

Sommaire

Introduction	1
Objectif	1
Cahier de charge	2
Setup	3
Caractéristique de notre univers de poche	3
1.1. Le tissu	3
1.2. Le cercle en PVC	4
1.3. Le support de cercle	4
1.4. Les épingles	4
Modélisation de notre univers	4
Paramètres de contrôles	5
3.1. Le poids de masse centrale	5
3.2. La dimension des billes	6
3.3. La vitesse des billes et l'angle de lancement	6
3.4. La caméra	7
Lanceur	7
Trajectoire théorique	8
Vitesse théorique initiale	8
Trajectoire théorique : force gravitationnelle	9
2.1. Première loi de Kepler	9
2.2. Deuxième loi de Kepler	9
2.3. Troisième loi de Kepler	9
Trajectoire théorique: modélisation de la déformation du tissu	10
3.1. Détection du contours du tissu	12
3.2. Prétraitement	14
3.3. Extraction de la partie utile de la courbe	15
3.4. Approximation de la courbe par un polynôme	16
3.5. Simulation de la trajectoire de la bille	19
Trajectoire expérimentale	22
Estimation de trajectoire par le filtre de Kalman	22
1.1. Équation d'état, équation d'observation et modèle physique	22
1.2. Mise en place de l'algorithme de Kalman appliqué au suivi de la bille	23
1.3. Résultat de la méthode	24
Estimation de trajectoire par la méthode de différentiation	26
Comparaison des resultats theoriques et experimentales	28
Vérification des lois de Kepler	28
Vérification du modèle théorique	29
Conclusion	30
Bibliographie	31
Annexes	31

I. Introduction

Pour comprendre les notions complexes en physique, on commence toujours par étudier des modèles simples qui nous permettent de faire une analyse préliminaire avant d'aborder les détails. Dans ce projet, notre but est de réaliser un modèle réaliste pour étudier la gravité des astres spatiaux avec les notions comme le traitement d'image, la mécanique, l'automatique etc. En créant notre univers de poche, on essaye de mieux comprendre le fonctionnement notre univers.

II. Objectif

Tout d'abord, notre objectif est de modéliser un univers avec des objets ordinaires qui représentent bien le fonctionnement des lois de Kepler sur le mouvement planétaire. Pour la modélisation de l'univers, on utilise des différents objets qui nous aident à simuler « l'attraction gravitationnelle ».

Après la réalisation de notre setup, un autre objectif principal est de prendre une vidéo d'astres évoluant dans notre univers, pour ensuite effectuer un traitement de vidéo. Avec ce traitement, on veut pouvoir suivre et tracer la trajectoire des astres autour du corps céleste et de les prédire avec les différentes méthodes. A partir de ces résultats, on pourra comparer les résultats expérimentaux avec les résultats théoriques.

Un des derniers objectifs est de faire un traitement de vidéo en live pour vérifier notre modèle et notre prédiction. Et finalement, un objectif « bonus » est d'implémenter une énergie dans l'univers pour modéliser au mieux les forces de l'univers. En effet, avec le système actuel, les astres subissent des frottements qui modifient leurs trajectoires. On pourrait ainsi compenser ces frottements. Ce dernier objectif « bonus » sera donc réalisé une fois que les objectifs précédents seront atteints.

III. Cahier de charge

Énoncé de la fonction	Critère d'appréciation	Niveaux d'exigence
L'univers doit être modélisable	<ul style="list-style-type: none"> - Le corps céleste - Astres - Univers 	<ul style="list-style-type: none"> - Bouteilles de l'eau de 500ml - Les billes - Lycra, un cercle, 6 pieds en bois
L'univers doit être quantifiable	<ul style="list-style-type: none"> - Dimensions de cercle - Déformabilité du tissu - Histogramme d'intensité de tissu 	<ul style="list-style-type: none"> - Périmètre : 475 cm - Demi-grand axe : 80,5 cm - Demi-petit axe : 70 cm
Tout ce qui est autour et dans l'univers doit être contrôlable	<ul style="list-style-type: none"> - Angle de lancement - La vitesse de la bille à la sortie du lanceur - Dimension des billes - La grande masse 	
Réaliser des vidéos traitables	<ul style="list-style-type: none"> - Clarté des mouvements des billes dans les vidéos 	<ul style="list-style-type: none"> - Caméra fixée au plafond
Les trajectoires de billes doivent être observable	<ul style="list-style-type: none"> - Les couleurs des billes et lycra 	Noir, bleu et vert sur le sol claire
Traitement hors ligne doit être réalisable	<ul style="list-style-type: none"> - Identification de contour des billes, la masse et l'univers - Suivre et prédire la trajectoire de la bille 	<ul style="list-style-type: none"> - Méthode de gradient - Filtrage Kalman et méthode de différentiation
Réalisation d'un modèle théorique de l'univers et simuler des trajectoires	<ul style="list-style-type: none"> - Obtenir un tracée de trajectoire cohérente avec la réalité 	<ul style="list-style-type: none"> - Prise en compte de la déformation du tissu - Prise en compte des frottement

Figure 1 : Le tableau de cahier de charge

IV. Setup

Le setup est une des parties les plus importantes du projet. Pour la modélisation de l'environnement, les astres, le corps céleste et leurs mouvements, on a utilisé différents objets qui peuvent nous servir. Voici une liste des objets qui nous ont servi :

Modélisation de l'environnement
Un tissu
Un cercle en pvc
Les supports de cercle
Les épingles
Modélisation du corps céleste
Les disques en bois
Les bouteilles de l'eau
Modélisation des astres
Les billes

Figure 2 : La liste des objets utiles à la modélisation

1. Caractéristique de notre univers de poche

Avant d'effectuer le traitement d'image et faire une étude théorique, il est nécessaire de connaître tous les paramètres de notre modèle: les dimensions des objets qu'on a utilisés, leurs emplacements, leurs propriétés etc.

1.1. Le tissu

Le tissu qu'on a utilisé est fait en lycra, une matière assez flexible qui nous permet de faire illusion de la traction gravitationnelle mais peu déformable sous les poids des objets qu'on utilise pour la modélisation de corps principale. Ses dimensions sont de 1.50 m par 1.56m. En outre, on s'intéresse à son histogramme de RVB ce qui est important pour réaliser les traitements vidéo.

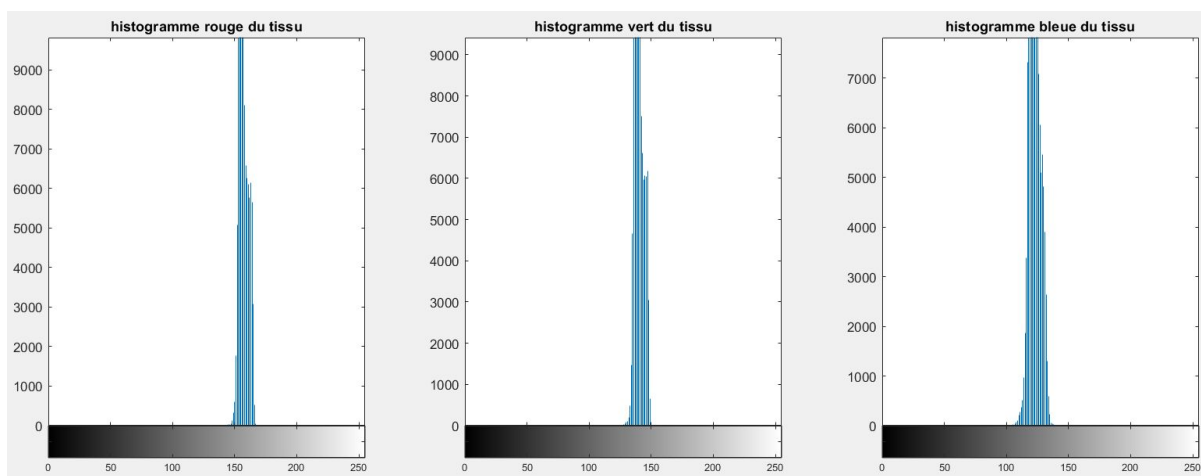


Figure 3 : L'histogramme RVB de tissu

Les couleurs du tissu sont composées d'un pic pour chaque couleur primaire. Ces pics ont pour moyenne d'intensité les suivantes : en rouge 156, en vert 140, et en bleu 122.

1.2. Le cercle en PVC

Pour le cercle, on utilise un tuyau en PVC de diamètre de 2 cm et de longueur de 475 cm. En tordant le tuyau dans une forme circulaire, on voulait créer un « contour » pour l'univers.

1.3. Le support de cercle

Pour support de cercle, on a utilisé 6 piliers en bois en dimension de 7 cm x 7 cm x 80 cm. Ils sont répartis équitablement, un pilier tous les 80 cm, en dessous de cercle en PVC.

1.4. Les épingles

Une vingtaine d'épingles a été utilisée pour maintenir accrocher le tissu sur l'anneau. Les épingles sont aussi réparties équitablement sur le tissu pour avoir une tension équilibrée sur le tissu et aucune région du tissu ne subit une force différente qu'une autre.

2. Modélisation de notre univers

Notre univers peut être modélisé de deux manières différentes: soit par un cercle, soit par une ellipse. Après des mesures plus précises, en ré-utilisant le périmètre du cercle et en utilisant des différentes formules pour un cercle et pour une ellipse, nous déterminons les paramètres suivants :

Pour le cercle:

Périmètre	475 cm
Rayon	75.6 cm
Aire	1.80 m ²

Figure 4 : Le tableau des paramètres du cercle

Pour l'ellipse :

Périmètre	475 cm
Demi-grand axe	80,5 cm
Demi-petit axe	70 cm
Excentricité	0.48
Aire	1.77 m ²

Figure 5 : Le tableau des paramètres de l'ellipse

3. Paramètres de contrôles

Tout d'abord, on a défini quelques paramètres de contrôle lorsqu'on a préparé notre setup.

3.1. Le poids de masse centrale

Le premier paramètre est la masse qu'on a mise au centre du tissu qui représente le corps céleste. On a voulu créer un système où on peut jouer sur l'assouplissement de ce tissu. On voulait observer le comportement de la bille, son accélération, sa trajectoire vis-à-vis des masses de différentes poids et de différentes dimensions. Pour cela, on a manufacturé des disques de bois de 5 mm d'épaisseur et de diamètre. En posant des bouteilles plastiques remplies d'eau sur ces disques, on a obtenu un intervalle de poids de 500 g à 1000 g.

Diamètre de disques (en cm)	Poids (en g)
30	271
20	116
10	30

Figure 6 : Le tableau des paramètres des disques

Type de bouteille	Poids (la bouteille remplie totalement)
Une bouteille de Volvic	535 g
Une bouteille de Cristaline	481 g

Figure 7 : Le tableau des paramètres des bouteilles

3.2. La dimension des billes

Notre deuxième paramètre est la dimension des billes. Les billes représentent les corps qui tournent autour de la masse centrale. Pour observer l'effet de leurs dimensions sur leur mouvement, on a commandé des billes de diamètres différents et de poids différents.

Diamètre (en mm)	Poids (en g)
12	2,41
15	5,48
22	18,5

Figure 8 : Le tableau de paramètres des billes

3.3. La vitesse des billes et l'angle de lancement

Les autres paramètres importants à contrôler sont la vitesse de la bille à la sortie du lanceur, et donc l'angle du lancement de la bille sur le tissu. Ainsi, on a réussi à les contrôler avec le lanceur qu'on a désigné. En changeant l'inclinaison du lanceur, on joue sur la vitesse à la sortie. L'angle d'inclinaison choisi varie entre 0° et 45° et on a une liberté d'angle de lancement entre 0° et 60°.

3.4. La caméra

Un des derniers paramètres de contrôle est la distance entre la caméra et le setup. Tout d'abord, on a choisi de mettre le cadre au plafond avec une pièce désignée pour attacher la caméra au plafond afin d'avoir une vue complète du setup. La distance entre-eux mesure 2,58 m. Il est possible de diminuer cette distance mais on risque de perdre complètement notre vue et du coup, de perdre des informations nécessaires. Également, l'autre paramètre qu'on peut contrôler est la fréquence de l'image récupérée. Ici, avec cette caméra, cette valeur est de 24.99 images/seconde. On peut augmenter cette valeur pour une étude plus délicate en remplaçant la caméra actuelle avec une autre caméra de meilleure qualité.

4. Lanceur

Notre lanceur est formé des pièces à monter. Une pièce à visser sur le support de l'anneau qui a une forme de demi-disque de diamètre de 15 cm nous permet de mesurer facilement l'angle de lancement. Une deuxième pièce a une liaison pivot avec le demi-disque. Elle a une partie cylindrique en bas d'épaisseur 1 cm de diamètre de 2 cm qui en contact avec le demi-cercle. De plus, le reste est une forme de rectangle et tout en haut, elle est creuse de forme cylindrique pour qu'elle puisse supporter la troisième pièce, le tuyau. Ce tuyau joue un rôle de toboggan entre la pièce qu'on vient de décrire et la dernière pièce. Cette dernière pièce est juste une longue tige en bois. On l'a percé sur plusieurs points pour avoir des différents angles d'inclinaison. Entre chaque trou, il y a 2 cm d'espace. Le premier trou se trouve à 2 cm au-dessus du niveau de l'anneau qui tient le tissu. Cela nous permet d'avoir un petit angle pour le lancement des billes dans le tuyau en appliquant aucune force. Avec une tige filetée qui passe par ces trous, elle forme un support pour le tuyau. En changeant le placement de la tige filetée, on modifie l'inclinaison. Voici l'image qui peut illustrer cette description.

