

- ImageJ est un logiciel libre : le code source est en accès libre et peut être modifié. C'est un logiciel multiplateforme, en raison de son fonctionnement sur une machine virtuelle Java. ImageJ peut visualiser, éditer, traiter et analyser des images ou piles d'images en 8-bits, 16-bits, 32-bits et des images couleur. ImageJ est capable d'ouvrir et sauvegarder des images dans de nombreux formats, notamment en TIFF, PNG, JPEG, BMP, DICOM, FITS, mais aussi des images en format brut (raw).

- Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous forme de graphiques. Elle peut être combinée avec les bibliothèques Python de calcul scientifique NumPy et SciPy. Elle fournit également une API orientée objet, permettant d'intégrer des graphiques dans des applications, utilisant des outils d'interface graphiques polyvalent.

- Unity est un moteur de jeu multiplateforme développé par Unity Technologies. Il a la particularité de proposer une licence gratuite avec quelques limitations de technologie avancées au niveau de l'éditeur, mais sans limitation au niveau du moteur. Le logiciel est utilisé dans le cadre du stage pour de la visualisation en 3D.

- Paraview est un logiciel libre de visualisation de données. Il est fondé sur la bibliothèque VTK et publié sous licence BSD. Il est développé principalement par le Sandia National Laboratories, le Los Alamos National Laboratory et la société Kitware Inc.

3 Missions

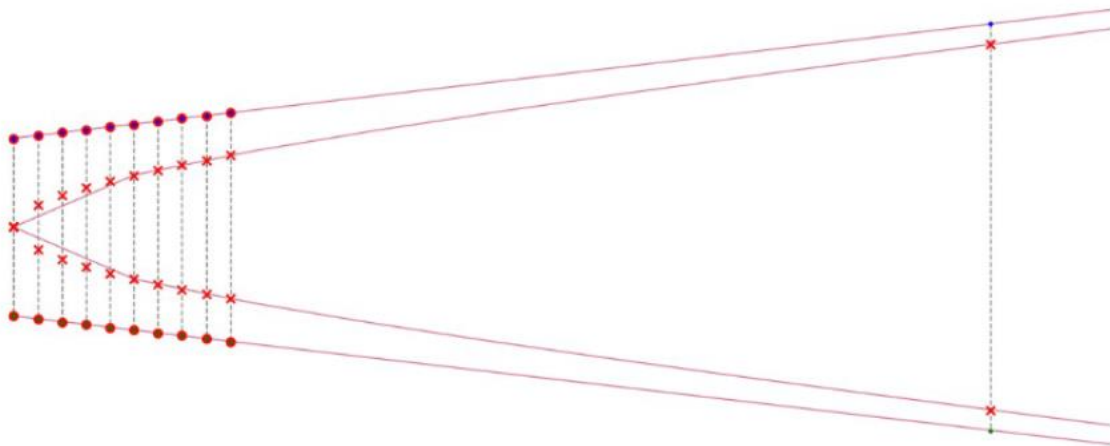
Au cours de cette cinquième semaine de stage, je vais améliorer la représentation l'hyperbole obtenue la semaine dernière sur Unity. En effet, je vais dans un premier temps effectuer un échantillonnage de points plus important vers la pointe de l'hyperbole sur matplotlib en utilisant les méthodes précédentes. Puis, je vais représenter l'hyperbole sur Unity avec le bon échantillonnage de points.

3.1 Amélioration de la représentation de l'hyperbole

3.1.1 Échantillonnage des points

Pour commencer, on a décidé de tracer 10 points sur chacune des asymptotes de l'hyperbole afin d'avoir une courbe plus intéressante et plus précise de celle-ci.

Voici la représentation de cet échantillonnage serré :



3.1.2 Relevé des distances

Ensuite, on a réappliqué les méthodes utilisées précédemment, c'est-à-dire le calcul de chaque distance des points sur l'hyperbole et les asymptotes.

On a d'abord calculé la distance qui sépare chacun des points sur les asymptotes et le centre de l'hyperbole. Puis, on a aussi calculé la distance qui sépare chacun des points sur les asymptotes et l'intersection des droites avec l'hyperbole.

