**Sommaire**

1. Contexte et Motivation
   * Applications en informatique graphique, modélisation et robotique
2. Objectifs du Travail
3. Partie 1 : Rotation en 2D
   * Chapitre 1 : Fondements Mathématiques de la Rotation en 2D
     + Notions de Base
     + Matrice de Rotation en 2D
     + Coordonnées Homogènes en 2D
   * Chapitre 2 : Implémentation et Visualisation en 2D
     + Méthodes Numériques
     + Programmation et Simulation
     + Applications et Exemples
4. Partie 2 : Rotation en 3D
   * Chapitre 3 : Fondements Mathématiques de la Rotation en 3D
     + Matrice de Rotation en 3D
     + Quaternions et Rotation
     + Démonstration de toutes choses avancées
   * Chapitre 4 : Implémentation et Visualisation en 3D
     + Programmation de la Rotation 3D
5. Conclusion et Perspectives
   * Résumé des Résultats Obtenus
   * Perspectives et Travaux Futurs

**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 math. Durant la période à laquelle je cherchais un TM, j’ai reçu une mauvaise note en math appliqué sur des matrices, par curiosité et désire de vengeances, j’ai choisi de m’orienter de ce côté-là. Par chance, les matrices cochaient parfaitement mes désirs de mixité et il semblait très intéressent 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**

L’objectif de mon travail est de comprendre et d’expliquer les rotations de forme géométrique que ça soit dans l’espace ou sur le plan. Ma démarche sera de guider le lecteur à travers des explications des sujets, 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. Mon objectif ne sera pas de vulgariser ou d’écrire un texte trop léger, je souhaite être le plus rigoureux possible.

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

* Étudier et implémenter la rotation en 2D et en 3D
* Explorer les méthodes mathématiques et algorithmiques associées

**Partie 1 : Rotation en 2D**

**Chapitre 1 : Fondements Mathématiques de la Rotation en** **2D**

**Notions de Base**

Avant de commencer dans le développement le sujet de la rotation en 2 dimensions il est important de savoir ce qu’est un est un espace vectoriel et ce qu’est un vecteur.

**Définition d’un espace vectoriel :**

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

En algèbre linéaire un espace vectoriel est un ensemble d’objets, appelées des vecteurs. Un vecteur est un élément d’un espace vectoriel. Il est représenté comme un segment en forme de flèche ayant un point de départ et un point d’arrivé. Sa direction et sa taille sont définis à l’aide d’une matrice de dimension 2×1 () que l’on appelle ses composantes, a et b représente le nombre d’unités nécessaire pour aller d’un point de départ à un point d’arrivé en se déplacent de parallèlement à l’axe des abscisses puis celle des ordonnées. Sur le plan il y a deux vecteurs unitaires de référence, .

Les vecteurs peuvent être additionner entre eux pour créer de nouveaux vecteurs et peuvent être multiplier par un scalaire (généralement un nombre réel) pour s’allonger, se rétrécir ou changer de direction. L’addition des deux propriétés dites s’appelle une combinaison linéaire

On les représente comme ça dans un plan :

Une image contenant ligne, Parallèle, diagramme, Tracé

Le contenu généré par l’IA peut être incorrect.

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

Image à changer

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

Un plan est une surface plate illimité, munie de notion de d’alignement, d’angle et de distance. Il y a un point nommé origine au centre du plan et deux axes passant par ce point (axe : Ox et oY) parallèlement par rapport a eu. On peut y définir des points de façon cartésienne en notent ses coordonnées par rapport à l’axe oX et oY. Ou encore de façon polaire ou le point va être défini par la distance qui le sépare de l’origine et l’angle que forme le segment entre l’origine et le point par rapport à l’axe X.

On le représente :

Une image contenant diagramme, ligne, cercle, Tracé

Le contenu généré par l’IA peut être incorrect.

Le point B est donné comme avec des coordonnées cartésiennes.

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

La conversion de polaire a cartésien

* Espaces vectoriels et matrice de transformations
* Systèmes de coordonnées cartésien et polaire

**Transformation linéaire**

Une transformation linéaire de R2(le plan) T associée à une matrice Mf est une application qui, à un vecteur V, fait correspondre le vecteur f(v) = Mt\*v[[1]](#footnote-1)

2. *f(e1) représente la première colonne de la matrice f et f(e2) représente sa deuxième 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 et plus précisément les rotations. Nous pouvons imagines cette fonction comme un outil qui va modifier l’apparence du plan. Les deux colonnes de la matrice Mf représenterons les deux nouveau vecteurs e1 et e2 de la matrice modifiée. D’où la 4ème propriété.

Une image contenant texte, dessin, diagramme, conception

Le contenu généré par l’IA peut être incorrect.[[2]](#footnote-2)

**Homothétie**

Le principe de l’homothétie consiste en le fait d’agrandir ou de rétrécir. Nous y arrivons usuellement à l’aide d’une constant ce qui nous donne . Il nous faut donc une fonction f qui joue le même rôle donc :

Ce qui veut dire que la matrice de transformation avec car

Un exemple s’impose :

Je veux agrandir un vecteur  d’un facteur 2, donc k = 2 trouvons la matrice de transformation.