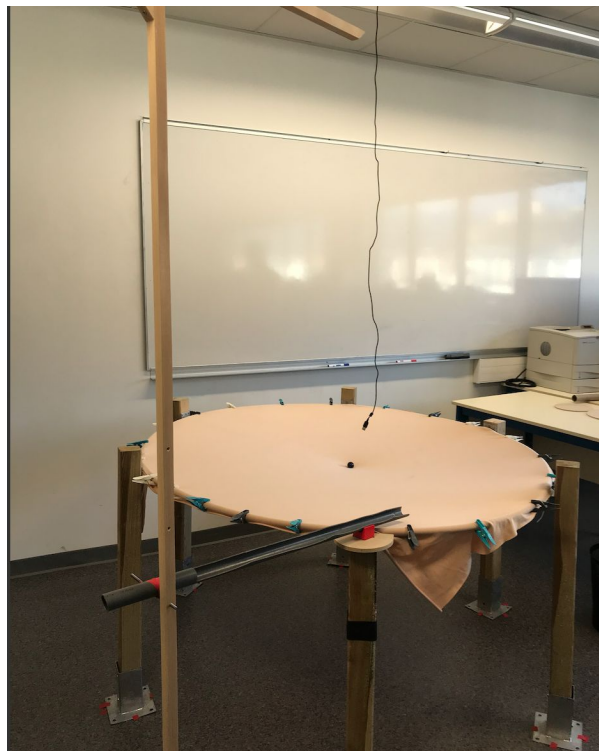


Figure 9 : L'image de lanceur

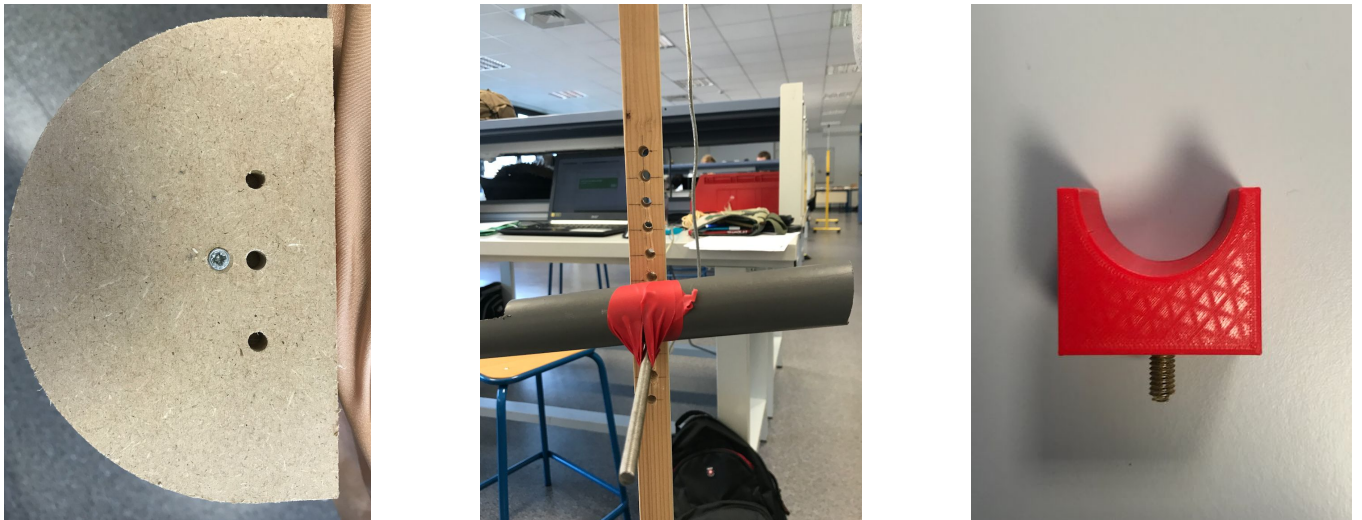


Figure 10 : Les images des pièces du lanceur

V. Trajectoire théorique

1. Vitesse théorique initiale

La vitesse initiale, comme indiqué dans la partie de paramètre de contrôle, est très importante pour prédire le comportement de la bille dans notre univers. Cela nous a amené à faire l'étude suivante.

Nous souhaitons ici donc déterminer quelles seront les conditions initiales de la bille en sortie du lanceur et notamment sa vitesse.

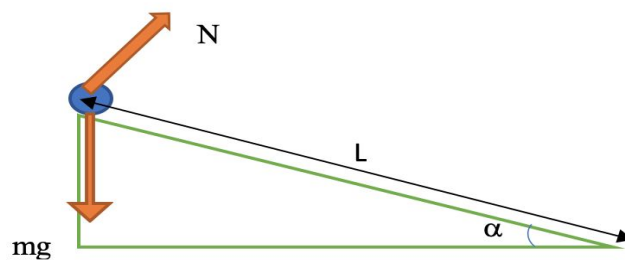


Figure 11 : Le modelé du lanceur

En haut du lanceur, la bille subit son poids, mg , et la force réaction du tuyau sur la bille, N . Ici, nous négligeons les forces de frottement donc la force tangentielle.

Après application du principe fondamental de la dynamique (PFD) et intégrations, nous déterminons la relation entre la vitesse de sortie du lanceur et l'angle du verticale du lanceur :

$$V = \sqrt{2Lg\sin(\alpha)}$$

Avec L , la longueur du lanceur de 84 cm. Cela nous donne :

$$V = 4.06 \times \sqrt{\sin(\alpha)}$$

2. Trajectoire théorique : force gravitationnelle

Pour mener une étude théorique sur la trajectoire théorique, on se réfère les lois de Kepler sur le mouvement planétaire.

2.1. Première loi de Kepler

« L'orbite de chaque planète est une ellipse avec le Soleil à l'un des deux foyers »

Mathématiquement, une ellipse peut être représentée par la formule suivante :

$$r = \frac{p}{1+e\cos\theta} ,$$

avec p le rectum demi-latus, e l'excentricité de l'ellipse, r la distance du corps céleste à l'astre, et l'angle θ l'angle entre la position actuelle de la planète depuis sa dernière position, vu du corps céleste. Donc (r, θ) sont des coordonnées polaires.

Pour $\theta = 0^\circ$, on trouve r_{min} (demi-petit axe) et pour $\theta = 180^\circ$, on trouve r_{max} (demi-grand axe). Le rectum demi-latus p est la moyenne harmonique entre r_{min} et r_{max} . Pour une ellipse, on a $0 < e < 1$, et dans le cas limite $e = 0$, l'orbite est un cercle avec le corps céleste au centre. On s'intéressera à comparer la forme de trajectoire expérimentale avec la théorie de Kepler.

2.2. Deuxième loi de Kepler

« Une ligne joignant une planète et le Soleil balaie des zones à intervalles de temps égaux »

Cela implique que le rayon orbital et la vitesse angulaire de l'astre sur l'orbite varieront. Pendant une durée de dt , la planète balaie un petit triangle ayant une ligne de base r , une hauteur de $r d\theta$ et une aire de $dA = \frac{1}{2}r^2 d\theta$.

Alors la vitesse surfacique constante est :

$$\frac{dA}{dt} = \frac{1}{2}r^2 \frac{d\theta}{dt} .$$

On regarde si la bille se déplace plus rapidement lorsqu'elle est plus proche de la masse centrale, puis plus lentement lorsqu'elle est plus éloignée de la masse, pendant le traitement de vidéo.

2.3. Troisième loi de Kepler

« Le carré de la période orbitale d'une planète est directement proportionnel au cube du demi-grand axe de son orbite »

Cela détecte la relation entre la distance de la bille de la masse centrale et sa période orbitale.

$$\frac{a^3}{T^2} = \frac{G(M+m)}{4\pi^2} \approx \frac{GM}{4\pi^2} ,$$

où M est la masse de corps céleste, m la masse de l'astre, G la constante gravitationnelle, T la période orbitale et a le demi-axe elliptique.

Avec cette formule, on peut trouver la constante gravitationnelle de notre univers.

3. Trajectoire théorique: modélisation de la déformation du tissu

Lorsqu'on place un poids au centre de notre univers, le tissu se déforme. C'est cette déformation du tissu qui permet aux billes d'effectuer leurs trajectoires elliptiques. La bille, par l'inclinaison du tissu vers le poids, est attirée vers le poids de la même manière que la Lune est attirée par la Terre. Cependant, en plus de cette force en direction du poids, la bille subit les frottements entre le tissu et la bille.

Pour pouvoir comparer les trajectoires expérimentaux avec des trajectoires théoriques, nous allons effectuer une simulation de la bille, en prenant en compte les deux forces évoquées précédemment. L'idée est de donner une situation initiale, c'est-à-dire une position et une vitesse, pour obtenir la trajectoire de la bille en prenant en compte ces deux forces.

Pour itérer les positions au fur et à mesure, il suffit d'appliquer la formule de Taylor-Young à l'ordre 2, sur la position :

$$\vec{p}(t+dt) = \vec{p}(t) + \vec{v}(t).dt + \frac{\vec{a}(t)}{2}.dt^2,$$

avec $\vec{p}(t)$ le vecteur position à l'instant t , $\vec{v}(t)$ le vecteur vitesse de la bille à l'instant t , avec $\vec{a}(t)$ le vecteur accélération de la bille à l'instant t , et avec dt une constante symbolisant le pas de temps.

La vitesse étant la dérivée de la position, on peut mettre à jour la vitesse de la manière suivante (en reprenant la formule de Taylor-Young à l'ordre 1) :

$$\vec{v}(t) = \frac{\vec{p}(t+dt) - \vec{p}(t)}{dt}$$

L'accélération peut être déterminé à partir du principe fondamental de la dynamique :

$$\vec{a}.m = \sum_i \vec{F}_i,$$

avec m la masse de la bille, et \vec{F}_i les forces que la bille subit.

Les forces que subit la bille sont son poids d'expression :

$$\vec{f}_p = m.g.\sin(\alpha(p)).\vec{u},$$

avec g la constante de pesanteur, et $\alpha(p)$ l'angle entre le tissu et la verticale à la position p , et \vec{u} un vecteur unitaire en direction du centre.

La bille subit aussi des forces de frottement, qu'on écrit comme étant :

$$\vec{f}_f = -m.f.\vec{v},$$

avec \vec{v} le vecteur vitesse, et f un coefficient de frottement.

Le schéma suivant décrit les forces que la bille subit :

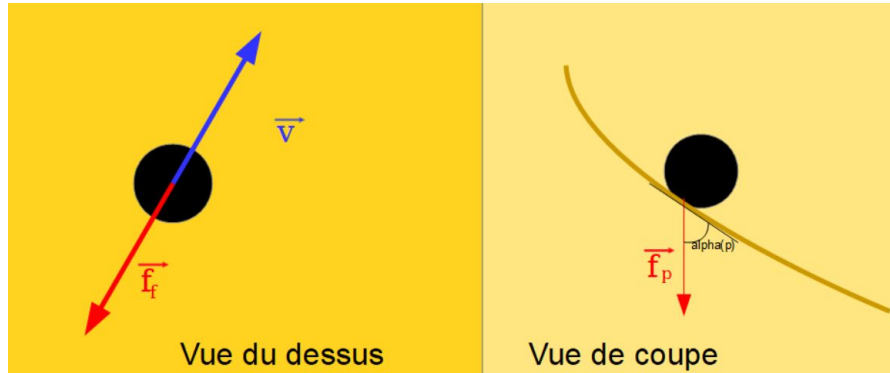


Figure 12 : Schéma explicatif des forces que la bille subit

L'accélération de la bille s'écrit donc ainsi dans notre système :

$$\vec{a} = g \cdot \sin(\alpha(p)) \cdot \vec{u} - f \cdot \vec{v}$$

Puisque nous connaissons déjà \vec{v} , le seul inconnu à déterminer est donc $\alpha(p)$ lié à la manière dont le tissu se déforme.

L'idée pour obtenir ce paramètre consiste à prendre une photo de la déformation du tissu, en déduire sa courbe de déformation, et enfin, obtenir l'angle "instantané" en fonction de la position dans l'espace. En considérant l'univers comme un cercle, il suffit d'observer la déformation du tissu selon une coupe verticale. Ainsi du point de vue du centre (où se trouve le poids), la déformation est identique, et ce peu importe la direction. Il n'est plus nécessaire de connaître la position de la bille pour déterminer l'angle, mais seulement la distance qui la sépare au centre.

Voici une photo de la déformation du tissu qui permet d'obtenir ce résultat :

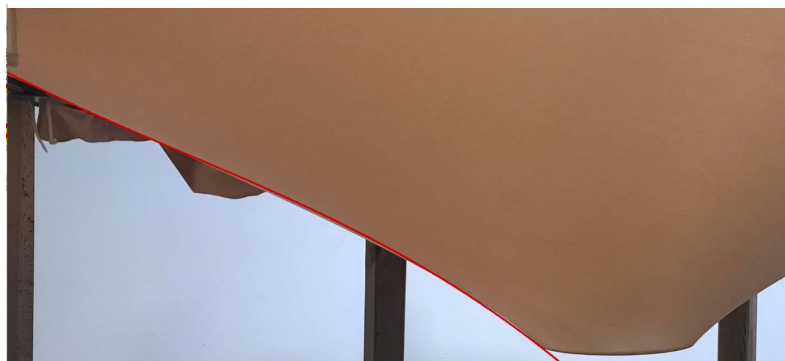


Figure 13 : Déformation du tissu

Ici, il a été placé au centre un disque de 20 cm de diamètre, et de 1 kg. La courbe rouge représente la déformation du tissu. On observe que l'angle d'inclinaison est le plus fort proche du poids. Cependant, il évolue, décroît, en s'éloignant puis devient constant.

Cette courbe rouge sera obtenue en effectuant du traitement d'image, c'est-à-dire en effectuant un moyennage de la détection de contours, et divers prétraitement.

Une fois cette courbe obtenue, nous allons approximer cette courbe par un polynôme à l'aide de la méthode du gradient à pas fixe. Cette approximation va permettre de calculer plus facilement la courbe dérivée de la courbure.

