

Voici une représentation du ventricule droit, avec en noir, la courbe représentant l'hyperbole qui permet le déroulé du ventricule droit. Le but sera donc de déterminer les caractéristiques paramétriques de cette hyperbole, à partir des données expérimentales.

L'hyperbole passe par 2 points essentiels : la valve tricuspide, et la valve pulmonaire, représentés dans le schéma par "TV" et "PV"

Gestion du projet

1. Environnement de travail

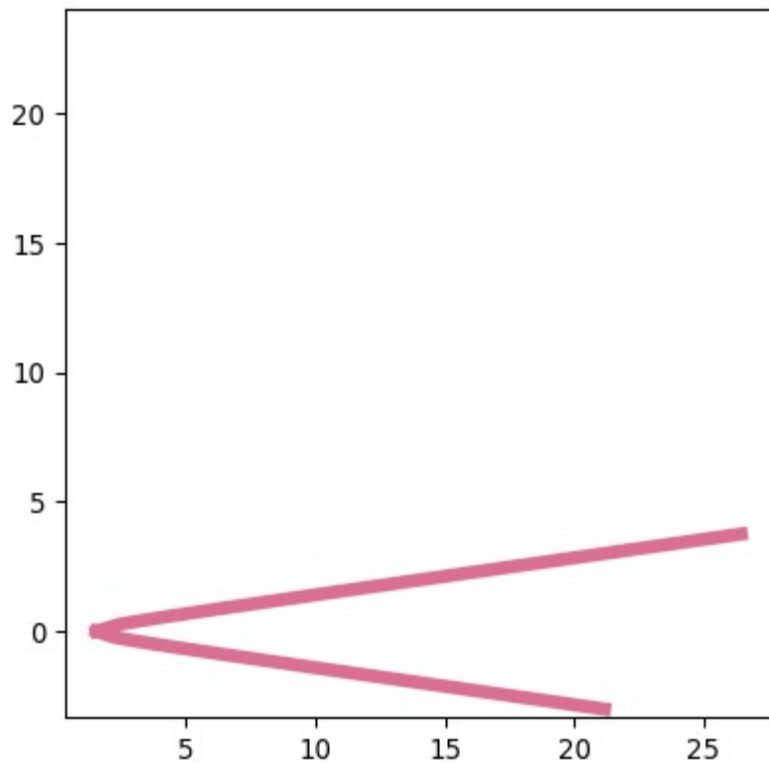
Durant cette première semaine, j'ai premièrement utilisé un outil de traitement et d'analyse d'images appelé Imagej, qui est répandu dans le milieu biomédical.

Cet outil m'a permis d'extraire des coordonnées en x,y,z d'une liste de points à partir de scanners de ventricule droit, que j'ai par la suite pu modéliser à l'aide de : Matplotlib et python. Matplotlib est une librairie python permettant de faire de l'analyse et visualisation de données.

Missions

Nous avons débuté le travail concernant l'hyperbole sur Matplotlib, en 2d

Une fois l'obtention de l'équation paramétrique de l'hyperbole grâce aux paramètres relevés sur Unity3d, voici l'hyperbole en 2d sur matplotlib :



```
a = 1.55675
b = m.radians(12.7)

x1 = np.linspace(1.55675, 26.44617, 30)
x2 = np.linspace(1.55675, 21.16413, 30)

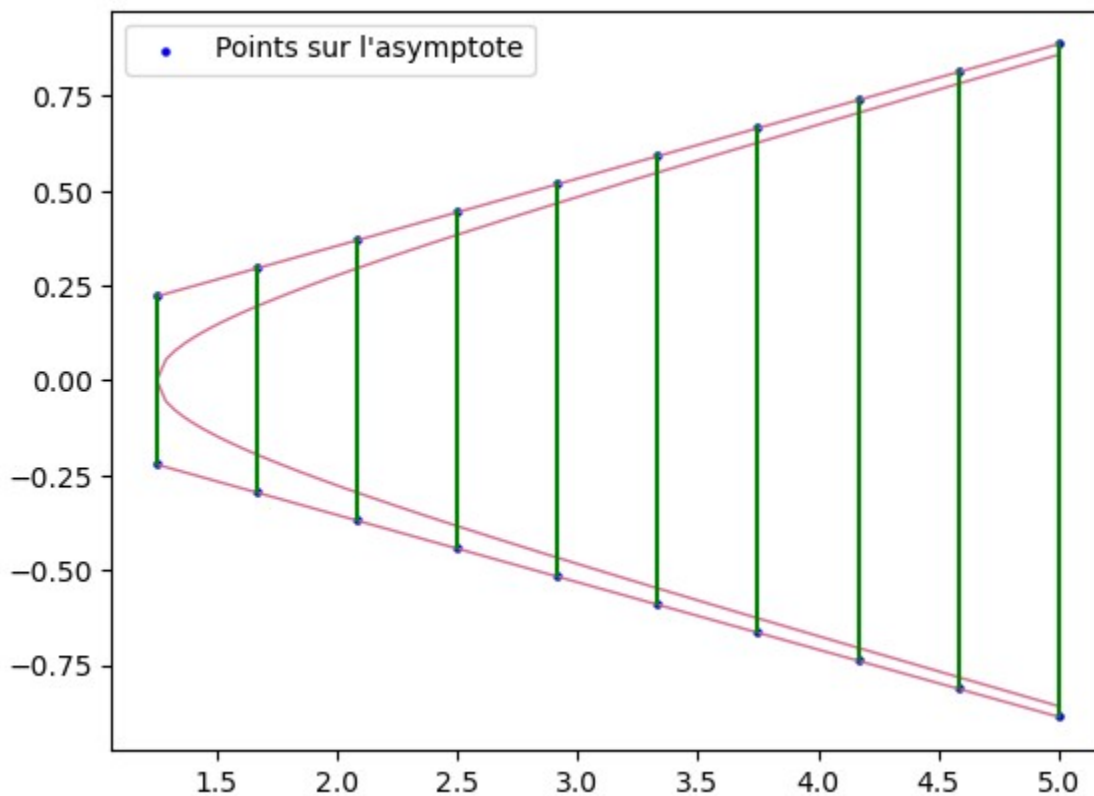
ypos = b * ((x1**2/a**2)-1)**0.5
yneg = - b * ((x2**2/a**2)-1)**0.5

# Create a 3D plot
```

