431413154	54	230098431413154		

[illegible][illegible][illegible]

```
[ ]: array([ 0.30180666, -0.03139018, -1.44360365])

[ ]: plt.plot(time_1, theta1)
      plt.plot(time_1, omega(time_1, 'wppg_1'))
      plt.xlabel("Time [s]")
      plt.ylabel("Theta [rad]")
      plt.savefig("795_1.png")



[ ]: omega = []

for a in range(len(theta1)):
    omega.append((theta1[a]-theta1[a-1])/frame_rate)

[ ]: time_2 = np.arange(len(omega))/frame_rate
      wppg_2 = np.optimize.curve_fit(time_2, omega,
                                     wppg_2)

[ ]: array([ 0.82087467, -2.30306945])

[ ]: plt.plot(time_2, omega)
      plt.plot(time_2, time_2*wppg_2[0] + wppg_2[1])
```

```

pit.subset(t="times $[3:8]")
pit.plot("timesplit $[4:8] times s~[1:13]")
pit.newvar("gamma_psig")



[ ] alpha = []

for i in range(len(msgage)-1):
    alpha.append(msgage[i+1]-msgage[i])/(frame_rate)

time_3 = np.arange(len(alpha))/frame_rate
page_3 = ~ np.optimize_curve_fit(time_3, alpha)
print(page_3)

pit.plot(np.arange(len(alpha))/frame_rate, alpha)
pit.plot(np.arange(len(alpha))/frame_rate, frame_rate, linetime_3, "page_3")

pit.subset(t="times $[3:8]")
pit.subset("gamma $[4:8]~<20")
pit.newvar("gamma_psig")

[0.0677486 0.6392022]

```

```
[ ] acc_queue = pygame.Queue(10)
acc_lim = pygame.Rect(
    pygame.Rect(acc_lim), acc_lim)
acc = acc_queue.get(acc_lim)
lim = 0.1 * acc * 0.1
pygame.draw.rect
print(f'{g}')

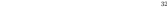
0.0038036114907118 0.6326744702267927
0.01665000377059662

3.4. Video m70

[ ] video_path = "m70.mp4"
cap = cv2.VideoCapture(video_path)

[ ] frame_rate = cap.get(cv2.CAP_PROP_FPS)
print(f'Frame rate = {frame_rate}')

Frame rate: 30.0
```







```

1 frame = []
2 while True:
3     rot, frame = cap.read()
4     if not rot:
5         break
6     frame.append(frame)
7
8 frame_rate = cap.get(cv2.CAP_PROP_FPS)
9 print('Frame rate: ', frame_rate)
10
11 frame_rate = 50.781008007775
12
13 def disp_img(img):
14     plt.imshow(frame[img])
15     plt.show()
16
17 vidwriter = cv2.VideoWriter('out.avi', cv2.VideoWriter_FOURCC('D', 'I', 'V', 'I'), 200, 0, 0, 0)
18 step=1
19
20 interactive(childsw=initSlider(value=0, description='ma', max=100),
21             _dim, _class='vidger-interac-')
22
23 function _main_disp_img(img):
24     _dim, _class = frame[img]
25
26 def pos_track(x,y):
27     img = analyze[img].copy()
28     w, h = img.shape
29     cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 4)
30     plt.imshow(img)
31
32 vidwriter.write(pos_track, cv2.cvtColor(frame[img], cv2.COLOR_BGR2RGB))
33 step+=1
34
35 interactive(childsw=initSlider(value=0, description='ma', max=100),
36             _dim, _class='vidger-interac-')
37
38 function _main_pos_track(x,y):
39     _dim, _class = frame[img]
40
41 tracker = cv2.TrackerDNN_create()
42 track_frame = analyze.copy()
43
44 init_trw = track_frame[img]
45 rot = (0, 20, 50, 50)
46 tracker.init(init_trw, rot)

```

49

```

1 results = []
2 while True:
3     for i in range(1, len(track_frames)):
4         # find the next frame
5         frame = track_frames[i]
6
7         # Update the tracker
8         success, rot = tracker.update(frame)
9
10        if success:
11            # Draw a bounding box around the tracked object
12            (x, y, w, h) = tuple(rot[0:4])
13            results = [(x,y,w,h)]