L'angle qu'on souhaite déterminer à partir de cette courbe sera calculé de la manière suivante :

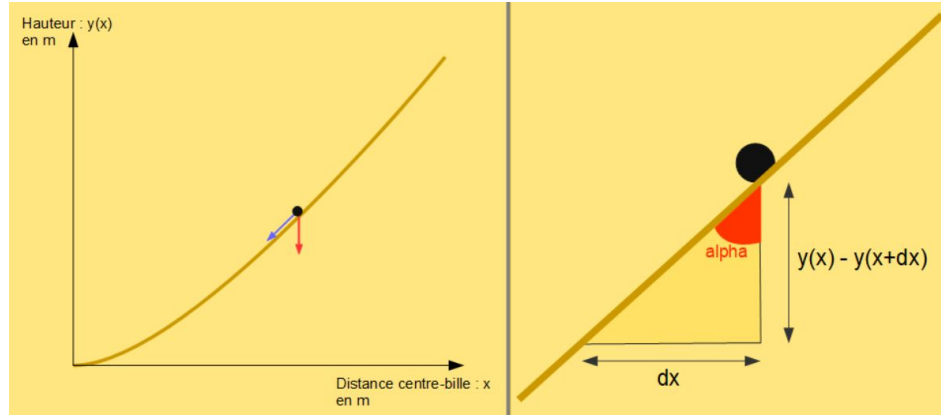


Figure 14 : Schéma explicatif du calcul de l'angle d'inclinaison

Sur la partie gauche de la figure, on retrouve une représentation de la courbe de déformation notée ici $y(x)$ et sur la droite, un zoom. Vu que nous souhaitons déterminer l'angle alpha en position x , nous pouvons nous servir du triangle formé par les distances infinitésimales. Ainsi on peut noter :

$$\alpha(x) = \arctan\left(\frac{dx}{y(x) - y(x+dx)}\right)$$

On reconnaît dans le arctangente, l'inverse de la dérivée de la fonction y . Cela permet d'écrire :

$$\alpha(x) = \arctan\left(\frac{1}{y'(x)}\right)$$

Une fois l'angle alpha déterminé, nous avons tous les outils pour simuler la trajectoire de la bille à partir de l'équation d'itération.

Le premier défi de cette simulation va donc être de modéliser au mieux la déformation du tissu. Voici le détail des étapes permettant d'accomplir ce défi.

3.1. Détection du contours du tissu

L'image sur laquelle nous travaillons est une image du dessous de l'univers, pour pouvoir observer la déformation du tissu. Nous prendrons l'image ci-dessous comme exemple :

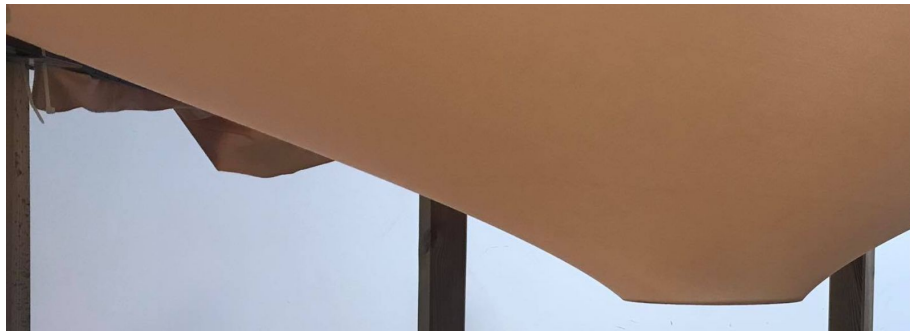


Figure 15 : Image de départ

Nous effectuons dans un premier temps une inversion de l'image, pour que la courbe de déformation croît au lieu de décroître. Cela permettra d'obtenir une courbe qui évolue en fonction de l'éloignement avec le centre.

Pour détecter le contour du tissu, nous allons appliquer le masque de convolution suivant :

0	1	0
0	4	0
2	-14	2
0	4	0
0	1	0

Le masque est volontairement étiré verticalement, puisque la séparation entre le tissu et le reste se fait principalement vers la verticale.

Avant ce filtrage, nous appliquons sur l'image un masque moyennneur, pour rendre le tissu le plus homogène possible : `masque = fspecial('disk',3);`

Nous effectuons ensuite un balayage de haut en bas, pour détecter les hautes valeurs correspondant à la bordure du tissu. Le seuil de détection utilisé dans le cas de notre image est de 30. Nous plaçons un point rouge permettant de repérer où se trouve la bordure selon le programme.

Nous obtenons ce résultat :

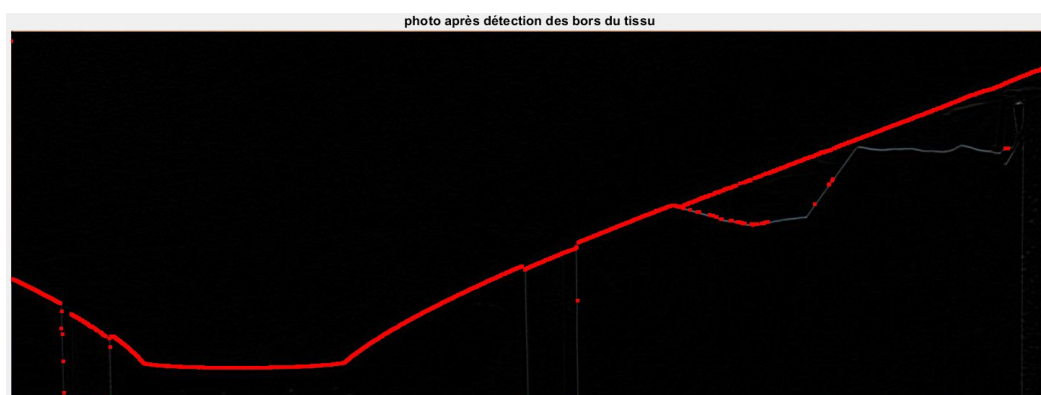


Figure 16 : Détection de la bordure

On peut voir que globalement la méthode fonctionne très bien. On repère la forme du tissu. Cependant un nombre non négligeable de points aberrants ne correspondant pas à la bordure du tissu sont apparus.

De cette méthode, nous en retirons un vecteur contenant les positions verticales des points rouges. Ce vecteur correspond ainsi à la courbe de déformation du tissu.

Avant de passer à la suite des traitements, il est cependant nécessaire d'effectuer un prétraitement sur cette courbe, pour notamment éliminer les points aberrants.

La fonction implémentée pour effectuer cette première action de détection du contours du tissu est :

Courbe = *DetectionCourbure*(*Im*, *seuil_detection*, *numfigure*);

Elle prend en argument : l'image *Im* sur laquelle il faut travailler, le seuil *seuil_detection* permettant de considérer un point comme faisant partie de la bordure, et *numfigure* permettant d'afficher quelques figures à partir du numéro *numfigure*.

La fonction renvoie la courbe correspondant à la déformation du tissu.

3.2. Prétraitement

La première étape effectuée pour éliminer ces points aberrants est de parcourir la courbe et d'observer si il y a un saut important entre deux points. Si c'est le cas, la valeur est considérée comme aberrante, et on lui affecte la valeur précédente. Dans notre exemple, le seuil déterminant si la variation entre deux points est trop important est de 30.

Ce premier traitement est efficace et permet d'éliminer les valeurs les plus aberrantes. Cependant, elle laisse quelques valeurs aberrantes et crée des plateaux sur la courbe.

Pour améliorer la courbe, nous réalisons un lissage. Pour cela, nous convoluons la courbe avec un vecteur d'un nombre de *moy* de $1/moy$ ($ones(1,moy)/moy$). Dans l'exemple nous utilisons un vecteur de *moy*=10.

Voici les résultats que nous obtenons lors de cette étape :

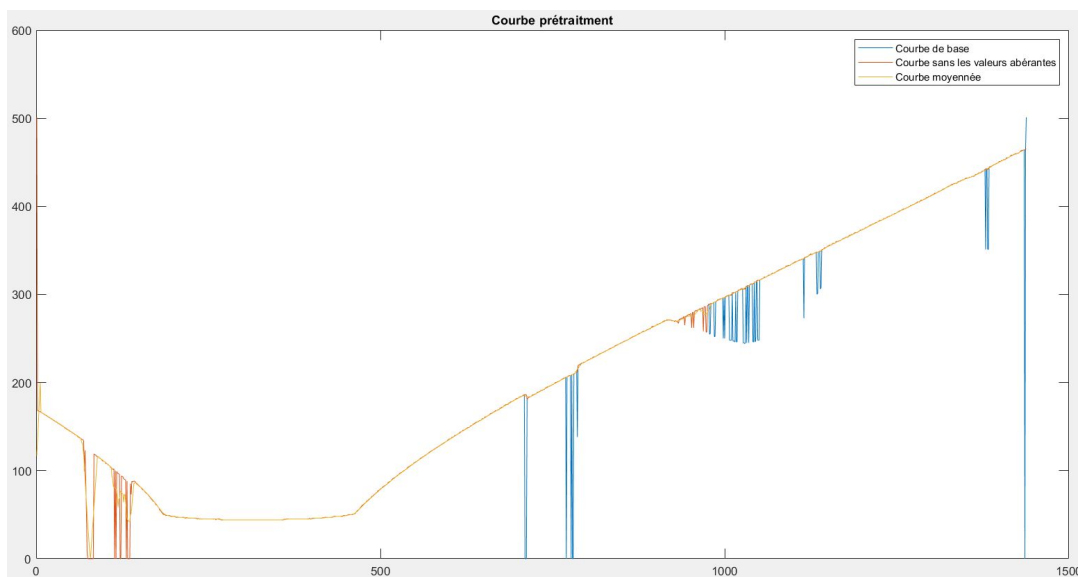


Figure 17 : Prétraitement de la courbe. En bleu, la courbe d'origine. En rouge, la courbe après élimination des points aberrants. En jaune, la courbe après lissage.

La courbe a été améliorée mais il reste encore de nombreux défauts. Sur la partie de gauche de la courbe, il reste beaucoup de points aberrants. On observe des légers sauts qui ne correspondent pas à la déformation du tissu.

Cependant, ces défauts ne seront pas un problème pour la suite des traitements. En effet, nous ne souhaitons conserver que la partie droite de la courbe. L'univers est considéré comme un cercle. Ainsi, la déformation du tissu entre le centre de l'univers et l'extérieur est suffisante pour modéliser la déformation de l'univers en entier. De plus, nous allons ensuite approximer la partie de droite par une fonction polynomiale. La courbe aura alors aucun défaut.

Nous avons obtenu une belle courbe mais nous nous intéressons qu'à la partie droite de la courbe (après le plateau correspondant au disque posé au centre). Il faut donc déterminer précisément où commence cette courbe, où finit le plateau.

La fonction permettant d'effectuer le prétraitement décrit est :

MoyCourbe = *pretraitement*(*Courbe*, *seuil_val_aberant*, *moy*, *numfigure*);

Elle prend en argument : *Courbe* la courbe à traiter, *seuil_val_aberant* donnant la valeur à partir de laquelle une variation est jugée comme trop importante, *moy* la valeur de moyennage, et *numfigure* donnant le numéro de la figure sur laquelle la fonction peut afficher.

La fonction renvoie la courbe sans valeur aberrante, et lissée.

3.3. Extraction de la partie utile de la courbe

Pour extraire la partie de droite de la courbe, nous allons déterminer à quel indice correspond la fin du plateau. Puisque c'est un plateau, la variation de la courbe à cette endroit est quasiment nulle, et ce sur un grand nombre de points. L'idée est donc de calculer la dérivée de la courbe, parcourir la courbe, et de compter les valeurs successives (quasiment) nulles. Si le nombre de valeurs successives nulles est suffisamment important alors cela correspond au plateau, et on trouve ainsi le début de notre courbe.

Pour calculer la courbe dérivée, nous convoluons la courbe avec le vecteur : $[1; -1]$

Voici la courbe dérivée obtenue :

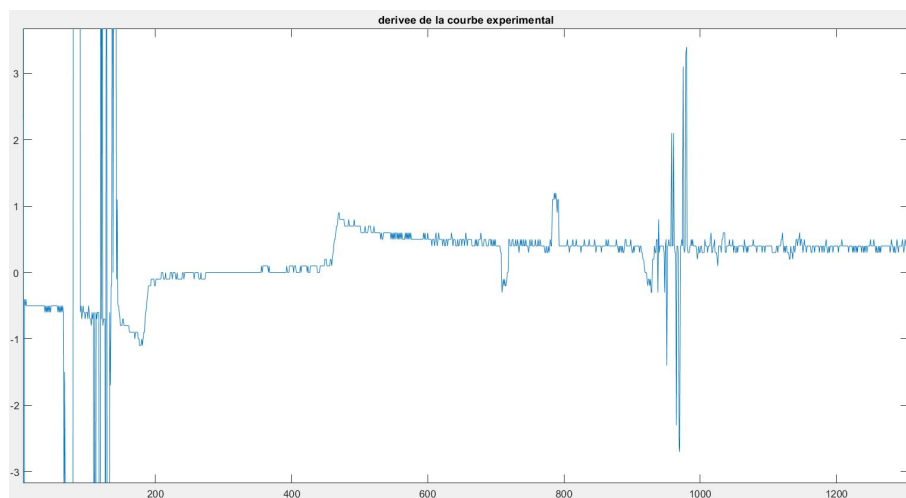


Figure 18 : Courbe dérivée. Zoom sur la partie intéressante.

On peut observer vers les indices 200 à 400 les valeurs nulles correspondant au plateau qu'on souhaite repérer. Nous traversons les valeurs de la courbe et comptons les valeurs consécutives inférieures (en valeur absolue) à une valeur val_max (dans l'exemple de cette valeur est de 0.4), et il y a plus de $long_mini$ valeur quasiment nulle (dans l'exemple de 100), alors on considère que cela correspond au plateau. Lorsque les valeurs arrêtent d'être quasiment nulle, l'indice correspond à celui à partir duquel notre nouvelle courbe commence. Il suffit donc d'extraire la courbe à partir de cette indice.

Nous normalisons cette nouvelle courbe obtenue par sa première valeur.

Nous avons ainsi obtenu la courbe correspondant à la déformation du tissu à partir du centre. Nous allons pouvoir approximer cette courbe par une fonction polynomiale.

La fonction utilisée pour effectuer cette extraction est la suivante:

$NCourbe = CourbeAprèsDisk(CourbeMoy, val_max, long_mini, numfig);$

Elle prend comme argument : $CourbeMoy$ la courbe après traitement, val_max la valeur en-dessous de laquelle on considère une valeur comme quasiment nulle, $long_mini$ la longueur minimum que doit faire une suite de valeur nulle pour la considérée comme étant le plateau, $numfig$ le numéro de la figure sur laquelle la fonction peut afficher.

