Rotation matricielle

Travail de maturité 2025

**Une image contenant avion, Transport aérien, transport, texte

Le contenu généré par l’IA peut être incorrect.**

Gymnase de bienne et du jura bernois

Eric friederich

Mentor : M. Hochuli

Table des matières

[Remerciements 2](#_Toc211592073)

[Contexte et Motivation 2](#_Toc211592074)

[Objectifs du Travail (introduction) 2](#_Toc211592075)

[Rotation en 2 dimensions 2](#_Toc211592076)

[Fondements mathématiques de la Rotation en 2D 2](#_Toc211592077)

[Transformation linéaire 4](#_Toc211592078)

[Homothétie 4](#_Toc211592079)

[Rotation 5](#_Toc211592080)

[Rotation autour d’une autre référence 7](#_Toc211592081)

[Rotation en 3 dimensions 8](#_Toc211592082)

[Fondements Mathématiques de la Rotation en 3D 9](#_Toc211592083)

[Trouver le vecteur autour duquel on tourne 9](#_Toc211592084)

[Quel est l’angle avec lequel on tourne autour d’un axe 10](#_Toc211592085)

[Faire tourner un vecteur autour d’un vecteur unitaire quelconque 10](#_Toc211592086)

[Angles d’Euler 13](#_Toc211592087)

[Trouver Angles d’Euler 14](#_Toc211592088)

[Conclusion 17](#_Toc211592089)

[Sources 18](#_Toc211592090)

[Annexe 19](#_Toc211592091)

# Remerciements

Avant tout, je souhaite exprimer ma gratitude à mon mentor, qui a su me guider avec bienveillance et efficacité tout au long de ce travail. J’ai également l’honneur d’être le dernier élève qu’il aura accompagné dans un travail de maturité.

# Contexte et Motivation

Ce qui m’a motivé à choisir ce thème était mon affinité avec les mathématiques. Mais je ne voulais pas effectuer un travail de maturité à 100 % en mathématiques, je cherchais une partie pratique dans laquelle je pourrais appliquer des maths. Durant la période à laquelle je cherchais un TM, j’ai reçu une mauvaise note en math appliquées sur des matrices. Par curiosité et désir de vengeance, j’ai choisi de m’orienter de ce côté-là. Par chance, les matrices cochaient parfaitement mes désirs de mixité et il semblait très intéressant à étudier les applications de celle-ci dans la rotation d’objet. De plus, ce sujet m’offrait une partie de codage challengeant pour moi qui ne suis pas très doué en informatique.

# Objectifs du Travail (introduction)

L’objectif de mon travail est de comprendre et d’expliquer les rotations de forme géométrique, que ce soit dans l’espace ou sur le plan. Ma démarche sera de guider le lecteur à travers les explications à ce sujet. De lui démontrer par voie calculatoire les sujets qui lui seront expliqués et pour finir, le convaincre par des simulations informatiques qui seront bien sûr aussi expliquées.

Durant ce travail, la première partie aura comme rôle d’introduire le lecteur à la rotation. Cette introduction sera faite par la vue de concepts basiques, puis de la rotation en 2 dimensions. Une fois les concepts bien en tête, nous pourrons attaquer la pierre angulaire de ce thème, la rotation en 3 dimensions. Diverses choses seront vues à ce sujet.

Durant la lecture de cette feuille, il est conseillé que le concept de matrice vous soit familier

# Rotation en 2 dimensions

## Fondements mathématiques de la Rotation en 2D

**Notions de Base**

Avant de commencer dans le sujet des rotations il est important de savoir ce qu’est un espace vectoriel et un vecteur.

**Définition d’un espace vectoriel et de vecteur en 2D et 3D:**

En algèbre linéaire un espace vectoriel est un ensemble d’objets appelés des vecteurs qui obéissent à des règles (des axiomes). Un vecteur peut être représenté géométriquement par une flèche (dans ou ), mais plus généralement, c’est un objet abstrait appartenant à l’espace vectoriel. Dans ou sa direction et sa taille sont définies à l’aide d’une matrice de dimension 2×1 () ou 3×1 (. Les composantes a,b et c nous indiquent de combien d’unités il faut se déplacer parallèlement à l’axe ox, oy et oz pour aller de l’origine du vecteur à son point terminal. Sur le plan il y a deux vecteurs unitaires de référence, .