14            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
15            frame_tracked.append(frame)
16
17        # Get disp_img(img)
18        plt.imshow(frame_tracked[img])
19
20        vidwriter.write(disp, cv2.cvtColor(frame_tracked[img], cv2.COLOR_BGR2RGB))
21        step+=1
22
23        interactive(childsw=initSlider(value=0, description='ma', max=100),
24                    _dim, _class='vidger-interac-')
25
26        function _main_disp_img(img):
27            _dim, _class = frame[img]
28
29        def pos_track(x,y):
30            img = analyze[img].copy()
31            w, h = img.shape
32            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 4)
33            plt.imshow(img)
34
35        vidwriter.write(pos_track, cv2.cvtColor(frame[img], cv2.COLOR_BGR2RGB))
36        step+=1
37
38        interactive(childsw=initSlider(value=0, description='ma', max=100),
39                    _dim, _class='vidger-interac-')
40
41        function _main_pos_track(x,y):
42            _dim, _class = frame[img]
43
44        tracker = cv2.TrackerDNN_create()
45        track_frame = analyze.copy()
46
47        init_trw = track_frame[img]
48        rot = (0, 20, 50, 50)
49        tracker.init(init_trw, rot)

```

50

```

1 if np.ubound(actual_dir_dir[1], next_dir[0]) > np.actbound(gwer_dir[1],
2     gwer_dir[0]) > np.pi:
3     cur_theta = -np.pi
4     np.ubound(actual_dir_dir[1], next_dir[0]) > np.actbound(gwer_dir[1],
5     gwer_dir[0]) > np.pi:
6     theta.append(cur_theta)
7     gwer_dir = next_dir
8     print(theta)
9
10 [0.0, -0.00312088134047, -0.0017662064180824, -0.0064203022661757,
11 -0.00731406306884, -0.00126417600009, -0.17860380206269,
12 -0.2064877999465, -0.23461301892979, -0.26032141234048,
13 -0.28606777226545, -0.30863877460033, -0.33407733573834,
14 -0.35661108176776, -0.37817069728738, -0.40131223232634,
15 -0.421188417016107, -0.44317069728738, -0.46441082071349,
16 -0.48438808223773, -0.5020383126232, -0.52112777766801,
17 -0.53830006282824, -0.5548416171146, -0.57396612171462,
18 -0.5847326644102, -0.5920200097278, -0.61743130126336,
19 -0.6300777131814, -0.64217069728738, -0.6584941768602,
20 -0.66803846477877, -0.67843084413209, -0.6905770622781,
21 -0.6978837701814, -0.7058877999465, -0.7184941768602,
22 -0.7230681188171, -0.7342817069728738, -0.73878969121, -0.7460227325881,
23 -0.7500313849476, -0.75999999999999, -0.7681156416162,
24 -0.7687301263417, -0.7687301263417, -0.7686100141729,
25 -0.7687301263417, -0.7687301263417, -0.7687301263417,
26 -0.7687301263417, -0.7687301263417, -0.7687301263417,
27 -0.7687301263417, -0.7687301263417, -0.7687301263417,
28 -0.7687301263417, -0.7687301263417, -0.7687301263417,
29 -0.7687301263417, -0.7687301263417, -0.7687301263417,
30 -0.7687301263417, -0.7687301263417, -0.7687301263417,
31 -0.7687301263417, -0.7687301263417, -0.7687301263417,
32 -0.7687301263417, -0.7687301263417, -0.7687301263417,
33 -0.7687301263417, -0.7687301263417, -0.7687301263417,
34 -0.7687301263417, -0.7687301263417, -0.7687301263417,
35 -0.7687301263417, -0.7687301263417, -0.7687301263417,
36 -0.7687301263417, -0.7687301263417, -0.7687301263417,
37 -0.7687301263417, -0.7687301263417, -0.7687301263417,
38 -0.7687301263417, -0.7687301263417, -0.7687301263417,