La fonction retourne en sortie la courbe commençant au bon endroit.

3.4. Approximation de la courbe par un polynôme

Pour approximer la courbe par un polynôme d'ordre n , nous appliquons la méthode du gradient à pas fixe. Le critère que nous souhaitons optimiser est l'erreur quadratique moyen entre la courbe et la courbe approximée (par le polynôme). Le vecteur que nous souhaitons optimiser est le vecteur contenant les coefficients du polynôme. Dans ce vecteur, nous fixons cependant la valeur du coefficient d'ordre 0, à 0, puisque la courbe a été normalisée, et que nous souhaitons modéliser efficacement la forte variation de la courbe proche du centre.

On note $P_n(x)$ la fonction polynomiale approximant la courbe, N le nombre de points dans la courbe, et \underline{X} le vecteur des coefficients de $P_n(x)$, coefficients notés a_i .

On rappelle que l'algorithme du gradient consiste à itérer le vecteur \underline{X} , de la manière suivante :

$$X^{j+1} = X^j + \rho \cdot w^j,$$

avec

$$w = -\nabla_x J(X),$$

opposé au gradient du critère J .

Le critère s'écrit dans notre cas comme suit :

$$J = \sqrt{\sum_{k=1}^N (P_n(k) - courbe(k))^2}$$

Le gradient de notre problème est un vecteur de n dérivées. La i-ème dérivée s'écrit comme suit :

$$\frac{\partial J(X)}{\partial a_i} = \frac{\sum_{k=1}^N k^i \cdot (a_0 + \dots + a_i \cdot k^i + \dots + a_n \cdot k^n - courbe(k))}{\sqrt{\sum_{k=1}^N (a_0 + \dots + a_i \cdot k^i + \dots + a_n \cdot k^n - courbe(k))^2}}$$

On remarque qu'il y a la même expression au dénumérateur et au numérateur. Sous Matlab, nous écrivons cette expression de la manière suivante :

$$ecart(k) = \underline{K}' \cdot \underline{X} - Courbe(k), \text{ avec } \underline{K} = (1 ; k ; k^2 ; \dots)$$

A partir de ces expressions, nous pouvons donc implémenter l'algorithme et le tester, avec un pas ρ de l'ordre de 0.001, 0.0001.

Pour comparer le résultat obtenu, nous utilisons aussi la fonction *polyfit()*; permettant de donner le même résultat qu'on souhaite obtenir.

Avec les premiers tests, nous nous rendons rapidement compte que les coefficients d'ordre élevé évoluent trop rapidement. Le faible changement qu'implique le pas est déjà trop élevé pour obtenir une valeur cohérente. Il faudrait donc diminuer fortement le pas, ce qui rendrait la variation des coefficients d'ordre faible trop lente. Il faudrait ainsi augmenter considérablement le nombre d'itérations à effectuer.

Pour éviter ce changement, et avoir un temps de calcul acceptable, nous modifions le vecteur \underline{X} à optimiser :

$$\underline{X} = [a_0 ; a_1 \cdot 10001 ; \dots ; a_n \cdot 1000n]$$

Ainsi en effectuant un poids sur les coefficients, les variations de ceci sont adaptées à leur ordre.

Nous gardons cependant le même critère. Pour conserver l'expression : $ecart(k) = \underline{K}' \cdot \underline{X} - Courbe(k)$, nous devons modifier le vecteur \underline{K} en : $\underline{K} = [1 ; k/1000 ; k^2/1000^2]$

Avec ce changement le résultat obtenu est bien plus correct :

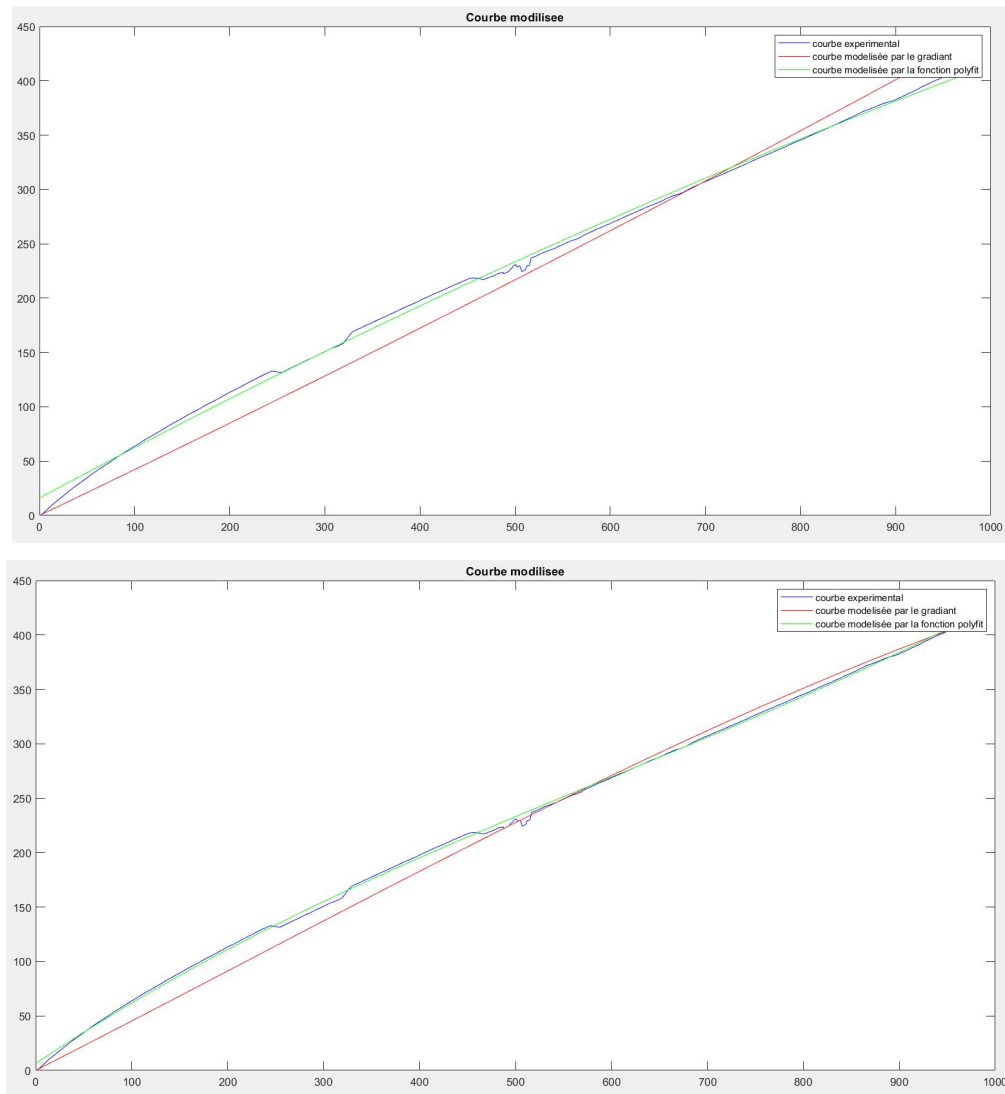


Figure 19 : Courbe approximée pour un polynôme d'ordre 2, puis 3 (de haut en bas). En bleu la courbe de base. En rouge la courbe approximée par la méthode du gradient. En vert la courbe approximée par polyfit().

Comme on peut l'observer, notre algorithme permet d'obtenir des résultats assez corrects. Pour les courbes approximées à l'ordre 2 aucun des deux algorithmes permettent d'obtenir une approximation satisfaisante (les variations proches du centre sont bien représentées). A l'ordre 3, l'approximation du gradient n'est toujours pas satisfaisante, l'écart pour les faibles indices est trop important. L'augmentation de l'ordre ne permet pas à notre algorithme d'obtenir un meilleur résultat. Cependant le résultat donnée par polyfit(), lui, est satisfaisant dès l'ordre 3.

C'est donc le résultat donné par polyfit que nous utiliserons par la suite.

Maintenant l'approximation de la courbe effectuée, nous pouvons enfin passer à la simulation de la trajectoire de la bille.

La fonction permettant l'approximation de la courbe de déformation est :

$[X, Xth] = \text{ModCourbGrad}(\text{Courbe}, n, \text{numfigure});$

Elle prend en argument : *Courbe* représentant la courbe de déformation, *n* l'ordre du polynôme d'approximation, *numfig* le numéro de la figure sur laquelle la fonction peut afficher.
 La fonction retourne en sortie les vecteurs X et X_{th} , respectivement le vecteur de coefficient de la méthode du gradient, et le vecteur de coefficient de polyfit.

3.5. Simulation de la trajectoire de la bille

Pour éviter que la simulation soit trop longue, elle s'arrête à 10000 itérations ou lorsque la bille atteint le centre (modélisée par un cercle représentant le poids placé au centre).

Pour simplifier les calculs, l'origine de la simulation est placée au centre de l'univers. Ainsi, on peut noter la position de la bille par un angle de position *alpha_pos*, et sa distance avec le centre *distance_cb*.

Une constante essentielle pour la simulation est le pas temporel notée *dt*. Plus il est petit et plus la simulation sera précise. Cependant, pour notre application, il n'est pas nécessaire d'obtenir une simulation extrêmement précise. Ainsi, nous fixons ce pas temporel à : $dt = 0.01$

Comme expliqué plus haut, nous avons besoin de la dérivée de la courbe de déformation. Puisque nous l'avons approximé par un polynôme. Il devient très facile de calculer la dérivée de la déformation. Il suffit de modifier les coefficients du polynôme.

Dans ce cas, le calcul de la norme de l'accélération liée au poids de la bille s'écrit ainsi :

$$a_norme = g * \sin(\text{atan}(1./\text{polyval}(Pc_d, \text{distance_cb} - \text{taille_disk})));$$

avec *g* une constante, *Pc_d* les coefficients du polynôme dérivée, et *taille_disk* la taille du disque en centre de l'univers.

Pour obtenir la dérivée de la courbure, on évalue le polynôme dérivée par *distance_cb - taille_disk*, puisqu'il faut prendre en compte la présence du disque non pris en compte dans la déformation du tissu.

L'accélération liée au poids de la bille est en direction du centre de l'univers. Ainsi par rapport à la position de la bille, cette direction est opposée à l'angle de position de la bille dans l'univers. La direction de l'univers est donc *alpha_pos - pi*.

Pour ce qui est de l'autre accélération, il suffit de prendre l'opposé de la vitesse avec un facteur *f*.

Ainsi l'itération de la position selon les axes *x* et *y* s'écrit comme suit:

$$X = [X; X(i-1) + vx * dt + (a_norme * \cos(\text{alpha_pos} - \pi) - vx * f) * dt * dt / 2];$$

$$Y = [Y; Y(i-1) + vy * dt + (a_norme * \sin(\text{alpha_pos} - \pi) - vy * f) * dt * dt / 2];$$

Une fois la position itérée, il faut mettre à jour les différentes variables. Tout d'abord, la vitesse à partir des positions :

$$vx = (X(i) - X(i-1))/dt;$$

$$vy = (Y(i) - Y(i-1))/dt;$$

Ensuite, la distance entre la bille et le centre, en prenant la norme de la position de la bille :

$$distance_cb = \sqrt{X(i)^2 + Y(i)^2};$$

Et pour finir, l'angle de position de la bille par un calcul de arctan selon la valeur de x.

Voici les résultats qu'on obtient avec les valeurs initiales suivantes :

$distance_cb = 1.8$; $alpha_pos-pi = \pi$; $vx = 0.1$; $vy = -2$; $f=0.8$

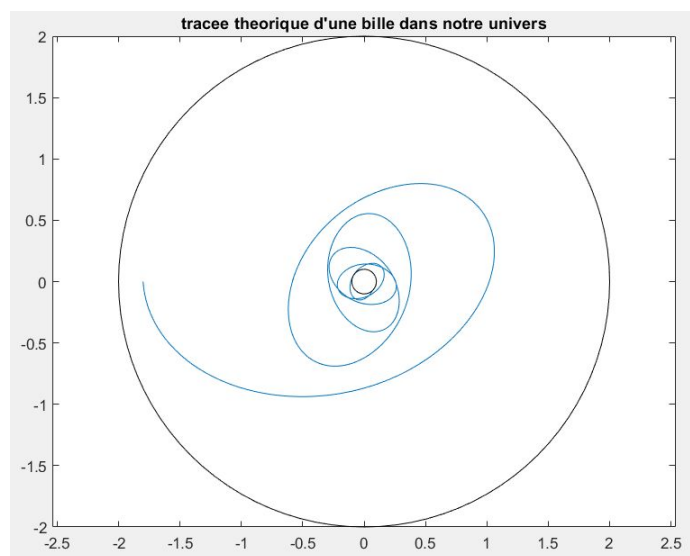


Figure 20 : tracée théorique de la bille dans l'univers

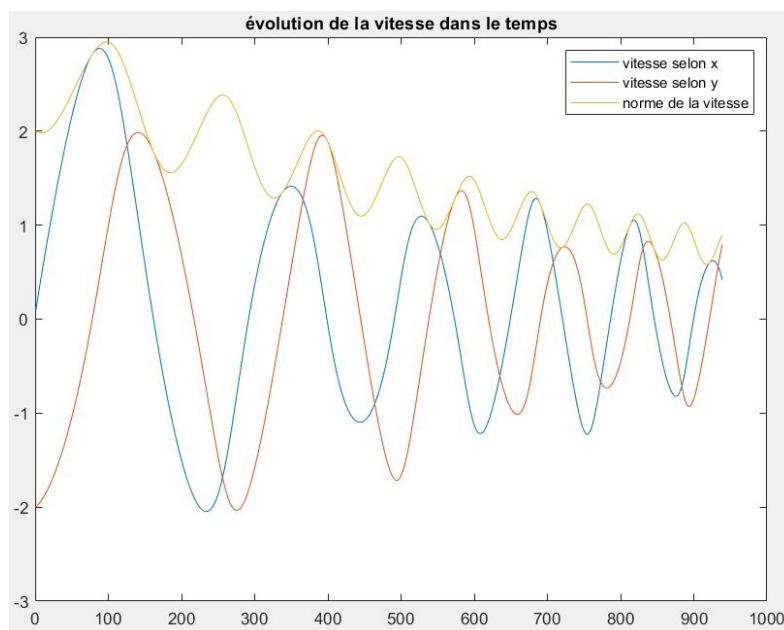


Figure 21 : Evolution de la vitesse

Le résultat est plutôt réaliste. On peut facilement supposer une trajectoire de bille comme celle-ci dans notre univers. La figure de la vitesse nous montre clairement l'effet du facteur f , par la décroissance progressive de la vitesse.