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Plot the hyperbola in 3D
ax.plot(x1, ypos, zs=0, linewidth=5, color='palevioletred')
ax.plot(x2, yneg, zs=0, linewidth=5, color='palevioletred')
```

Pour obtenir cette hyperbole sur unity3d nous avons décidé de passer par une méthode un peu particulière, nous aurions pu essayer de la tracer directement sur Unity mais il est plus compliqué de paramétrer l'hyperbole dans l'espace 3D. Alors on l'a d'abord tracée en 2d, puis la méthode pour reporter ce tracé, en 3d sur unity est la suivante :



Tracer les asymptotes de l'hyperbole qui sont d'équation :

$$y = (b/a) * x \text{ (pour l'asymptote positive)}$$

$$y = -(b/a) * x \text{ (pour l'asymptote négative)}$$

Puis placer sur chacune des asymptotes, des points répartis de façon équidistante.

Relier chaque point d'une asymptote avec son homologue sur l'autre asymptote.

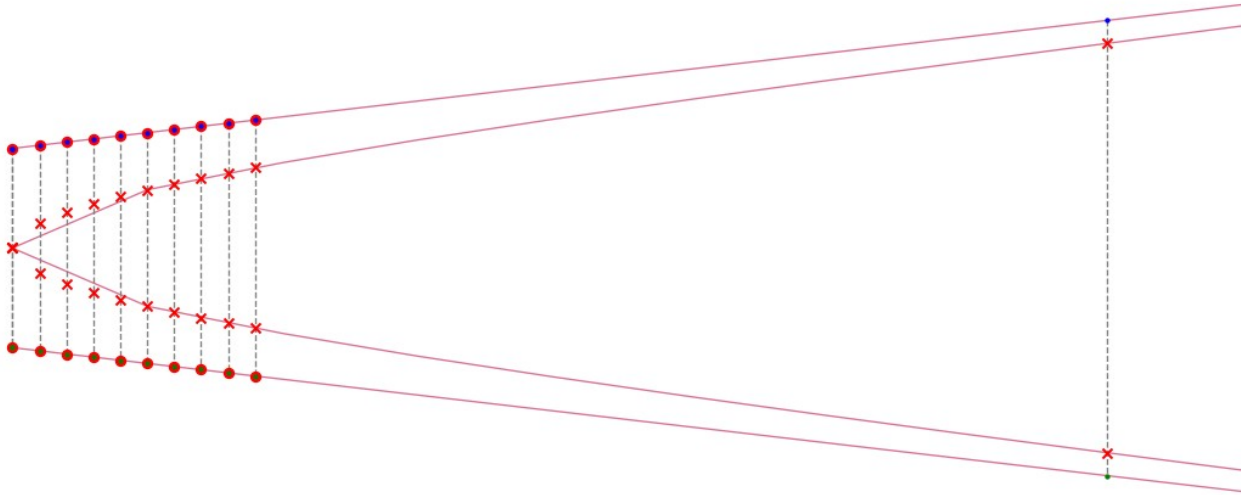
Le segment ainsi obtenu va avoir 2 points d'intersection avec l'hyperbole, tous ces points d'intersection, seront ensuite reportés sur unity3d afin de constituer les points de l'hyperbole, que l'on va ensuite relier, pour former la courbe de l'hyperbole.

Le problème étant le suivant : Il nous manque des points au niveau de l'apex de l'hyperbole :



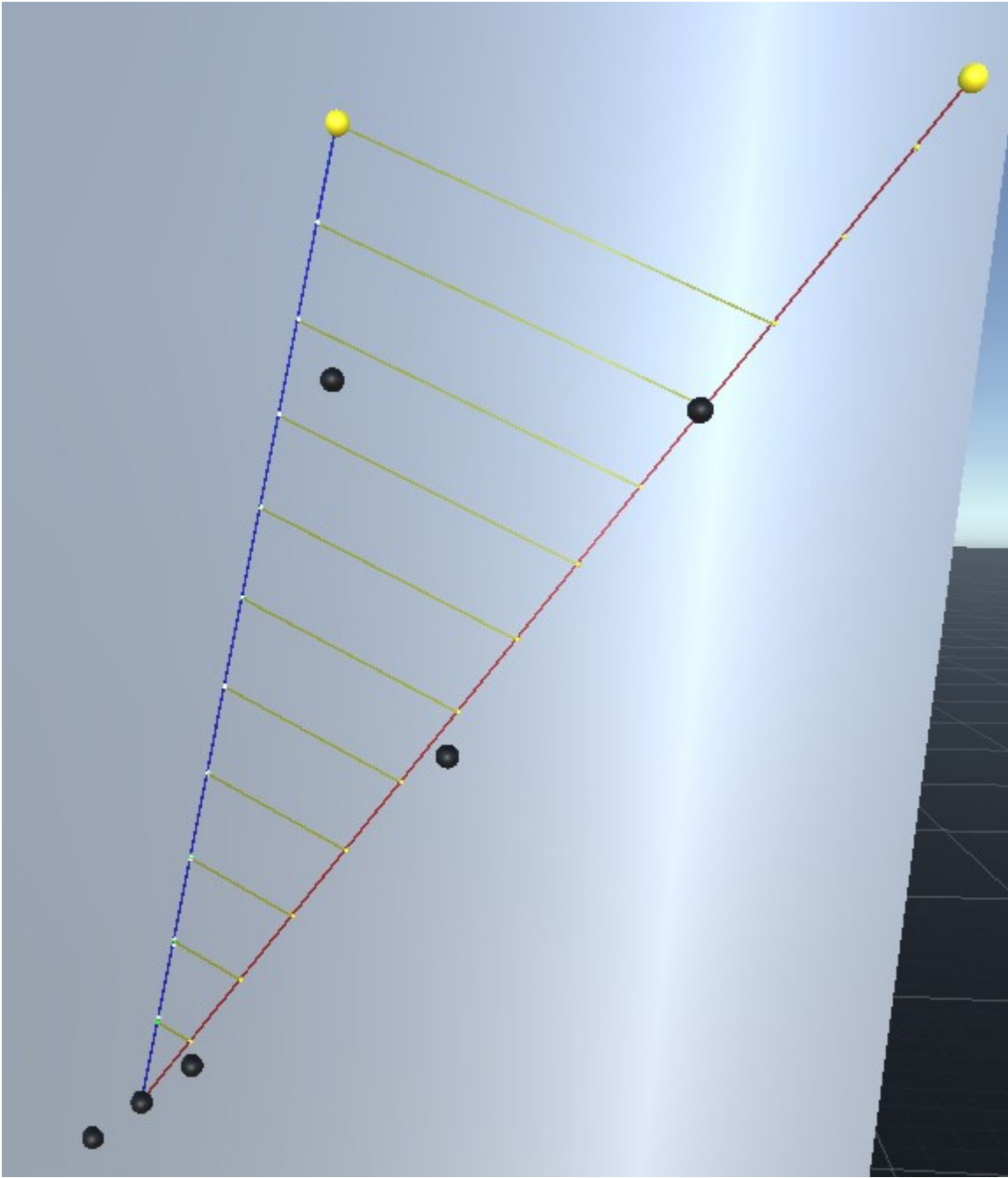