39 -0.7687301263417, -0.7687301263417, -0.7687301263417,
40 -0.7687301263417, -0.7687301263417, -0.7687301263417,
41 -0.7687301263417, -0.7687301263417, -0.7687301263417,
42 -0.7687301263417, -0.7687301263417, -0.7687301263417,
43 -0.7687301263417, -0.7687301263417, -0.7687301263417,
44 -0.7687301263417, -0.7687301263417, -0.7687301263417,
45 -0.7687301263417, -0.7687301263417, -0.7687301263417,
46 -0.7687301263417, -0.7687301263417, -0.7687301263417,
47 -0.7687301263417, -0.7687301263417, -0.7687301263417,
48 -0.7687301263417, -0.7687301263417, -0.7687301263417,
49 -0.7687301263417, -0.7687301263417, -0.7687301263417,
50 -0.7687301263417, -0.7687301263417, -0.7687301263417,
51 -0.7687301263417, -0.7687301263417, -0.7687301263417,
52 -0.7687301263417, -0.7687301263417, -0.7687301263417,
53 -0.7687301263417, -0.7687301263417, -0.7687301263417,
54 -0.7687301263417, -0.7687301263417, -0.7687301263417,
55 -0.7687301263417, -0.7687301263417, -0.7687301263417,
56 -0.7687301263417, -0.7687301263417, -0.7687301263417,
57 -0.7687301263417, -0.7687301263417, -0.7687301263417,
58 -0.7687301263417, -0.7687301263417, -0.7687301263417,
59 -0.7687301263417, -0.7687301263417, -0.7687301263417,
60 -0.7687301263417, -0.7687301263417, -0.7687301263417,
61 -0.7687301263417, -0.7687301263417, -0.7687301263417,
62 -0.7687301263417, -0.7687301263417, -0.7687301263417,
63 -0.7687301263417, -0.7687301263417, -0.7687301263417,
64 -0.7687301263417, -0.7687301263417, -0.7687301263417,
65 -0.7687301263417, -0.7687301263417, -0.7687301263417,
66 -0.7687301263417, -0.7687301263417, -0.7687301263417,
67 -0.7687301263417, -0.7687301263417, -0.7687301263417,
68 -0.7687301263417, -0.7687301263417, -0.7687301263417,
69 -0.7687301263417, -0.7687301263417, -0.7687301263417,
70 -0.7687301263417, -0.7687301263417, -0.7687301263417,
71 -0.7687301263417, -0.7687301263417, -0.7687301263417,
72 -0.7687301263417, -0.7687301263417, -0.7687301263417,
73 -0.7687301263417, -0.7687301263417, -0.7687301263417,
74 -0.7687301263417, -0.7687301263417, -0.7687301263417,
75 -0.7687301263417, -0.7687301263417, -0.7687301263417,
76 -0.7687301263417, -0.7687301263417, -0.7687301263417,
77 -0.7687301263417, -0.7687301263417, -0.7687301263417,
78 -0.7687301263417, -0.7687301263417, -0.7687301263417,
79 -0.7687301263417, -0.7687301263417, -0.7687301263417,
80 -0.7687301263417, -0.7687301263417, -0.7687301263417,
81 -0.7687301263417, -0.7687301263417, -0.7687301263417,
82 -0.7687301263417, -0.7687301263417, -0.7687301263417,
83 -0.7687301263417, -0.7687301263417, -0.7687301263417,
84 -0.7687301263417, -0.7687301263417, -0.7687301263417,
85 -0.7687301263417, -0.7687301263417, -0.7687301263417,
86 -0.7687301263417, -0.7687301263417, -0.7687301263417,
87 -0.7687301263417, -0.7687301263417, -0.7687301263417,
88 -0.7687301263417, -0.7687301263417, -0.7687301263417,
89 -0.7687301263417, -0.7687301263417, -0.7687301263417,
90 -0.7687301263417, -0.7687301263417, -0.7687301263417,
91 -0.7687301263417, -0.7687301263417, -0.7687301263417,
92 -0.7687301263417, -0.7687301263417, -0.7687301263417,
93 -0.7687301263417, -0.7687301263417, -0.7687301263417,
94 -0.7687301263417, -0.7687301263417, -0.7687301263417,
95 -0.7687301263417, -0.7687301263417, -0.7687301263417,
96 -0.7687301263417, -0.7687301263417, -0.7687301263417,