Ce tracé de trajectoire théorique va permettre de le comparer au tracé réel de la bille. Ainsi, nous pourrons valider ou non les différentes hypothèses qui ont été utilisées dans ce modèle théorique. Et dans le cas où ce modèle est suffisamment réaliste, nous pourrons déterminer les valeurs de notre modèle permettant d'obtenir les trajectoires expérimentales.

L'étape suivante est donc d'extraire d'une vidéo la trajectoire de la bille.

La fonction permettant d'effectuer cette simulation est:

TrajectoireTh($P_coubure$, $distance_cb_0$, $alpha_pos_0$, $vx0$, $vy0$, f , r_px_m , $taille_disk$, $numfigure$);

Elle prend en argument, $P_coubure$ les coefficients du polynôme de la courbure du tissu, $distance_cb_0$ la distance initiale entre la bille et le centre, $alpha_pos_0$ l'angle initial entre la bille et le centre, $vx0$ la vitesse initiale selon x, $vy0$ la vitesse initiale selon y, r_px_m le rapport entre pixel et mètre, $taille_disk$ la taille du disque central, $numfig$ le numéro de la figure sur laquelle la fonction peut afficher.

VI. Trajectoire expérimentale

1. Estimation de trajectoire par le filtre de Kalman

Tout d'abord, le filtre de Kalman est un filtre récursif. Cela nous permet de faire un traitement en temps réel, comme on a indiqué dans le cahier de charge : le traitement en temps réel est une de notre objectif. Il est aussi adaptatif, les non-stationnarités peuvent être directement prise en compte dans le modèle afin de poursuivre les évolutions du système étudié. Et finalement, il est optimal c'est-à-dire la solution qu'on trouve avec ce filtre minimise l'erreur quadratique moyenne. Grâce à toutes ces propriétés, le filtre de Kalman est une bonne méthode pour l'estimation de la trajectoire de la bille.

Dans cette application, l'algorithme de Kalman va être utilisé pour effectuer une prédiction sur la position de la bille. Une fois cette prédiction faite, il suffira de faire un traitement d'image autour de cette position prédite pour en déduire la position réelle de la bille. Ainsi, une fois cette observation faite, la position réelle de la bille est connue et est réinjectée dans l'algorithme de Kalman.

1.1. Équation d'état, équation d'observation et modèle physique

Le filtre de Kalman suppose le véritable état au moment $k+1$ est évolué à partir de l'état en k selon l'équation d'état :

$$X(k+1) = A(k)X(k) + B(k)U(k) + G(k)W(k)$$

où,

- $X \in R^n$ est le vecteur d'états
- $U \in R^q$ est le vecteur de commande
- $W \in R^p$ est le vecteur de bruit du système
- $A \in R^{n \times n}$ est la matrice de transition du système
- $B \in R^{n \times q}$ est la matrice de commande
- $G \in R^{n \times p}$ est la matrice de perturbation

On a quelques hypothèses sur l'équation d'états :

- Les processus mis en causes X , U et W sont réguliers au second ordre
- Le processus de perturbation W est centré et blanc :

$$\text{cov}(W(k), W(l)) = E[W(k)W(l)^T] = Q(k)\delta_{k,l}$$

où $Q(k)$ représente la variance de bruit à l'instant k .

Dans notre étude, comme on veut estimer la trajectoire, on s'intéresse à trouver la position de la bille et sa vitesse. Alors, notre vecteur d'états est formé par les coordonnées x , y de la bille et ses vitesses par rapport à ces axes de coordonnées. Notre système peut s'expliquer par les équations suivantes :

$$x(k+1) = x(k) + v_x(k) \times dt \quad (1)$$

$$v_x(k+1) = v_x(k) \quad (2)$$

$$y(k+1) = y(k) + v_y(k) \times dt \quad (3)$$

$$v_y(k+1) = v_y(k) \quad (4)$$

Ici, $X \in R^4$, $A \in R^{4 \times 4}$ et $G \in R^{4 \times 1}$. Comme il n'y a pas de processus de commande, on peut écrire notre équation d'état de la forme suivante :

$$X(k+1) = AX(k) + GW$$

où,

$$X = \begin{pmatrix} x(k) \\ v_x(k) \\ y(k) \\ v_y(k) \end{pmatrix} \quad A = \begin{pmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

W est le bruit blanc centré de variance Q. La variance est importante dans notre modèle car quand on définit notre vecteur d'états, on n'a pas pris en compte l'accélération.

Maintenant, on écrit l'équation d'observation sous forme :

$$Y(k) = C(k)X(k) + V(k)$$

où,

- $Y \in R^m$ est le vecteur des observations
- $C \in R^{m \times n}$ est la matrice de l'observation
- $V \in R^m$ est le bruit d'observation, supposé centré et blanc

$$\text{cov}(V(k), V(l)) = E[V(k)V(l)^T] = R(k)\delta_{k,l}$$

où R(k) représente la variance de bruit à l'instant k.

Enfin, on veut observer la position de la bille donc on fait les choix suivante pour la matrice de d'observation et définit l'équation d'observation de façon suivante :

$$Y(k) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x(k) \\ V_x(k) \\ y(k) \\ V_y(k) \end{pmatrix} + V(k)$$

La variance de V(k) et R(k) sont assez faibles car on suppose que la position observée est la véritable position.

1.2. Mise en place de l'algorithme de Kalman appliqué au suivi de la bille

Comme expliqué plus haut, l'observation de la position de la bille se fait à partir de la position prédite par l'algorithme de Kalman. L'idée est donc de faire un zoom de l'image autour de la position prédite. Ce zoom va réduire l'image à une image de dimension : $tF_{zoom} * tF_{zoom}$.

La bille doit cependant se trouver sur cette image zoomée. Dans le cas contraire, cela signifie que la prédiction de l'algorithme de Kalman était mauvaise.

On suppose la bille dans cette zone. Pour détecter la présence de la bille et déterminer le centre de la bille, nous détectons les contours de la bille et en faisons une moyenne.

La détection des contours se fait à partir de deux masques de détection l'un selon x, et l'autre selon y :

$$G_x = [-1 \ 2 \ -1];$$

$$G_y = [-1 \ ; \ 2 \ ; \ -1];$$

Après un seuillage, on obtient l'image suivante :

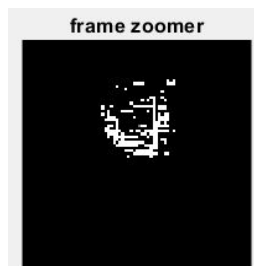


Figure 22 : Contour de la bille dans l'observation de Kalman

Il suffit maintenant de faire une moyenne de tous ces points pour déterminer la coordonnée de la bille. Cette coordonnée est ensuite injectée dans le vecteur d'observation Y, pour effectuer la mise à jour et la prédiction de Kalman. Puisque cette coordonnée est supposée comme étant la position réelle de la bille, la valeur de la matrice R est très faible. L'observation est une faible variance.

R utilisé est de l'ordre de : $0.0001 * [1 \ 0; 0 \ 1]$;

Les équations de mise à jour de Kalman permettent de mettre à jour les matrices X et P, en prenant en compte l'observation effectuée Y. Et ainsi de corriger au fur et à mesure les prédictions.

```
Mn = P*C'/(C*P*C'+R);           %gain de Kalman
x = x + Mn*(double(y(:,m)) - C*x); % x[n|n]
P = (eye(4)-Mn*C)*P*(eye(4)-Mn*C)' + Mn*R*Mn'; % P[n|n]
```

Une fois les mises à jour faites, l'algorithme réalise la prédiction à partir notamment de la matrice A, reprenant de cette application les équations du mouvement.

```
x = A*x ;                          % x[n+1|n]
P = A*P*A' + G*Q*G'; % P[n+1|n]
```

Une fois la prédiction faite, on peut recommencer en effectuant une observation autour de la position prédite par X, re-mettre à jour, re-prédire ...

1.3. Résultat de la méthode

Le principe de la méthode fonctionne : le traitement de l'image est correctement effectué et l'observation est juste. L'algorithme de Kalman prédit une nouvelle position de la bille cohérente. Mais l'algorithme de Kalman n'arrive pas à s'adapter suffisamment rapidement. La bille n'est plus détectée dans l'image zoomée, l'observation ne peut plus avoir lieu. Et l'algorithme s'arrête.

Voici le début de trajectoire que nous arrivons à tracer, ainsi que la vitesse estimée et mesurée:

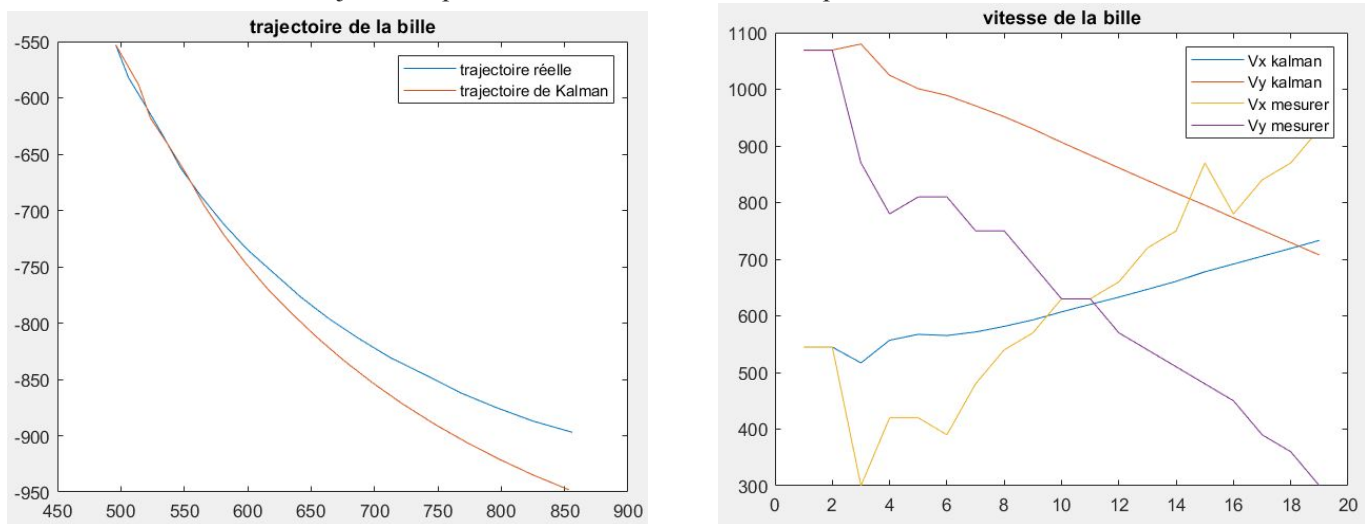


Figure 23 : Trajectoire déterminée par Kalman et mesurée. Vitesse déterminée par Kalman et mesurée.

Comme on peut le voir sur la trajectoire de la bille, l'algorithme arrive à suivre correctement la bille au début. Mais la bille change trop rapidement de direction et l'algorithme n'arrive pas à s'adapter suffisamment rapidement. On observe aussi le manque d'adaptivité de l'algorithme avec la vitesse mesurée et prédite par Kalman. Les vitesses de Kalman se dirigent vers la bonne direction mais n'évoluent pas assez vite.

Pour améliorer l'adaptativité de l'algorithme, on peut rajouter dans le vecteur observateur Y les informations sur la vitesse. Ainsi, la méthode devrait avoir une meilleure adaptativité sur la vitesse. Parmi les diverses modifications sur les matrices, la matrice de variance R est modifiée ainsi :

$$R=[0.0001 \ 0 \ 0 \ 0 ; 0 \ 5 \ 0 \ 0 ; 0 \ 0 \ 0.0001 \ 0 ; 0 \ 0 \ 0 \ 5];$$

Les variances sur l'observation de la position sont toujours aussi faibles, mais celles sur la vitesse est de l'ordre de 5 puisque la mesure de celle-ci est trop approximative.

Voici les nouveaux résultats obtenus :

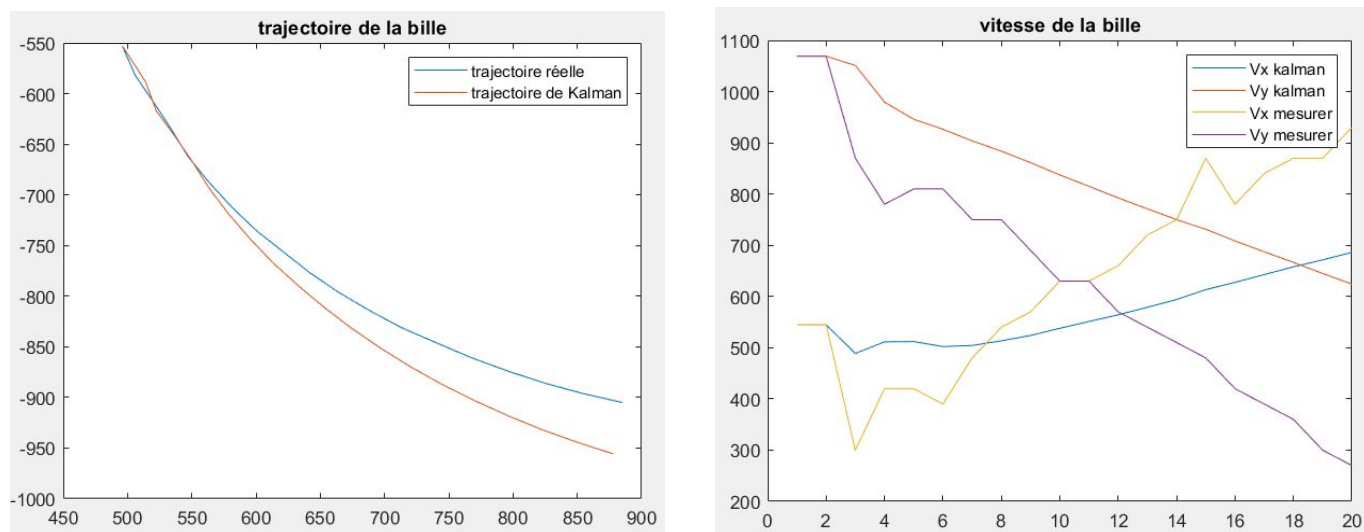


Figure 24 : Trajectoire déterminée par Kalman et mesurée, avec observation sur la vitesse. Vitesse déterminée par Kalman et mesurée, avec observation sur la vitesse.