Les vecteurs peuvent être additionnés entre eux pour créer de nouveaux vecteurs et peuvent être multipliés par un scalaire (généralement un nombre réel) pour s’allonger, se rétrécir ou changer de direction. Ces deux propriétés nous permettent de faire des combinaisons linéaires. Une combinaison linéaire est une addition de vecteurs multipliés par des scalaires.

Exemple :

Ici nous avons fait deux combinaisons linéaires pour représenter les vecteurs à l’aide des deux vecteurs unitaires de référence et le scalaire 1 et -2.

**Définition du système de coordonnées cartésienne et polaire :**

Chaque point du plan peut être repéré par ses coordonnées cartésiennes (x ;y), qui indiquent sa position par rapport aux axes ox et oy. On peut aussi repérer ces points par leurs coordonnées polaires (r,θ). On aura r comme la distance entre l’origine et le point et θ l’angle que fait le segment reliant l’origine au point avec l’axe ox.

On le représente :

Une image contenant diagramme, ligne, cercle, Tracé

Le contenu généré par l’IA peut être incorrect.

Figure

Le point B est donné comme avec des coordonnées cartésiennes.

Le point B est donné comme (distance ; angle en degré ou radiant)

Voici la conversion de polaire à cartésien

## Transformation linéaire

Une transformation linéaire dans ou ou associée à une matrice est une application qui, à un vecteur , fait correspondre le vecteur avec les propriétés suivantes :

2. *représente la première colonne de la matrice* ,  *la deuxième colonne et la troisième colonne*

La transformation linaire nous donne plusieurs possibilités, comme celle de faire des homothéties ou encore des rotations. Nous allons brièvement explorer les homothéties comme exemple à la transformation linéaire, puis plus précisément les rotations.

On peut voir la transformation linéaire comme un outil qui transforme la position des points du plan.

Une image contenant ligne, diagramme, Tracé, Parallèle

Le contenu généré par l’IA peut être incorrect.

Figure

Les colonnes de la matrice correspondent aux images des vecteurs de base et après la transformation linéaire. (4ème propriété)

Sur l’image nous voyons que et on construit donc grâce à la 4ème propriété la matrice de la transformation présente sur l’image.

### Homothétie

Le principe de l’homothétie consiste à agrandir ou rétrécir. Nous y arrivons généralement à l’aide d’un scalaire k ce qui nous donne . Il nous faut donc une fonction qui joue le même rôle :

On multiplie la matrice identité par le scalaire ce qui donne la matrice , qui réalise une homothétie de rapport. On multiplie par si on se trouve dans et si on est dans .

Exemple :

Je veux agrandir un vecteur  d’un facteur 2.

est donc égal à 2. On multiplie à car dans l’exemple nous nous trouvons dans l’espace vectoriel .

Calculons maintenant :

a effectivement été allongé dans la même direction d’un facteur 2.

## Rotation

Passons maintenant à la rotation. Nous souhaitons tourner un vecteur autour de l’origine.

En premier cas : Imaginons le vecteur que nous allons faire passer au vecteur , cela représentera une rotation de 90 degrés pour le vecteur par rapport à l’origine.

En deuxième cas : le vecteur deviendra s’il tourne de 90 degrés par rapport à l’origine.

Rappelons-nous maintenant de la 4ème propriété de la transformation linéaire qui dit :

*représente la première colonne de la matrice et représente sa deuxième colonne.*

Cela nous permet de construire notre première matrice de rotation. Cette matrice de rotation nous permet de faire tourner un vecteur de 90 degrés par rapport à l’origine.

Cette méthode nous permet de faire une rotation de 180 degrés en prenant et . Mais elle se limite à des angles multiples de 90, comme 270 degrés.

Cette réflexion nous emmène au fait qu’il est possible d’élaborer une matrice nous permettant de faire tourner un vecteur d’un angle , .

Une image contenant diagramme, ligne, cercle, Tracé

Le contenu généré par l’IA peut être incorrect.Avant de commencer je tiens à rappeler que la norme des deux vecteurs est égale à 1. Deuxièmement, ont pour origine O ce qui veut dire que son point terminal coïncide avec ses composantes.