97 -0.7687301263417, -0.7687301263417, -0.7687301263417,
98 -0.7687301263417, -0.7687301263417, -0.7687301263417,
99 -0.7687301263417, -0.7687301263417, -0.7687301263417,
100 -0.7687301263417, -0.7687301263417, -0.7687301263417,
101 -0.7687301263417, -0.7687301263417, -0.7687301263417,
102 -0.7687301263417, -0.7687301263417, -0.7687301263417,
103 -0.7687301263417, -0.7687301263417, -0.7687301263417,
104 -0.7687301263417, -0.7687301263417, -0.7687301263417,
105 -0.7687301263417, -0.7687301263417, -0.7687301263417,
106 -0.7687301263417, -0.7687301263417, -0.7687301263417,
107 -0.7687301263417, -0.7687301263417, -0.7687301263417,
108 -0.7687301263417, -0.7687301263417, -0.7687301263417,
109 -0.7687301263417, -0.7687301263417, -0.7687301263417,
110 -0.7687301263417, -0.7687301263417, -0.7687301263417,
111 -0.7687301263417, -0.7687301263417, -0.7687301263417,
112 -0.7687301263417, -0.7687301263417, -0.7687301263417,
113 -0.7687301263417, -0.7687301263417, -0.7687301263417,
114 -0.7687301263417, -0.7687301263417, -0.7687301263417,
115 -0.7687301263417, -0.7687301263417, -0.7687301263417,
116 -0.7687301263417, -0.7687301263417, -0.7687301263417,
117 -0.7687301263417, -0.7687301263417, -0.7687301263417,
118 -0.7687301263417, -0.7687301263417, -0.7687301263417,
119 -0.7687301263417, -0.7687301263417, -0.7687301263417,
120 -0.7687301263417, -0.7687301263417, -0.7687301263417,
121 -0.7687301263417, -0.7687301263417, -0.7687301263417,
122 -0.7687301263417, -0.7687301263417, -0.7687301263417,
123 -0.7687301263417, -0.7687301263417, -0.7687301263417,
124 -0.7687301263417, -0.7687301263417, -0.7687301263417,
125 -0.7687301263417, -0.7687301263417, -0.7687301263417,
126 -0.7687301263417, -0.7687301263417, -0.7687301263417,
127 -0.7687301263417, -0.7687301263417, -0.7687301263417,
128 -0.7687301263417, -0.7687301263417, -0.7687301263417,
129 -0.7687301263417, -0.7687301263417, -0.7687301263417,
130 -0.7687301263417, -0.7687301263417, -0.7687301263417,
131 -0.7687301263417, -0.7687301263417, -0.7687301263417,
132 -0.7687301263417, -0.7687301263417, -0.7687301263417,
133 -0.7687301263417, -0.7687301263417, -0.7687301263417,
134 -0.7687301263417, -0.7687301263417, -0.7687301263417,
135 -0.7687301263417, -0.7687301263417, -0.7687301263417,
136 -0.7687301263417, -0.7687301263417, -0.7687301263417,
137 -0.7687301263417, -0.7687301263417, -0.7687301263417,
138 -0.7687301263417, -0.7687301263417, -0.7687301263417,
139 -0.7687301263417, -0.7687301263417, -0.7687301263417,
140 -0.7687301263417, -0.7687301263417, -0.7687301263417,
141 -0.7687301263417, -0.7687301263417, -0.7687301263417,
142 -0.7687301263417, -0.7687301263417, -0.7687301263417,
143 -0.7687301263417, -0.7687301263417, -0.7687301263417,
144 -0.7687301263417, -0.7687301263417, -0.7687301263417,
145 -0.7687301263417, -0.7687301263417, -0.7687301263417,
146 -0.7687301263417, -0.7687301263417, -0.7687301263417,
147 -0.7687301263417, -0.7687301263417, -0.7687301263417,
148 -0.7687301263417, -0.7687301263417, -0.7687301263417,
149 -0.7687301263417, -0.7687301263417, -0.7687301263417,
150 -0.7687301263417, -0.7687301263417, -0.7687301263417,
151 -0.7687301263417, -0.7687301263417, -0.7687301263417,
152 -0.7687301263417, -0.7687301263417, -0.7687301263417,
153 -0.7687301263417, -0.7687301263417, -0.7687301263417,