Calculons maintenant :

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

Vérifions les conditions ……………………….

Prenons maintenant

**Rotation**

Passons maintenant à la rotation. Nous souhaitons rotationnel un vecteur au tour de l’origine. En premier cas : Imaginons le vecteur e1 que nous allons faire passe au vecteur e2, cela représentera une rotation de 90 degrés pour le vecteur e1 par rapport à l’origine. En deuxième cas : le vecteur e2 deviendra -e1 s’il rotation de 90 degrés par rapport à l’origine. Mis en équation ça nous donne ça :

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

*F(e1) représente la première colonne de la matrice f et f(e2) représente sa deuxième colonne*

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

EXEMPLE

Cette méthode nous permet de faire une rotation de 180 dégrées en prenant f(e1)=-e1 et f(e2)=-e2. Mais cette méthode se limite à des angles multiples de 90 degrés comme 270 degrés.

Cette réflexion nous emmène au fait qu’il est possible d’élaborer une matrice nous permettant de faire la rotation d’un vecteur ou d’un point d’un angle α.

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 les vecteurs ont comme base l’origine, ce qui veut dire que les coordonnées du point d’arrivé des vecteurs sont également égale à leurs composantes.

Ici nous voyons le vecteur e1 et e2 rotationner d’un angle α. Ecrivons les coordonnées polaires des points d’arrivées des vecteurs puis, transformons-les en coordonnées cartésiennes. Et n’oublions pas qu’un angle de 90 degrés sépare le vecteur e1 et e2.

)

En trouvent les coordonnées des points d’arrivés, on trouve aussi les composantes des vecteur f(e1) et f(e2) qui sont égal aux coordonnées.

et

Une fois l’image des deux vecteurs trouvés, plus qu’à écrire la matrice qui va être égal à :

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

Un exemple s’impose :

Imagions le vecteur e1 au quel nous allons appliquer une rotation de 90°.Logiquement, nous allons tomber sur le vecteur e2. Commençons par définir la bonne matrice.

= La matrice Mf obtenue ici est cohérente avec celle déterminée plus haut à partir de la quatrième propriété des transformations linéaires.

**Rotation autour d’une autre référence**

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

Ainsi, 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 d’abord effectuer une translation pour ramener le centre de rotation à l’origine. Cette translation sera applique à le ou les points sensé rotationer. On effectue la rotation normalement. Enfin, on effectue la translation inverse.

Un exemple s’impose :

Imaginons un point de coordonné P(2;1) que nous voulons faire tourner de 90 degrés autour du point C(1;1) et non l’origine.

Pour ramener le point de centre de rotation à l’origine on va trouver une translation qu’on pourra aussi appliquer à P :

et

On obtient comme nouvelle coordonné On effectue la rotation de 90 degrés

A présent, faisons la translation inverse :

et

Le centre de rotation est à présent de nouveau en C(1;1) et on trouve P(1;2) après rotation.

Illustration

* Définition et propriétés
* Transformation d’un point dans le plan

**Coordonnées Homogènes en 2D**

* Introduction aux matrices 3×3
* Avantages pour les transformations combinées

**Chapitre 2 : Implémentation et Visualisation en 2D**

**1. Méthodes Numériques**

* Rotation par matrice de transformation et de rotation

**2. Programmation et Simulation**

* Implémentation en Python avec Matplotlib

**3. Applications et Exemples**

* Rotation d’objets géométriques (polygones, cercles)

**Partie 2 : Rotation en 3D**

**Chapitre 3 : Fondements Mathématiques de la Rotation en 3D**

**Introduction**

Nous allons maintenant évoluer dans un espace vectoriel en 3d, ce qui veut dire qu’aux axes oX et oy nous allons y ajouter l’axe oZ (V3). Les vecteurs auront 3 composantes et les les points auront des coordonnées selon les 3axes. On note V3 l’ensemble des vecteurs dans l’espace.

DEFINIR L’ESPACE ET LES VECTEURS DANS L’ESPACE

1. **Matrice de Rotation en 3D**

* Rotation autour des axes X, Y, Z
* Composition de rotations (angles d'Euler) et effet de Blocage de quadrant
* Partie calculatoire

**2. Quaternions et Rotation**

* Définition et propriétés
* Multiplication et interpolation de quaternions
* Partie calculatoire

**3. Démonstration de toutes choses avancées**

**Chapitre 4 : Implémentation et Visualisation en 3D**

**1. Programmation de la Rotation 3D**

* Implémentation avec matrices 4×4 (translation)
* Gestion des quaternions en Python
* Animation d’objets en 3D

**Conclusion et Perspectives**

**1. Résumé des Résultats Obtenus**

* Comparaison des méthodes utilisées en 2D et 3D
* Avantages et inconvénients des approches étudiées

**2. Perspectives et Travaux Futurs**

* Applications avancées (cinématique inverse, rendu photoréaliste)

1. Cours sur les matrices de M.Hochuli p.60 [↑](#footnote-ref-1)
2. Vidéo sur les transformations linéaire [20-Les transformations linéaires](https://www.youtube.com/watch?v=cAV0AVEt4ec) [↑](#footnote-ref-2)