Figure

Ici nous voyons tourner d’un angle autour de O. Ecrivons les coordonnées polaires des points terminaux des vecteurs puis, transformons-les en coordonnées cartésiennes. Et n’oublions pas qu’un angle de 90 degrés sépare le vecteur par rapport à l’origine.

)

En trouvant les coordonnées des points terminaux, on obtient aussi les composantes de .

et

Une fois l’image de trouvées, plus qu’à écrire la matrice grâce à la 4ème propriété de la combinaison linéaire. Elle va être égal à :

Voilà notre matrice nous permettant de faire tourner un vecteur d’un angle autour de l’origine.

Un exemple s’impose :

Imaginons auquel nous allons appliquer une rotation de 90 degrés.Logiquement, nous allons tomber sur le vecteur . Commençons par définir la bonne matrice.

=

La matrice obtenue ici est cohérente avec celle déterminée plus tôt par expérimentation. Et

Essayons maintenant une rotation de 45 degrés.

Une image contenant ligne, texte, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.On aura donc :

Figure

### Rotation autour d’une autre référence

Pour clôturer le thème de la rotation, parlons de rotation autour d’un autre point que l’origine.

Parlons tout d’abord de translation. , cela nous informe sur la translation qui sera appliquée à un point. Une translation est une application affine et non linéaire, car elle ne conserve pas l’origine (3ème propriété). Elle est telle que

Ici on parlera de vecteur comme des points avec leurs coordonnées à la place de leur composante.

Si on souhaite faire tourner un point autour d’un point qui n’est pas l’origine, il faudra suivre une démarche particulière.

On va tout d’abord effectuer une translation pour ramener le centre de rotation à l’origine. Cette translation sera appliquée à le ou les points censés tourner.

On effectue la rotation normalement.

Enfin, on effectue la translation inverse à la première .

Cette démarche permet de simuler une rotation autour de l’origine en ramenant le repère créé par le point de rotation à l’origine.

Un exemple s’impose :

Imaginons un point de coordonnées que nous voulons faire tourner de 90 degrés autour du point et non de l’origine.

Pour ramener le point de centre de rotation à l’origine on va trouver une translation qui fera passer à l’origine donc . Ici on aura qu’on devrait appliquer à .

Maintenant que l’on sait quelle translation faire, on va l’appliquer au point que nous voulons faire tourner.

Faisons maintenant la rotation du nouveau point qui redeviendra un vecteur le temps d’une application linéaire.

On obtient donc le point de après rotation.

Plus qu’à faire la translation inverse. Cela va replacer le centre de rotation à son emplacement antérieur.

Une image contenant ligne, diagramme, Parallèle, texte

Le contenu généré par l’IA peut être incorrect.

Figure

# Rotation en 3 dimensions

## Fondements Mathématiques de la Rotation en 3D

**Introduction**

Ici, comme en 2D, on va faire des rotations à l’aide de transformations linéaires. Une matrice 3×3 sera appliquée à un vecteur 3×1. Le vecteur subira une rotation antihoraire autour d’un un axe selon le principe de la main droite.  
Cet axe pourra être ox, oy ou oz, ou encore un axe induit par un vecteur quelconque, comme nous le verrons plus tard.

En tournant autour de l’axe ox, on fera une rotation sur le plan oyz, cette logique se reporte aux autres axes : Avec oy, nous nous trouvons sur le plan oxz et avec oz, c’est sur le plan oxy. Nous reconnaissons dans le dernier cas notre abscisse et ordonnée et donc le plan sur lequel nous avons travaillé dans l’espace . Bien sûr, si nous tournons par exemple autour de ox, tous vecteurs parallèles à cet axe seront invariants à la rotation, ce qui représente ().

Avec ça en tête nous allons pouvoir construire les matrices de rotation qui feront tourner un vecteur autour des 3 axes. Nous allons utiliser nos 3 vecteurs unitaires pour construire nos matrices.

Commençons par une rotation autour de donc l’axe Z :

Pour commencer, nous voyons que la troisième colonne représente f(e3) d’après la 4ème règle des transformations linéaires. ce qui veut dire que est un invariant à la rotation. Ça concorde avec notre but, faire tourner les vecteurs autour de .

Deuxièmement, la dernière ligne de la colonne 1 et 2 sont nulles car le vecteur tourne sur le plan oxy et ne varie donc pas sur l’axe oz.

Enfin, nous pouvons placer la matrice de rotation faisant tourner le vecteur sur le plan oxy par sa position.