154 -0.7687301263417, -0.7687301263417, -0.7687301263417,
155 -0.7687301263417, -0.7687301263417, -0.7687301263417,
156 -0.7687301263417, -0.7687301263417, -0.7687301263417,
157 -0.7687301263417, -0.7687301263417, -0.7687301263417,
158 -0.7687301263417, -0.7687301263417, -0.7687301263417,
159 -0.7687301263417, -0.7687301263417, -0.7687301263417,
160 -0.7687301263417, -0.7687301263417, -0.7687301263417,
161 -0.7687301263417, -0.7687301263417, -0.7687301263417,
162 -0.7687301263417, -0.7687301263417, -0.7687301263417,
163 -0.7687301263417, -0.7687301263417, -0.7687301263417,
164 -0.7687301263417, -0.7687301263417, -0.7687301263417,
165 -0.7687301263417, -0.7687301263417, -0.7687301263417,
166 -0.7687301263417, -0.7687301263417, -0.7687301263417,
167 -0.7687301263417, -0.7687301263417, -0.7687301263417,
168 -0.7687301263417, -0.7687301263417, -0.7687301263417,
169 -0.7687301263417, -0.7687301263417, -0.7687301263417,
170 -0.7687301263417, -0.7687301263417, -0.7687301263417,
171 -0.7687301263417, -0.7687301263417, -0.7687301263417,
172 -0.7687301263417, -0.7687301263417, -0.7687301263417,
173 -0.7687301263417, -0.7687301263417, -0.7687301263417,
174 -0.7687301263417, -0.7687301263417, -0.7687301263417,
175 -0.7687301263417, -0.7687301263417, -0.7687301263417,
176 -0.7687301263417, -0.7687301263417, -0.7687301263417,
177 -0.7687301263417, -0.7687301263417, -0.7687301263417,
178 -0.7687301263417, -0.7687301263417, -0.7687301263417,
179 -0.7687301263417, -0.7687301263417, -0.7687301263417,
180 -0.7687301263417, -0.7687301263417, -0.7687301263417,
181 -0.7687301263417, -0.7687301263417, -0.7687301263417,
182 -0.7687301263417, -0.7687301263417, -0.7687301263417,
183 -0.7687301263417, -0.7687301263417, -0.7687301263417,
184 -0.7687301263417, -0.7687301263417, -0.7687301263417,
185 -0.7687301263417, -0.7687301263417, -0.7687301263417,
186 -0.7687301263417, -0.7687301263417, -0.7687301263417,
187 -0.7687301263417, -0.7687301263417, -0.7687301263417,
188 -0.7687301263417, -0.7687301263417, -0.7687301263417,
189 -0.7687301263417, -0.7687301263417, -0.7687301263417,
190 -0.7687301263417, -0.7687301263417, -0.7687301263417,
191 -0.7687301263417, -0.7687301263417, -0.7687301263417,
192 -0.7687301263417, -0.7687301263417, -0.7687301263417,
193 -0.7687301263417, -0.7687301263417, -0.7687301263417,
194 -0.7687301263417, -0.7687301263417, -0.7687301263417,
195 -0.7687301263417, -0.7687301263417, -0.7687301263417,
196 -0.7687301263417, -0.7687301263417, -0.7687301263417,
197 -0.7687301263417, -0.7687301263417, -0.7687301263417,
198 -0.7687301263417, -0.7687301263417, -0.7687301263417,
199 -0.7687301263417, -0.7687301263417, -0.7687301263417,
200 -0.7687301263417, -0.7687301263417, -0.7687301263417,
201 -0.7687301263417, -0.7687301263417, -0.7687301263417,
202 -0.7687301263417, -0.7687301263417, -0.7687301263417,
203 -0.7687301263417, -0.7687301263417, -0.7687301263417,
204 -0.7687301263417, -0.7687301263417, -0.7687301263417,
205 -0.7687301263417, -0.7687301263417, -0.7687301263417,
206 -0.7687301263417, -0.7687301263417, -0.7687301263417,
207 -0.7687301263417, -0.7687301263417, -0.7687301263417,
208 -0.7687301263417, -0.7687301263417, -0.7687301263417,
209 -0.7687301263417, -0.7687301263417, -0.7687301263417,
210 -0.7687301263417, -0.7687301263417, -0.7687301263417,
211 -0.76
```