3.1.3 Calculs des rapports de distances

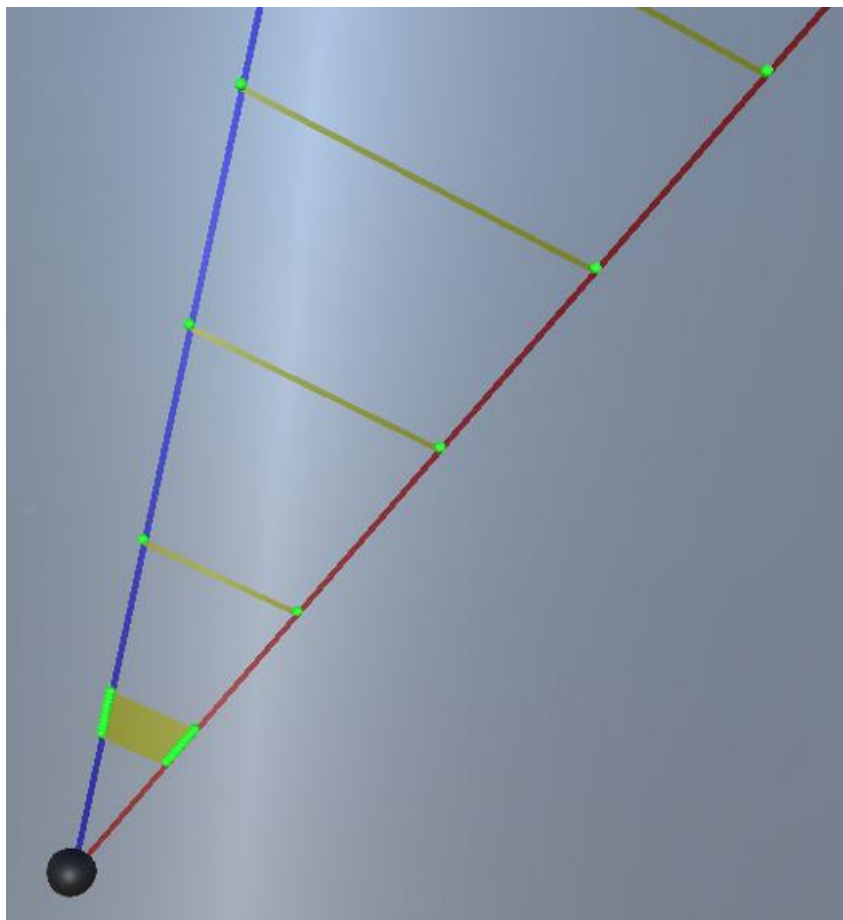
Afin d'obtenir des valeurs de distances correctes, on a décidé de calculer des rapports de distance. En prenant les distances brutes, il aurait été possible que les valeurs ne correspondent peut-être pas entre matplotlib et Unity.

Nous avons alors calculé 2 rapports de distances : un premier rapport entre la distance séparant chaque point des asymptotes avec le centre et la longueur totale de l'asymptote. Puis, un deuxième rapport entre la distance séparant chaque point des asymptotes avec l'intersection de la droite et l'hyperbole, et la longueur totale de cette droite.

3.2 Représentation finale de l'hyperbole sur Unity

3.2.1 Utiliser les rapports pour représenter les points

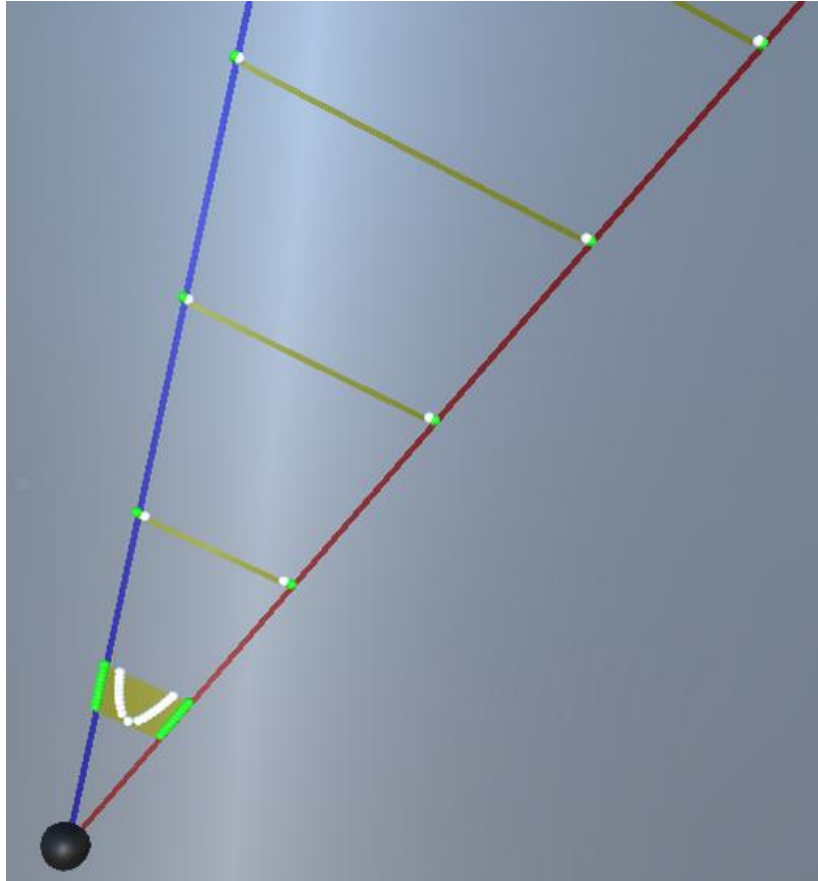
Une fois tout réalisé sur matplotlib, j'ai effectué les mêmes démarches afin de représenter tous les points ainsi que toutes les distances sur Unity. J'ai commencé par représenter les points sur les asymptotes en utilisant les rapports obtenus sur matplotlib et j'ai également tracé les droites qui relient les paires de points entre les deux asymptotes.