Cette zone manquera de détails lorsque l'on va reporter les points sur Unity3d.

Alors nous avons décidé d'appliquer un échantillonnage serré dans la zone d'intérêt, afin d'obtenir une meilleure résolution de la courbure de l'hyperbole :



Les points d'intersection sont représentés par des croix rouges, comme on peut le voir il y en a beaucoup plus, nous augmenterons peut-être encore plus l'échantillonnage si cela ne suffit pas.

Voici le tracé des asymptotes, et des points sur Unity3d :



Pour la suite, afin de tracer l'hyperbole, nous avons calculé les rapports suivants (sur Matplotlib) :

- Le rapport du nombre de points dans la zone échantillonnée sur le nombre de points total.
- Le rapport de la longueur de la zone échantillonnée sur la longueur totale de l'asymptote.
- Le rapport de la distance entre chaque point de l'asymptote et l'intersection qui lui correspond.

Ainsi, on va pouvoir reporter tout cela sur Unity3d, ce travail sera effectué la semaine prochaine.

Conclusion

Nous avons bien avancé concernant le travail sur l'hyperbole, la semaine prochaine nous commencerons à voir le résultat final. Il faudra ensuite améliorer l'ensemble et tenter d'optimiser le code, ajouter des fonctionnalités d'interactivité permettant aux chercheurs de pouvoir travailler sur plusieurs coeurs sans avoir à changer de programmes en permanence. (Idéalement)