Une fois encore l'algorithme ne s'adapte pas assez vite. Il semble mieux suivre la bille que précédemment.

Une solution pour pallier à ce problème sera d'augmenter la fenêtre de zoom. De cette manière, l'algorithme arrive à suivre plus longtemps la bille, mais l'algorithme aurait tout de même du mal à s'adapter. En augmentant cette fenêtre, il faudrait aussi modifier la méthode de détection de la bille, puisqu'avec cette méthode, on risque de s'approcher trop près du bord de l'univers, et donc de mal détecter la position de la bille.

Une autre solution pour qu'il puisse s'adapter de manière plus efficace serait d'augmenter le nombre d'image par seconde. Dans ce cas, cela permettrait d'avoir des changements de direction plus douces, et donc de laisser plus de temps à l'algorithme de s'adapter. Cependant, cela nécessite une nouvelle caméra, ou de pouvoir prendre de nouvelle vidéo de trajectoire, ce qui n'est pas possible.

En outre, si l'algorithme n'arrive pas à s'adapter assez vite : c'est parce qu'il n'a pas de connaissance, de prise en compte des changements de direction de la bille lors de la prédiction. Il faudrait ainsi modifier la matrice A pour qu'elle prenne en compte l'accélération ou du moins une attraction éventuelle vers le centre de l'univers.

Le tracée de la trajectoire de la bille ayant échoué par cette méthode, on a développé une méthode de suivi plus simple, mais plus coûteuse en calcul.

2. Estimation de trajectoire par la méthode de différentiation

Cette fois, pour obtenir la trajectoire de la bille, on a utilisé une autre approche. L'idée est de partir d'une image référente, où la bille ne se trouve pas sur le tissu. Lors du parcours de la vidéo image par image, l'image référente est soustraite à l'image parcourue. En effet, la bille ne se trouve pas sur le tissu sur l'image référente, mais se trouve sur l'image parcourue. De plus, c'est (quasiment) la seule chose qui bouge lors de la vidéo. Une fois la soustraction effectuée, un seuillage est effectué pour mettre en évidence les soustractions de couleur les plus susceptibles d'être la position de la bille. Puisque la bille se trouve sur le tissu et que la bille est noire, la soustraction des images va placer à la position de la bille la couleur du tissu. La connaissance de l'histogramme de couleurs du tissu permet donc de choisir le seuil optimal (voir figure 3). On a alors effectué un seuillage sur la couleur rouge à un niveau de 150.

Voici l'image référente :



Figure 25 : Image référente

Cette méthode est simple et efficace mais suppose que par rapport à l'image référente seule la bille bouge. Ce qui n'est évidemment pas le cas. Parmi les objets qui bougent lors de la vidéo, on retrouve le bout du lanceur, les épingles du bas, et quelques éléments à l'extérieur de l'univers. Pour éviter que ces objets ne perturbent trop la détection de la position de la bille, plusieurs solutions ont été testées.

La première idée est de redimensionner les images. En effet, toute la partie gauche des images ne correspond pas à l'univers. Cependant cette méthode est coûteuse en calcul.

Toujours dans l'idée de supprimer ces zones compromettantes, le placement artificiel de zone noire permet de ne pas prendre en compte les mouvements dans ces zones.

Finalement, la solution la plus efficace et retenue est de repérer la position de la bille en effectuant une convolution de l'image avec la forme de la bille, c'est-à-dire un disque de rayon 8 pixels. Ainsi, même s'il ressort d'autres objets que la bille dans l'image, la convolution va mettre en évidence les formes ressemblant à un disque. La valeur maximale de l'image après convolution doit donc être la position de la bille. Le défaut de cette solution est qu'elle est coûteuse.

L'élimination de la partie gauche des images sera tout de même effectuée pour éviter que des objets trop grands dans l'image soient détectés comme étant la bille.

Voici le type d'image obtenu après la soustraction et seuillage, et après convolution :

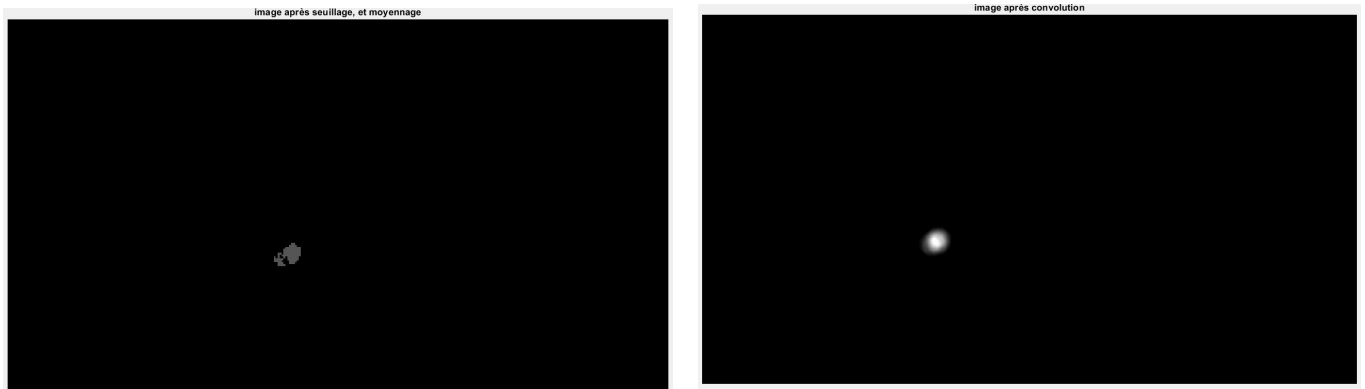


Figure 26 : Image après seuillage (à gauche) et après convolution (à droite). Zoom effectué autour de la bille.

L'image après seuillage montre que la soustraction permet en effet de mettre en évidence la bille. La convolution permet de mettre encore plus en évidence la bille, mais donne aussi une valeur maximum correspondant au milieu de la bille. Ainsi la coordonnée de la bille se détermine facilement. Après normalisation de l'image, le bout de code suivant permet de retrouver l'indice et puis sa coordonnée :

```
coord = find(thisFrame==1);
coord = [ mod(coord,size(thisFrame,1)) int32(coord/size(thisFrame,1))];
```

La coordonnée de la bille est ensuite stockée dans un vecteur, et on peut passer à l'image suivante.

Une fois toutes les images parcourues, on a effectué un prétraitement sur la liste des coordonnées pour éliminer les coordonnées aberrantes. La liste est parcourue en calculant la différence entre deux points consécutifs. Si la différence est trop importante (jugée ici supérieur à 0.1) la coordonnée est jugée comme aberrante, et prend la valeur précédente. Ce traitement est efficace mais pose problème lors du calcul de la vitesse expérimentale de la bille. En effet, on obtient artificiellement des valeurs de vitesse nulle, puisque les coordonnées ne sont pas modifiées, et des pics de vitesse, puisque la variation après les valeurs nulles sont plus fortes. Ainsi, on a effectué un second traitement sur la vitesse où les valeurs nulles sont retirées ainsi que les valeurs suivantes.

Voici les résultats de trajectoire et de vitesse expérimentaux obtenus :

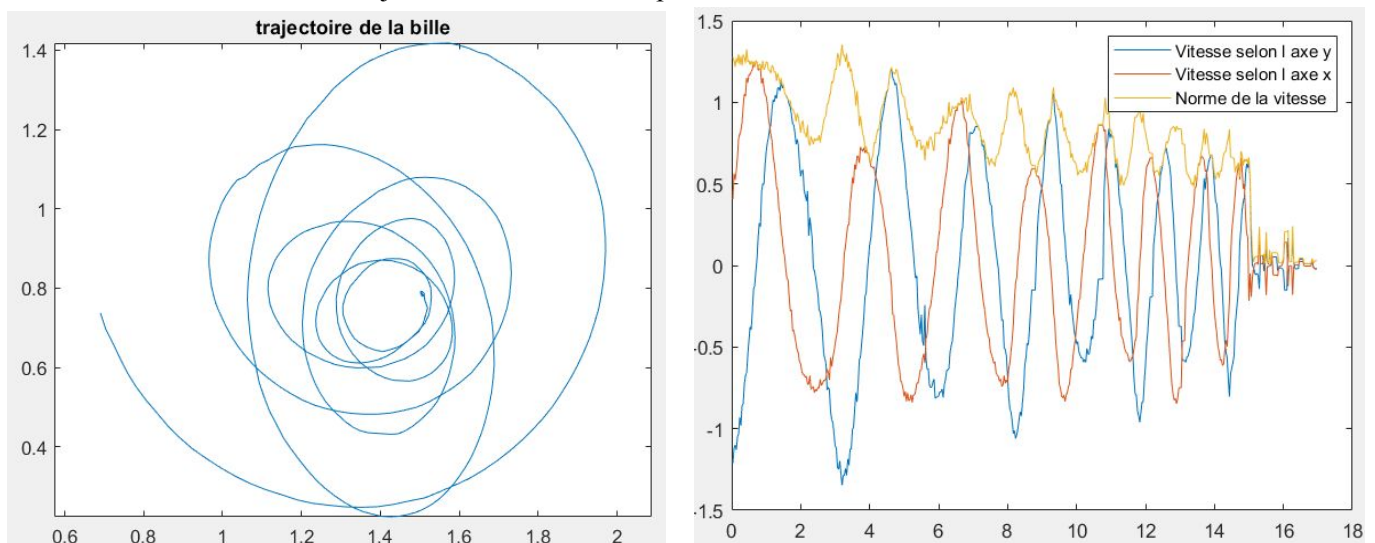


Figure 27: Trajectoire de la bille. Vitesse de la bille.

Malgré de légers défauts, la trajectoire de la bille est correctement extraite. La vitesse calculée met en évidence ces défauts, mais on reconnaît les formes sinusoïdales liées au poids de la bille et de la déformation du tissu, ainsi que la décroissance progressive liée au frottement.

Maintenant que la trajectoire de la bille a été extraite, la comparaison entre le modèle théorique et l'expérimentation est possible.

VII. Comparaison des resultats theoriques et experimentales

Dans cette partie de rapport on voulait comparer les résultats expérimentaux avec les résultats théoriques qu'on a étudié.

1. Verification des lois de Kepler

Comme on détaillé dans la partie "trajectoire théorique : force gravitationnelle", la première loi de Kepler nous dit que le trajectoire planétaire est de forme elliptique pour une excentricité entre 0 et 1. Quand on regarde les formes de trajectoire qu'on a trouvé dans la figure 27, même s'il a une forme quasi-elliptique, on ne peut pas dire que cette loi est vérifiée. Cela vient de l'imperfection de notre système. La force frottement entre la bille et le tissu n'a pas été prise en compte dans notre étude et cette force négligée dans le calcul a faussé nos résultats.

La deuxième loi de Kepler explique que la vitesse de la planète augmente lorsqu'elle s'approche du Soleil et diminue lorsqu'elle s'éloigne du Soleil. Dans notre expérience, on observe que l'accélération augmente quand la bille s'approche de la masse centrale mais tout de même, on ne peut pas dire que cette loi est vérifiée car la loi est valable pour une trajectoire elliptique qui a une forme symétrique. Mais la trajectoire de la bille change à chaque période de temps et perd les propriétés pour appliquer la deuxième lois de Kepler.

Pour la troisième loi de Kepler, on regarde le rapport entre la taille de l'orbite d'une planète et sa période sidérale. Mais comme on n'a pas pu vérifier les deux premières lois de Kepler à cause des raisons qu'on a précisé au-dessus, on n'a pas pu approfondir nos études pour vérifier le troisième loi. On voulait trouver une constante gravitationnelle de notre univers mais à cause du manque de temps et de vidéo sur lesquelles on voulait faire l'expérience, on n'a pas pu trouver cette valeur.

2. Vérification du modèle théorique

Le théorique qui se base sur la déformation du tissu, et la modélisation de l'univers en un cercle permet d'obtenir des trajectoires de bille à partir de diverses conditions initiales.