Le résultat obtenu sur Unity est bien ressemblant à ce que l'on a représenté sur matplotlib.

3.2.2 Représenter les points de l'hyperbole

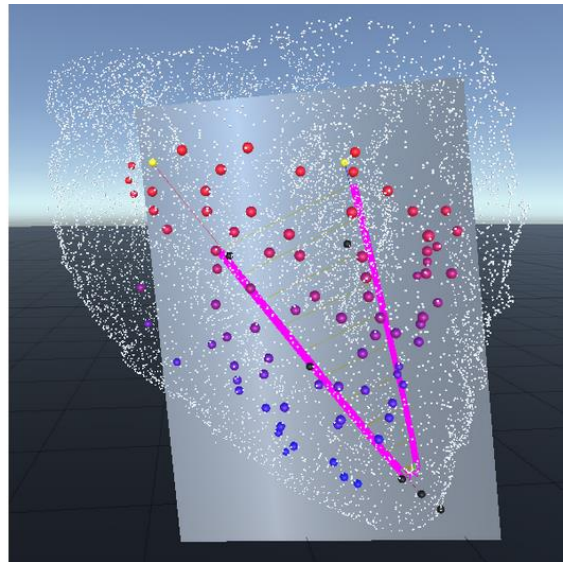
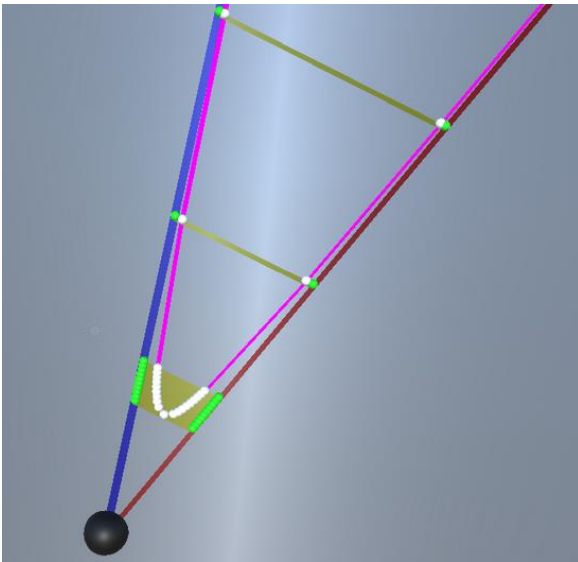
J'ai par la suite affiché les points sur l'hyperbole dans les deux parties, la partie avec l'échantillonnage serré et le reste des points le long de l'hyperbole.



Le résultat obtenu nous permet bien de voir la courbure de l'hyperbole à son centre. Il ne reste plus qu'à représenter l'hyperbole le long des points blancs sur l'image.

3.2.3 Représenter l'hyperbole

Je vais maintenant réaliser la représentation de l'hyperbole finale en remplissant bien tous les prérequis en suivant bien tous les points blancs récupérés sur matplotlib et également passant par les points tricuspide et pulmonaire.



On peut voir sur l'image de gauche, l'hyperbole passant par tous les points. Puis, sur la droite, on voit l'hyperbole tracée dans le plan avec une vision de l'hyperbole dans le ventricule droit du cœur. Il est également bon à rappeler que l'hyperbole doit bien passer par les points tricuspide et pulmonaire soit les points jaunes.