En prenant la même logique nous retrouvons aisément les matrices permettant la rotation autour de et qui sont :

et

Précision : Dans et quand on fait une rotation, la matrice rotation aura un déterminant égal à 1 et elle sera orthogonale.

### Trouver le vecteur autour duquel on tourne

Imaginons maintenant que nous nous trouvons devant une matrice de rotation et qu’il nous est demandé de trouver autour de quel vecteur cette matrice fait tourner, dans ce cas il existe une méthode pour retrouver ce vecteur. On suppose que ça tourne autour d’un vecteur propre de valeur propre 1. Donc est un vecteur non nul qui satisfait cette équation (. Plus qu’à résoudre.

### Quel est l’angle avec lequel on tourne autour d’un axe

Maintenant que nous savons trouver le vecteur autour duquel tourne un vecteur, nous voulons savoir l’angle de rotation θ associé à la matrice R. Pour ceci, nous allons calculer la trace de cette matrice de rotation.

La somme des valeurs présentes sur la diagonale allant de est égale à sur les trois matrices de rotation. C’est la trace. La méthode consiste à additionner les valeurs sur la diagonal. Faire une égalité entre cette valeur et puis de résoudre l’équation en isolant l’angle .

Un exemple s’impose :

Voici une matrice. Nous savons que c’est une rotation.

Sur quel axe tournons-nous ?

Nous allons poser :

On résout et on trouve :

de façon plus claire, on va donc tourner autour de

Maintenant que nous savons autour de quel axe nous tournons, trouvons l’angle auquel on tourne.

La somme des valeurs sur la diagonal est égale à et qui est aussi égale à on obtient donc :

Passons maintenant à un sujet qui a déjà été évoqué.

Comment faire tourner un vecteur autour d’un axe autre que ox,oy et oz

## 

## Faire tourner un vecteur autour d’un vecteur unitaire quelconque

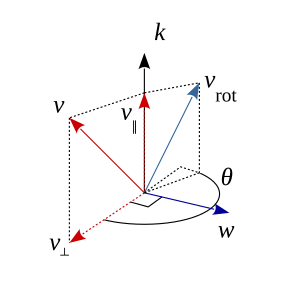
Dans cette partie nous souhaitons faire tourner un vecteur autour d’un autre vecteur unitaire quelconque. Pour y arriver nous aurons besoin de « la formule de rotation de Rodrigues » étant attribuée à Léonard Euler et Olinde Rodrigues. Elle fait tourner un vecteur autour d’un vecteur en décomposant les composantes parallèles et perpendiculaires à et en faisant tourner uniquement la composante perpendiculaire. Je vais la démontrer et la mettre en application.

Soit un vecteur quelconque tournant autour de un vecteur unitaire définissant l’axe de rotation. En utilisant les produits scalaire et vectoriel, le vecteur peut être décomposé en une composante parallèle et perpendiculaire à .

La composante parallèle à est la projection orthogonale de celui-ci sur . La démonstration est très largement inspirée de celle sur Wikipédia.

Pour commencer la formule de Rodrigues affirme que :

Avec comme vecteur après rotation, comme vecteur avant rotation, comme vecteur unitaire autour duquel va tourner et pour finir comme angle de rotation antihoraire.



Figure

En connaissant la composante parallèle et v on trouve la composante perpendiculaire

Par la formule du produit triple vectoriel ().

On va poser :

Car est unitaire.

On rend négatif car est à l’opposé de

Une image contenant transport, avion, conception

Le contenu généré par l’IA peut être incorrect.

Figure

Durant une rotation, la composante parallèle sera invariante à la rotation contrairement à la composante perpendiculaire. La composante perpendiculaire va tourner sur un plan perpendiculaire engendré par . Après rotation sur ce nouveau plan la composante perpendiculaire sera égale à :

Avec comme la composante parallèle à , on la multiplie à pour ajuster ça norme. On fait de même pour qui lui est la composante parallèle à .

En additionnant la composante perpendiculaire après rotation et la composante parallèle qui n’a pas changé après la rotation on retombe sur le vecteur que nous cherchions à l’origine.