Voici la trajectoire qu'on obtient avec les conditions initiales suivantes :

$distance_cb = 1.8$; $alpha_pos = 11.\pi/10$; $vx = 1$; $vy = -3$; $f = 0.7$

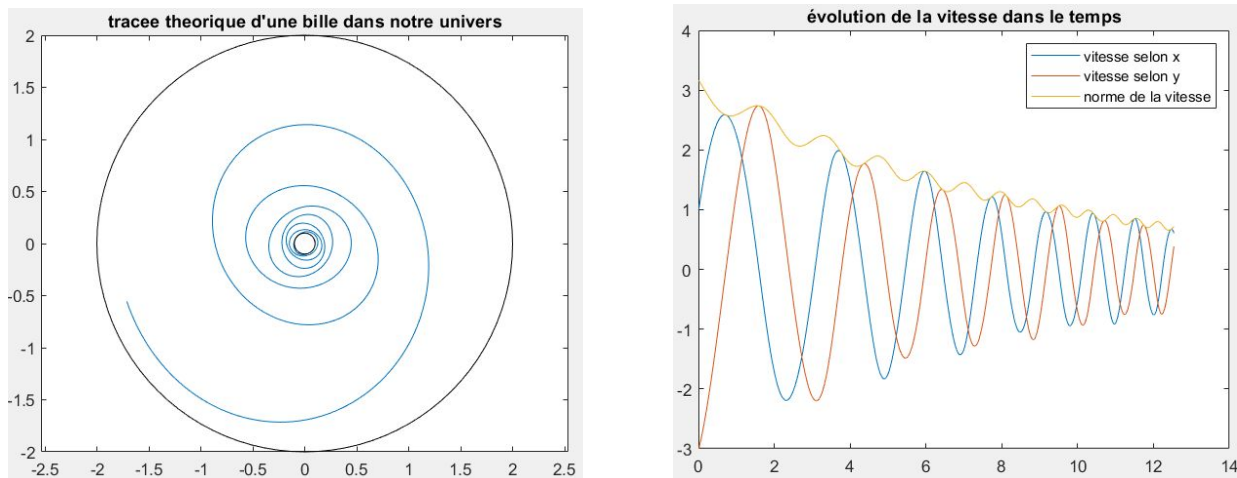


Figure 28: Trajectoire théorique, et vitesse théorique

Avec ses paramètres, nous obtenons une trajectoire qui imite au mieux la trajectoire mesurée expérimentalement (figure 27). En effet, les conditions initiales sont identiques (à un facteur 2 près, liées au rapport pixel/mètre utilisée). Et pour les premières valeurs, la trajectoire est identique. C'est une fois que la bille passe dans la partie gauche de l'univers que la trajectoire commence à différer.

Cette différence de trajectoire provient de l'hypothèse de départ sur la modélisation de l'univers comme étant un cercle. Cela permettant de faire du traitement d'image sur une seule photo et d'extrapoler sur l'ensemble de l'univers. Mais lorsqu'on observe la trajectoire expérimentale, on remarque que la bille forme un cercle au début de la trajectoire, et non une ellipse. Cela permet de dire que la bille semble subir une plus faible attraction vers le centre. Cela indique que la déformation de l'univers est plus faible à cette endroit. Cette remarque est cohérente avec les formes elliptiques de l'univers. En effet, notre univers peut être modélisé par une ellipse, avec un demi-grand axe horizontal (par rapport à la vu du dessus), et un demi-petit axe vertical. Ainsi avec ce modèle, la déformation du tissu est plus importante selon le petit axe que selon le grand axe. Or pour notre modèle théorique, les photos ont été prises selon le petit axe. Ainsi, en considérant maintenant l'univers comme une ellipse, notre modèle surévalue la déformation du tissu.

On en conclut que l'hypothèse de départ considérant l'univers comme un cercle est faux. Il faudrait reprendre notre modèle en considérant l'univers comme une ellipse. Il faudrait ainsi utiliser deux photos de la déformation du tissu: l'un sur le demi-grand axe, l'autre sur le demi-petit axe. Et ensuite, effectuer une extrapolation sur le reste de l'univers.

Malgré ce problème d'hypothèse, nous pouvons tout de même valider les autres hypothèses de notre modèle qui permet d'obtenir des trajectoires de bille cohérente avec la réalité.

Pour effectuer une étude plus poussée et plus scientifique sur ce modèle, il faudrait pouvoir comparer le modèle à plus de trajectoire expérimentale.

VIII. Conclusion

Dans notre projet, il y avait plusieurs étapes où on a pris des décisions importantes, comme le design de setup et du lanceur, le choix de la méthode de détection et de prédiction de trajectoire etc. Certains de ces choix étaient performants et certains ne l'étaient pas. Pour le design de setup, comme on ne voulait pas perdre beaucoup de temps sur la manufacture et l'achat des pièces, on a utilisé beaucoup de pièces des groupes précédents. Cela

n'a pas vraiment influencé notre étude puisqu'on a pu prendre une bonne vidéo, et on a réalisé le traitement souhaité. Pour le design de lanceur, on a passé beaucoup de temps, et avons réussi à créer quelque chose qui nous donne la flexibilité de faire les études qu'on voulait.

Le modèle théorique de notre univers a permis de simuler la trajectoire d'une bille, et donc de pouvoir comparer cette trajectoire avec la réalité, pour pouvoir valider les hypothèses du modèle

Pour la détection et le suivi des trajectoires, on a utilisé une des méthodes très utilisées dans plusieurs domaines : le filtre de Kalman. Mais malheureusement, on n'a pas pu obtenir des résultats satisfaisants. Si on pouvait avoir la chance de refaire notre expérience avec une caméra qui a une fréquence d'image plus élevée, on aurait eu de meilleurs résultats. Avec la méthode différentiation, on a pu avoir de bons résultats. Mais comme on a pu prendre qu'une seule vidéo, le résultat n'est pas représentatif pour toutes les expériences possibles. On ne peut pas être sûr que cette méthode soit généralisable pour d'autres vidéos. Enfin, comme on a indiqué dans le cahier de charge, on voulait faire un traitement en live mais dans la conditions actuelle cet objectif était infaisable.

Pour finir, ce projet nous a permis d'appliquer les éléments des cours sur un exemple concret. Cela nous a donné la possibilité d'approfondir nos connaissances sur les filtrage, le traitement d'image et de calcul scientifique.

Bibliographie

- "Kepler's Laws of Planetary Motion." *Wikipedia*, Wikimedia Foundation, 23 Apr. 2020, en.wikipedia.org/wiki/Kepler%27s_laws_of_planetary_motion.
- Cours de Filtrage, filtrage adaptatif, filtrage de Kalman
- Cours de Traitement de l'image
- Documentation Matlab: "MATLAB." *MATLAB Documentation - MathWorks France*, fr.mathworks.com/help/matlab/.

Annexes

Annexe 1: Scripte modélisation de l'univers

%%

%Ce scripte permet de modéliser la déformation du tissu à partir d'une

%photo prise en dessous de l'univers. Une simulation théorique de la

```

%trajectoire d'une bille est ensuite effectuée, à partir de condition
%initial
%
%
%%
clear all;
clc;
close all;
%lecture de l'image
Im = imread("moy_1.bmp");
%inversion de l'image
Im_inv=0*Im;
for i= 1:length( Im(1,:) )
    Im_inv(:,i,:) = Im(:, length(Im(1,:))-i+1 ,:);
end
figure(1)
imshow(Im_inv)
title('image de base');
Im=Im_inv;
%detection de la deformation de l'image
Courbe = DetectionCourbure( Im, 30, 2);
%prétraitement de la courbe symbolisant la déformation du tissu
CourbeMoy = pretraitement(Courbe, 30, 10, 4);
%récupération de la partie utile de la courbe
NCourbe = CourbeApresDisk(CourbeMoy, 0.4, 100, 5);
%modélisation de la courbure par la méthode du gradient
[X,Xth] = ModCourbGrad( NCourbe, 4, 6);
%simulation de la trajectoire théorique dans notre univers
TrajectoireTh(Xth, 1.8, pi, 0.1, -2, 0.8, 970/0.752, 0.1, 7);

```

Annexe 2: fonction détection de la courbure

```

function Courbe = DetectionCourbure( Im, seuil_detection, numfigure)
%%
%fonction permettant de detecter les contours de la déformation du tissu.
%
%
%Im: image sur laquelle il faut travailler
%
%seuil_detection: seuil à partir du quelle on considère que le point est le
%contour du tissu ou non
%
%numfigure: donne le numéro de la figure sur laquelle on peu dessiner
%
%%
%lissage de la photo
masque = fspecial('disk',3);%[1/8 1/8 1/8 ; 1/8 1/8 1/8 ; 1/8 1/8 1/8];
Im = imfilter(Im, masque);
%masque de detection de contours

```



```

masque = [ [0;0;2;0;0] [1 ; 4 ; -14 ; 4 ; 1] [0;0;2;0;0] ];
Im = imfilter(Im, masque, 'symmetric' ); %detection des contours
%affichage de la photo de base
figure(numfigure);
imshow(Im);
title('photo après filtrage');
%initialisation des variables
lenH=length(Im(1,:,3));
Courbe=[];
J=Im;
for i= 1:lenH
    %on effectue une coupe
    coupe = Im(:,i,3);
    %on cherche l'endroit où il y a une forte variation
    n=15;
    while coupe(n) < seuil_detection && n<length(coupe) %espèce de seuillage
        n=n+1;
    end

    %on place un gros point rouge (5px * 5px) sur l'image J
    for j= -2:2
        for k= -2:2
            if i+k>0 && i+k<lenH-1
                J(n+j,i+k,1)=255;
                J(n+j,i+k,3)=0;
                J(n+j,i+k,2)=0;
            end
        end
    end
    %on remplit le vecteur
    Courbe=[Courbe(:) ; length(coupe)-n];
end
figure(numfigure+1);
imshow(J);
title('photo après détection des bords du tissu');
end

```

Annexe 3: fonction prétraitement

```

function MoyCourbe = pretraitement(Courbe, seuil_val_aberant, moy, numfigure)
%%
%fonction effectuant un prétraitement sur la courbe déterminée
%expérimentalement, enlève les points aberrants, et effectue un lissage
%
%
%Courbe: courbe expérimentale
%
%seuil_val_aberant: seuil déterminant si un point est considéré comme
%abérant ou non

```

```

%
%moy: valeur déterminant le nombre de point sur lequel on fait un lissage
%
%numfigure: numéro de la figure sur laquelle on affiche les courbes
%
%%
%affichage de la courbe de base
figure(numfigure);
plot(Courbe);
hold on
%On enlève les points abérents de la courbe
y=2;
while abs(y-length(mean(Courbe)/2)) < 300 %on commence avec une valeur cohérente au centre de l'image
    y = y + 1;
end
for i= y:length(Courbe)
    if abs(Courbe(i)-Courbe(i-1)) > seuil_val_aberant %si l'écart est supérieur au seuil
        Courbe(i) = Courbe(i-1); %on redonne la valeur précédente
    end
end
%affichage de la courbe sans les points abérants
plot(Courbe);
hold on
%lissage de la courbe
MoyCourbe = filter(ones(1,moy)/moy , 1 , Courbe);
MoyCourbe = MoyCourbe([moy/2:length(MoyCourbe)]);
%affichage de la courbe moyennée
plot(MoyCourbe);
legend('Courbe de base','Courbe sans les valeurs abérantes','Courbe moyennée');
title('Courbe prétraitment');
end

```

Annexe 4: fonction courbe après disque

```

function NCourbe = CourbeApresDisk(CourbeMoy, val_max, long_mini, numfig)
%%
%fonction permettant de déterminer la position après le disque, et de
%récupérer la courbe qu'à partir du disque
%
%CourbeMoy: courbe expérimental lisée
%
%val_max: valeur maximal où nous considérons le plateau correspondant le disque
%
%long_mini: longueur minimum où nous considérons le plateau correspondant le disque
%
%numfig: numéro de la figure sur laquelle nous affichons
%
%

```

```

%%
%Calcul de la courbe dérivée.
CourbeDerivee = filter([1 ; -1] , 1 ,CourbeMoy);
figure(numfig)
plot(CourbeDerivee);
title('derivee de la courbe experimental');
%détermination de la fin de du disque, et donc du début de la courbe de
%déformation du tissu
count=0;
i=1;
while i<length(CourbeDerivee) && count ~= -1
    if abs(CourbeDerivee(i)) < val_max
        count = count +1;
    else
        if count > long_mini
            count = -1;
        else
            count = 0;
        end
    end
    i=i+1;
end
%récupération de la partie utile de la courbe
NCourbe=CourbeMoy(i:length(CourbeMoy));
NCourbe = NCourbe - NCourbe(1); %normalisation
end

```

Annexe 5: fonction approximation de la courbure par un polynôme

```

function [X,Xth] = ModCourbGrad( Courbe , n, numfigure)
%%
%fonction modelisant une courbe par un polynome d'ordre n à partir de la
%méthode d'obtimation du gradient à pas fixe, avec comme critère à
%optimiser: l'erreur quadratique moyen (eqm)
%
%
%Courbe: vecteur de la courbe à modéliser
%
%n: ordre du polynome à modélisateur
%
%numfigure: numéro d'affichage de la figure
%
%%
%initialisation des paramètres
N = length(Courbe);
X = 0*ones(n,1);
rho = 1e-1;
for j= 1: 1000

```

%calcul du gradient de l'eqm, en 2 étape

```

ecart=[Courbe(1)];
for k= 1:N
    K=[];
    for i= 1:n
        K=[ K ; (k/1000)^i];
    end
    ecart = [ecart ; K'*X-Courbe(k) ];
end

grad= [];
for i= 1:n
    terme_haut = 0;
    for k= 1:N
        terme_haut = terme_haut + ((k/1000)^i)*ecart(k);
    end
    grad = [grad ; terme_haut/sqrt(ecart'*ecart) ];
end

%itération de la méthode
X = X - rho*grad;
end
%calcul de la nouvelle courbe obtenue
Newcourbe = [];
X = [ Courbe(1) ; X];
for k=1:N
    K=[];
    Kth=[];
    for i= 0:n
        K=[ K ; (k/1000)^i];
        Kth=[ Kth ; (k)^i];
    end
    Newcourbe = [Newcourbe ; K'*X ];
end

Xth = polyfit( [0:N-1]' , Courbe , n);
NewcourbeTh = polyval(Xth,[0:N-1]);
%affectation des bon coefficients
for i = 1:n
    X(i) = X(i)/( 1000^(i-1) );
end
%affichage de la courbe de départ, de la courbe obtenue par la méthode du
%gradient, et celle obtenu par la fonction polyfit
figure(numfigure);
plot(Courbe,'b');
hold on
plot(Newcourbe,'r');
hold on
plot(NewcourbeTh,'g');
legend('courbe experimental','courbe modelisée par le gradient','courbe modelisée par la fonction polyfit');