Remplaçons par

Développons

Factorisons

On se souvient que

Maintenant que la formule de Rodrigue a été démontré, utilisons-là pour construire une matrice de rotation :

Nous voulons faire une matrice de produit vectorielle avec k afin de pouvoir transformer en une application linéaire ou on multiplierait a une matrice, ça nous sera utile pour plus tard :

= =

Donc :

Dorénavant, sera exprimé .

Reprenons la formule de Rodrigues :

En utilisant l’identité vectorielle et en factorisant on retombe sur l’expression :

Par l’application linéaire construite précédemment l’expression devient

=

Donc pour finir la matrice de rotation pour ce type de rotation est :

R =

## Angles d’Euler

Pour finir avec le thème des rotations, je vais introduire la rotation dans un format plus général. La première méthode de rotation en 3D nous permettait de faire tourner un vecteur autour des ox,oy et oz seulement. La deuxième méthode était meilleure car on pouvait choisir autour de quoi on allait tourner, cette méthode nous permettait théoriquement d’atteindre toutes les positions autour de l’origine, donc de faire tourner un cube par exemple dans le sens que l’on veut. Mais malheureusement elle n’est pas très pratique.

La solution est donc les rotations d’Euler bien qu’elle ait aussi ses défauts comme on le verra plus tard.

Cette partie est très largement inspirée de cette vidéo YouTube :

[(1588) 2.3 Rotations in 3D - YouTube](https://www.youtube.com/watch?v=wg9bI8-Qx2Q&list=PLYnuGw4Z3fygLBz0GsMArQ27z-NH1PfFG&index=6)

Pour commencer, une matrice de rotation dans l’espace peut être formée par des séquences de 3 rotations consécutives autour des axes primaires et d’angles , et . Ces rotations peuvent être dans n’importe quel ordre, mais pas deux fois sur le même axe à la suite. La rotation à la suite sur le même axe (le même angle) peut être défini simplement par une seule rotation.

Ça fait au total 12 (3\*2\*2) combinaisons de rotation possibles en excluant les rotations successives sur le même axe.

Le premier groupe est « Les angles d’Euler propres ». C’est quand dans la séquence il y a deux fois une rotation sur le même axe (pas à la suite). Il existe 6(3\*2\*1) combinaisons possibles.

Les voici : *z*-*x*-*z*, *x*-*y*-*x*, *y*-*z*-*y*, *z*-*y*-*z*, *x*-*z*-*x*, *y*-*x*-*y*

Le deuxième groupe est appelé « Les angles de cardan » ou encore « Les angles de Tait-Bryan ». Eux sont caractérisés par le fait qu’on va tourner sur tous les axes. Ça fait aussi 6 combinaisons possibles.

Les voici : *x*-*y*-*z*, *y*-*z*-*x*, *z*-*x*-*y*, *x*-*z*-*y*, *z*-*y*-*x*, *y*-*x*-*z*

Dans mon exemple, je vais choisir arbitrairement une rotation de séquence ZYX. Mais j’aurais très bien pu prendre d’autres angles de Cardan ou même des angles de Euler propres. La méthode reste la même. La seule différence est l’angle qui causera le blocage de cardan.

En quoi consiste cette rotation :

On va faire une rotation intrinsèque, autrement dit, les axes tournent avec l’objet, donc il faut multiplier les matrices de rotation de droite à gauche.

Il nous faudra un repaire orthonormé fixe (Re1) au début, puis nous allons faire 3 rotations consécutives grâce aux angles de Cardan qui vont nous amener à un repaire (Re3). La première rotation sera d’angle autour de l’axe ox, la deuxième sera d’angle autour de l’axe oy’ qui est l’axe oy dans le Re2 défini par la rotation précédente. Et pour finir, une rotation d’angle autour de l’axe oz’’ situé dans Re3 qui lui est défini par les deux dernières rotations.

La matrice de rotation sera donc égale à .

## Trouver Angles d’Euler

Dans cette partie notre objectif est :

Trouver l’angle auquel on va faire tourner un vecteur sur chaque axe de rotation.

Je vais prendre la suite de rotation Z’’Y’X. La matrice de rotation sera donc égale à .

(La valeur développée de est trouvable sur le tableau à la ligne 5 colonne 2)

Et ici « c » veut dire cosinus et « s » sinus.

Une image contenant texte, nombre, capture d’écran, écriture manuscrite

Le contenu généré par l’IA peut être incorrect.

Figure

On va s’aider de cette généralisation d’une matrice de rotation pour la démonstration.

Cette démonstration est largement inspirée de « Computing Euler angles from a rotation matrix » de Gregory G. Slabaugh

On trouve :

donc

Comme il y aura deux solutions possibles.

Le cas de sera traité plus tard.

Pour suivre on voit que :

donc

atan2 est une variante de la fonction arctan. Là où arctan ne fait pas la différence entre deux angles diamétralement opposés (ex : arctan(1/1)=arctan(-1/-1)) atan2 le fera. Ça spécifie sur quel quadrant du cercle trigonométrique se trouve la solution.

n’aura pas la même réponse en fonction du signe de à cause de la formule «atan2».

Si alors et alors

Une manière d’arranger sera de noter . Si est négatif le signe du cos au numérateur va rendre tout positif (ajouter pi) et placer la solution dans le bon quadrant.

Le cas de sera traité plus tard.

Il y a donc deux solutions :

En utilisant le même raisonnement on trouve :

En conclusion, nous auront deux solutions.

Et quoi si :

Dans ce cas on parlera de « gimba-lock »

Si alors ce qui va correspondre à

Si on essaie de résoudre comme avant on obtiendra

Ce qui ne marche pas. Nous allons utiliser d’autres élément de la matrice pour résoudre.

Avec :

Cela nous permet d’écrire cette équation.

donc et

Avec on tomberas sur un cas similaire

Et finalement

En conclusion quand il y aura une infinité d’angle possible pour alpha et gamma. En pratique on dira que alpha ou gamma est égal à zéro pour trouver la valeur de l’autre angle en fonction du premier.

**Trouver les angles avec une matrice et aucune information**

Dans cette dernière partie, on nous donne une matrice avec comme seule information que ça en est une de rotation. La question sera, comment trouver l’ordre des axes sur lesquels il va faire tourner et à quel angle. Une idée informatique de ma part serait de faire un programme. Ce programme testera sur cette matrice toutes les procédures pour trouver les angles. Ça veut dire, la procédure pour une rotation XYZ, XYX etc… S’il y a un problème durant la procédure, ça veut dire que la suite est erronée donc on passe à la prochaine. Si nous trouvons effectivement 3 angles, plus qu’à faire une matrice reprenant ces angles et la suite d’axes et à la comparer à la matrice initiale. Si elles sont semblables ça voudra dire que l’on a bien trouvé la suite d’axes et les bons angles ; en cas contraire, la solution est erronée et on passe à la suite prochaine.

# Conclusion

Tout au long de ce travail, on a pu survoler le thème de la rotation. On a vu des concepts et diverses théories associées à ce sujet. Mais bien sûr, le thème est bien plus vaste que ça.

La rotation est un thème très important dans les mathématiques et surtout en ingénierie. On y voit des applications en aviation, en robotique et en informatique pour ne citer que quelques-uns.

De manière plus personnelle, j'ai vraiment aimé effectuer ce travail. Étudier les divers sujets a vraiment été une source de satisfaction, que ce soit la recherche d'informations, la compréhension de la matière ou encore la structuration du travail. Entre le début et la fin de mon projet, j'ai vraiment ressenti une prise de maturité de ma part. Elle était caractérisée par la façon avec laquelle j’aborde les documents et aussi mon efficacité au travail.

Je me souviens encore du jour auquel j’ai commencé mon TM. J’avais des connaissances assez limitées sur le sujet qui était à ce moment-là, les rotations en 2 dimensions et les transformations linéaires. Je venais à peine de comprendre ce concept mathématique que je me suis aperçu que le nouveau thème en maths en traitait une partie. Je suis forcé d’avouer que sur le moment ça m’avait beaucoup aidé. Même si au début, j’étais assez embarrassé, car je craignais de passer pour quelqu’un qui ne travaille pas et qui se contentait de recopier le cours. Cette peur s’est rapidement dissipée quand j’ai commencé à traiter les rotations en 3D.

Ma démarche pour ce travail était simple, j'allais sur internet chercher des sources, si elles me semblaient pertinentes, je les analysais et en tirais le contenu intéressant que je retranscrivais sur ma feuille tout en l’ayant compris.