```

```

title('Courbe modilisee');
end

```

Annexe 6: fonction simulation de la trajectoire

```

function TrajectoireTh(P_coubure, distance_cb_0, alpha_pos_0, vx0, vy0, f,r_px_m, taille_disk, numfigure)
%%
%fonction permettant de tracer la trajectoire d'une bille à partir de la
%courbure du tissu

%P_coubure: vecteur contenant les coefficients polynomiaux de la modélisation
%      de la courbure du tissu, en pixel
%
%distance_cb_0: entier initialisant la distance initiale entre la bille et
%      le centre de l'univers, en mètre
%
%alpha_pos_0: entier initialisant l'angle de la position de départ de la
%      bille dans l'univers, en radian
%
%vx0: entier initialisant la vitesse initiale de la bille selon x, en
%      mètre/seconde
%
%vy0: entier initialisant la vitesse initiale de la bille selon y, en
%      mètre/seconde
%
%f: facteur de frottement tissu-bille
%
%r_px_m: rapport pixels / mètre
%
%taille_disk: rayon du disk central en mètre
%
%numfigure: numéro de la figure sur laquelle dessiner
%%
%calcul des coefficients dérivés du polynôme modélisateur
Pc_d = [];
for i = 2:size(P_coubure,1)
    Pc_d = [Pc_d (i-1)*P_coubure(i)];
end
%initialisation des composantes de l'univers
dt=0.01;
g=9.81*r_px_m;
taille_disk=taille_disk*r_px_m;
%initialisation des conditions initiales
vx=vx0*r_px_m;
vy=vy0*r_px_m;
V_x=[vx];
V_y=[vy];
distance_cb=distance_cb_0*r_px_m;
alpha_pos = alpha_pos_0;

```

```

X= [distance_cb*cos(alpha_pos) ];
Y= [distance_cb*sin(alpha_pos) ];
i=2;
while i<10000 && distance_cb>taille_disk
    %itération des variables
    a_norme = g*sin( atan(1./polyval(Pc_d,distance_cb-taille_disk)));
    X= [X ; X(i-1) + vx*dt + (a_norme*cos(alpha_pos-pi) - vx*f)*dt*dt/2];
    Y= [Y ; Y(i-1) + vy*dt + (a_norme*sin(alpha_pos-pi) - vy*f)*dt*dt/2];

    %mise à jour des variables
    vx = (X(i) - X(i-1))/dt;
    vy = (Y(i) - Y(i-1))/dt;

    V_x=[V_x vx];
    V_y=[V_y vy];

    distance_cb = sqrt( X(i)^2 + Y(i)^2 );

    %calcul de l'angle
    if X(i) >0
        alpha_pos = atan( Y(i)/X(i) );
    else
        if X(i) ==0
            if Y(i) > 0
                alpha_pos = pi/2;
            else
                alpha_pos = -pi/2;
            end
        else
            alpha_pos = pi + atan( Y(i)/X(i) );
        end
    end

    i = i +1;
end

%affichage des figures
figure(numfigure)
plot(X/r_px_m,Y/r_px_m)
rectangle('position', taille_disk/r_px_m*[-1 -1 2 2], 'curvature', 1);
rectangle('position', 2*[-1 -1 2 2], 'curvature', 1);
axis( 'equal');
title("tracee theorique d'une bille dans notre univers");
figure(numfigure+1)
plot((1:length(X))*dt,X/r_px_m)
hold on
plot((1:length(Y))*dt,Y/r_px_m)
legend('position selon x','position selon y');

```

```

title("évolution des positions selon x et y");
figure(numfigure+2)
plot((1:length(V_x))*dt,V_x/r_px_m)
hold on
plot((1:length(V_x))*dt,V_y/r_px_m)
plot((1:length(V_x))*dt, sqrt((V_x/r_px_m).^2 + (V_y/r_px_m).^2) )
legend("vitesse selon x","vitesse selon y","norme de la vitesse");
title("évolution de la vitesse dans le temps");
end

```

Annexe 7: Scripte détection de la trajectoire par l'algorithme de Kalman

```

%%
% Ce script permet de tracer la trajectoire d'une bille à partir d'une
% video. Il utilise l'algorithme de Kalman
%
%%
clear all;
close all;
%lecture de la video
v= VideoReader('1ereTrajectoireLow.mp4');
numberOfFrames = v.NumFrames;

%initialisation des matrices pour l'algorithme de Kalman
Q=0.1;
R=[0.0001 0 0 0 ; 0 5 0 0 ; 0 0 0.0001 0 ; 0 0 0 5];%0.0001*[1 0 ; 0 1]; %[1 0 0 0 ; 0 0.1 0 0 ; 0 0 0.1 0 ; 0 0 0 1];
dt=v.duration/numberOfFrames;
A=[1 dt 0 0; 0 1 0 0; 0 0 1 dt; 0 0 0 1];
G= [1; 1; 1; 1];
C= eye(4);%[1 0 0 0 ; 0 0 1 0]; %eye(4);
P = eye(4); %G*Q*G';
%initialisation de la position et de la vitesse de la bille
pos_x_init=496;
pos_y_init=553;
V= 1200;
Vx= [V*cos(-pi*63/180)];
Vy= [V*sin(-pi*63/180+pi)];

%initilisation du vecteur d'état et du vecteur d'observation
x = [pos_x_init; Vx ;pos_y_init;Vy];
Xmuseum = [x]; %sauvegarde du vecteur d'état
y = x;%[pos_x_init;pos_y_init]; %x;
%initilisation des paramètres de traitement de l'image
tFzoom =80;
start = 65;
Gx = [-1 2 -1];
Gy= [-1 ; 2 ; -1];
for frame = start:140
    m=frame-start+2;          %indice du vecteur y

```

```

%lecture de la frame
thisFrame = read(v, frame);

%zoom autour de la psition prédit
frameZoom= double( thisFrame(int16(x(3))-tFzoom/2:int16(x(3))+tFzoom/2 ,
int16(x(1))-tFzoom/2:int16(x(1))+tFzoom/2 , 1) );

%gradient de l'image zoomée
frameZoom = imfilter(frameZoom, Gx, 'symmetric' ).^2 + imfilter(frameZoom, Gy, 'symmetric' ).^2 ;

frameZoom = double(frameZoom>100); %seuillage

%détermination du centre de la bille
x_billeZ=0;
y_billeZ=0;
n=0;
for i = 1:tFzoom+1
for j = 1:tFzoom+1
if frameZoom(i,j)>0.5
x_billeZ = x_billeZ +j;
y_billeZ = y_billeZ +i;
n=n+1;
end
end
end

%affichage des données au cas où la bille est perdu
if n == 0
%affichage de l'évolution des paramètres

%affichage de la trajectoire selon Kalman, selon la réalité
figure(3)
plot(y(1,:),-y(3,:));
hold on
plot(Xmuseum(1,:),-Xmuseum(3,:));
legend('trajectoire réelle','trajectoire de Kalman');
title('trajectoire de la bille');

%affichage des vitesse
figure(4)
plot(Xmuseum(2,:))
hold on
plot(Xmuseum(4,:))
plot(Vx);
plot(Vy);
legend('Vx kalman','Vy kalman','Vx mesurer','Vy mesurer');
title('vitesse de la bille');
end

```



```

%mise à jour de la position de la bille dans l'image de base
x_bille = int16(x_billeZ/n) + int16(x(1)) -tFzoom/2;
y_bille = int16(y_billeZ/n) + int16(x(3)) -tFzoom/2;

%détermination expérimental de la vitesse
if frame ==start %pas d'évaluation dans pour la première image
vx_bille = Vx;
vy_bille = Vy;
else
vx_bille = (x_bille - y(1,m-1) )/dt;
vy_bille = (y_bille - y(3,m-1) )/dt;
end
%sauvegarde de la vitesse
Vx = [Vx vx_bille];
Vy = [Vy vy_bille];

%placement d'un point pour analyser le bon fonctionnement du programme
frameZoom(int16(y_billeZ/n),int16(x_billeZ/n)) = 0.5;
figure(2)
imshow(frameZoom);
title('frame zoomer');

%sauvegarde du vecteur y
% y =horzcat(y,[x_bille; y_bille]);
%dans le cas où y prend en compte la vitesse de la bille
y =horzcat(y,[x_bille ; vx_bille ; y_bille ; vy_bille]);

%placement d'un point pour analyser le bon fonctionnement du programme
thisFrame(y(3,m)-1:y(3,m)+1,y(1,m)-1:y(1,m)+1,1) =255;
figure(1)
imshow(thisFrame);
title('position de la bille');

%ralentissement du temps
pause(0);

% Mise a jour de l'algorithme de Kalman
Mn = P*C'/(C*P*C'+R); %Gain de Kalman
x = x + Mn*( double(y(:,m)) -C*x); % x[n|n]
P = (eye(4)-Mn*C)*P*(eye(4)-Mn*C)' + Mn*R*Mn'; % P[n|n]

% Prediction de l'algorithme de Kalman
x = A*x ; % x[n+1|n]
P = A*P*A' + G*Q*G'; % P[n+1|n]

%vecteur x entroposer
Xmuseum = [Xmuseum x];

```

end

```

%affichage de l'évolution des paramètres
%affichage de la trajectoire selon Kalman, selon la réalité
figure(3)
plot(y(1,:),-y(3,:));
hold on
plot(Xmuseum(1,:),-Xmuseum(3,:));
legend('trajectoire réelle','trajectoire de Kalman');
title('trajectoire de la bille');
%affichage des vitesses
figure(4)
plot(Xmuseum(2,:))
hold on
plot(Xmuseum(4,:))
plot(Vx);
plot(Vy);
legend('Vx kalman','Vy kalman','Vx mesurer','Vy mesurer');
title('vitesse de la bille');

```

Annexe 8: Scripte détection de la trajectoire par méthode de différentiation

```

%%
% Ce script permet de lire une video de trajectoire d'une bille, parcourir
% la video en retrouvant au fur et à mesure la coordonnée de la bille. Et
% donc de tracer la trajectoire de la bille, ainsi que sa vitesse.
%
%%
clear;
close all;
%initialisation du rapport pixel - metre
r_px_m = 0.752*2/1080;
%lecture de la video
v= VideoReader('1ereTrajectoireLow.mp4');
%initialisation du pas temporel
dt=v.duration/v.NumFrames;
FrameRef = read(v, 60); %extraction de l'image référence (bille en dehors du tissu)

%suppression des zones sensibles
% FrameRef(400:480,430:490,:)=0;
% FrameRef(920:1080,500:1500,:)=0;
FrameRef(:,1:400,:)=0;
% FrameRef(500:580,950:1080,:)=0;
%redimensionnement de l'image référence -solution non retenue-
% FrameRefRedi = FrameRef(20:1015,380:1620,:);
%affichage de la frame
figure(1)
imshow(FrameRef)
title('image référente');

```

```

%initialisation du disque de convolution (forme de la bille)
D = double(strel('disk',9).Neighborhood);
%initialisation de vecteur sauvegardant les coordonnées
CoordMuseum = [];
for frame = 65: v.NumFrames
    thisFrame = read(v, frame); %lecture de la frame

    %redimensionnement de l'image -solution non retenue-
    % thisFrameRedi = thisFrame(20:1015,380:1620,:);

    %suppression des zones sensibles
    % thisFrame(400:480,430:490,:)=0;
    % thisFrame(920:1080,500:1500,:)=0;
    thisFrame(:,1:400,:)=0;
    % thisFrame(500:580,950:1080,:)=0;
    %affichage de la frame
    % figure(2)
    % imshow(thisFrame)
    % title('image initial');
    %soustraction avec l'image référente
    thisFrame = FrameRef - thisFrame;

    thisFrame = double(thisFrame(:,:,:>150)); %seuillage
    thisFrame = mean(thisFrame,3);          %moyennage

    %affichage de la frame
    % figure(3)
    % imshow(thisFrame)
    % title('image après seuillage, et moyennage');

    %Convolution de la frame avec la forme de la bille
    thisFrame = imfilter(thisFrame,D);
    thisFrame = thisFrame / max(thisFrame,[],'all'); %normalisation
    %affichage de la frame
    % figure(4)
    % imshow(thisFrame)
    % title('image après convolution');

    %détermination de la coordonnée de la bille
    coord = find(thisFrame==1);
    coord = [ mod(coord,size(thisFrame,1)) int32(coord/size(thisFrame,1))];
    coord = double((mean(coord,1)));

    %concaténation des coordonnées de la bille
    CoordMuseum= horzcat(CoordMuseum, coord');
end
%re dimensionnement des axes x et y
CoordMuseum(1,:) = (size(FrameRef,1) - CoordMuseum(1,:))*r_px_m; %y

```

```

CoordMuseum(2,:) = CoordMuseum(2,:)*r_px_m; %x

%élimination des valeurs abérentes
for i = 2:size(CoordMuseum,2)
    if abs(CoordMuseum(2,i) - CoordMuseum(2,i-1)) > 0.1
        CoordMuseum(:,i) = CoordMuseum(:,i+1) ;
    end
end

%calcul de la vitesse de la bille
Vy=imfilter(double(CoordMuseum(1,:)),[-1 1],'symmetric') /dt;
Vx=imfilter(double(CoordMuseum(2,:)),[-1 1],'symmetric') /dt;
%compensation sur les valeurs abérantes
indAber = find(Vx==0);
for i = 1:size(indAber,2)
    Vx(indAber(i)) = Vx(indAber(i)-1);
    Vy(indAber(i)) = Vy(indAber(i)-1);
    Vx(indAber(i)+1) = Vx(indAber(i));
    Vy(indAber(i)+1) = Vy(indAber(i));
end

V = sqrt( Vx.^2 + Vy.^2 );
%affichage de la trajectoire de la bille
figure(5)
plot(CoordMuseum(2,:),CoordMuseum(1,:));
axis('equal');
title('trajectoire de la bille');
%affichage des projections sur l'axe x et y
figure(6)
plot((1:length(CoordMuseum(1,:)))*dt,CoordMuseum(1,:));
hold on
plot((1:length(CoordMuseum(1,:)))*dt,CoordMuseum(2,:));
legend('position en y','position en x');
%affichage des vitesse de la bille
figure(7)
plot((1:length(Vy))*dt,Vy);
hold on
plot((1:length(Vy))*dt,Vx);
plot((1:length(Vy))*dt,V);
legend('Vitesse selon l axe y','Vitesse selon l axe x','Norme de la vitesse');

```