Les moments forts étaient la découverte de la matière. J'aimais beaucoup le moment où je découvrais le thème, j'avais une sensation de vertige par rapport à l'information qui était à traiter, et en même temps une envie de m'en imprégner. La partie que j'ai beaucoup moins aimée était l'écrit. Ce n'était vraiment pas évident de retranscrire mon savoir sur une feuille, dans ma tête tout paraissait limpide, mais face à une feuille blanche c'était plus compliqué. Les points où j’aurais pu me rater sont déjà pour commencer, l’écriture comme j’ai pu le dire avant. Mais aussi peut-être l’exactitude mathématique. J’ai dû me lancer dans l’écriture de ce document mathématique à l’aide de mes connaissances. S’aider d’un exemple aurait pu faire du bien à mon travail. Malgré tout ce que j’ai dit, j’estime qu’il y a aussi eu des choses bien dans mon travail. Je trouve par exemple mon travail assez facile à comprendre. Cette facilité de compréhension peut être due à sa structuration. Je suis aussi assez fier de moi pour le programme informatique que j’ai fait.

Tout au long de ce TM, j’ai eu la chance de travailler dans un bon environnement et j’ai aussi été très bien entouré. Je n’ai honnêtement aucun regret. Même si, si c’était à refaire, je changerais quelques choses. Comme ma régularité au travail, j’ai surtout eu des périodes où je travaillais beaucoup et d’autres où je ne travaillais pas. J’aurais aussi intégré une partie sur les quaternions dans mon travail. J’avais vraiment envie d’étudier ce thème, mais par faute de temps ça n’a pas été possible. Finalement, j’aurais aussi fait un réel travail sur mon écriture.

Pour finir merci d’avoir lu mon premier document scientifique sérieux.

# Sources

Je me suis aidé de l’IA dans la formulation des phrases dans les remerciements.

**Transformation linéaire**

20-Les transformations linéaires de Daniel Audet :

<https://www.youtube.com/watch?v=cAV0AVEt4ec>

Le parti sur les transformations linéaire et la rotation du cours de Patrick Hochuli

**Rotation en 2 dimensions :**

Rotation autour d’autres point que l’origine et rotation en général :

<https://www.alloprof.qc.ca/fr/eleves/bv/mathematiques/les-rotations-dans-un-plan-cartesien-m1322>

<https://www.alloprof.qc.ca/fr/eleves/bv/mathematiques/les-matrices-de-transformation-m1432>

**Rotation en 3 dimensions :**

Rotation in the space de Yan-Bin Jia :

<https://faculty.sites.iastate.edu/jia/files/inline-files/rotation.pdf>

Rotation de l’espace auteur inconnu :

<https://exo7math.github.io/mathgame-exo7/rotation/rotation.pdf>

Angle d’Euler, Euler angles et Rodrigues' rotation formula de Wikipédia :

[Rodrigues' rotation formula - Wikipedia](https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula)

<https://fr.wikipedia.org/wiki/Angles_d%27Euler#Rotations_d'Euler>

<https://en.wikipedia.org/wiki/Euler_angles>

**Images :**

L’image de couverture :

<https://aviation.stackexchange.com/questions/16531/what-is-the-relation-between-roll-angle-and-pitch-angle>

Pour les figures 1 et 3, elles ont été faites par moi sur Geogebra.

Pour les figures 2,4 et 5, elles ont été faites par ChatGPT à la suite de mes instruction. Les images proviennent Python(matplotlib)

Pour les figures 6 et 7, elles ont été trouvées sur Wikipédia.

<https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula>

Pour la figure 8, elle a été trouvé sur Wikipédia.

<https://en.wikipedia.org/wiki/Euler_angles>

**Programme informatique :**

Pour m’aider sur la librairie PIL, j’ai eu un exemple de programme fait par lui et :

<https://www.geeksforgeeks.org/python/python-pillow-tutorial/>

<https://apprendrepython.com/comment-utiliser-pillow-pil-python-imaging-library/>

<https://cgouygou.github.io/TNSI/T09_Extras/Image/>

<https://www.codecademy.com/resources/docs/pillow/image/>

# Annexe

Comme allongement de ce projet, j’ai aussi codé un programme de rotation en 2d. Ce programme fait tourner une image à l’aide d’une matrice de rotation. Je l’ai codé grâce à mes connaissances acquises durant le projet.

J’ai aussi commenté un programme codé par chat GPT afin de démontrer mes connaissances sur des rotations en 3d.