

---

# Modélisation 3D Projet

---

ANDRES Baptiste  
ROTH Tom

24 Janvier 2023

Rapport créé avec la template latex ESIREM.

Enseignant : L. GARNIER

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Réalisation du projet</b>	<b>2</b>
2.1	Phase initiale . . . . .	2
2.2	Position et direction de la caméra . . . . .	2
2.3	Choix concernant la création de la trajectoire . . . . .	2
2.4	Courbe de Bézier . . . . .	2
2.5	Projection stéréographique . . . . .	2
2.6	Animation de la petite sphère $S_0$ . . . . .	3
2.7	Approximation de la courbe projetée à la surface de la sphère $S$ par une courbe de Bézier . . . . .	4
2.8	Difficultés rencontrées . . . . .	4

# 1 Introduction

L'objectif de ce projet consiste à réaliser l'animation d'une petite sphère  $S_0$  qui se déplace sur une sphère de centre  $O$ . Dans notre cas, la trajectoire de cette sphère suit une courbe définie par projection stéréographique de  $S$  où on ne garde que les points appartenant à la surface du cercle. Notre application fonctionne uniquement grâce à THREE.js. Ce projet a été l'occasion pour nous de revoir certains points théoriques abordés en cours mais aussi de nous familiariser davantage avec THREE.js.

## 2 Réalisation du projet

### 2.1 Phase initiale

Nous avons repris la base de code donnée lors des séances de TP pour réaliser notre projet.

Au début de notre projet, nous avons créé un repère orthonormé ainsi que les deux sphères avec les caractéristiques indiquées dans le sujet. Pour créer la petite sphère  $S_0$ , nous avons défini le centre de cette dernière comme étant un point appartenant à la sphère  $S$  de centre  $O$ .

L'apparence des deux sphères est modifiable grâce au menu GUI.

### 2.2 Position et direction de la caméra

Dans le menu GUI, il est possible de modifier la position et la direction de la caméra pour visualiser sous n'importe quel angle notre réalisation.

### 2.3 Choix concernant la création de la trajectoire

Nous avons ensuite choisi de réaliser la trajectoire par une courbe obtenue en plusieurs temps, comme expliqué dans le sujet. Nous avons tout de même tenté de réaliser la trajectoire avec l'autre méthode mais nous n'avons pas réussi à placer correctement l'arc de cercle de modélisé par une courbe de Bézier.

### 2.4 Courbe de Bézier

Une courbe de Bézier est définie par des points de contrôle. Elles peuvent être modifiées en changeant les positions des points de contrôle, ce qui permet une grande flexibilité dans la conception de formes complexes. L'ordre des courbes de Bézier dépend du nombre de point de contrôle utilisé.

### 2.5 Projection stéréographique

La projection stéréographique est une méthode de projection cartographique utilisée pour représenter des sphères (en trois dimensions) en un plan (en deux dimensions). Elle consiste à projeter les points sur la sphère sur un plan tangent à un point choisi sur la sphère, généralement le pôle Nord ou le pôle Sud. Dans notre cas, nous avons utilisé le pôle Sud.

Nous avons défini le pôle Nord comme étant le point  $(0, 0, R)$  où  $R = Rayon\ de\ la\ sphère\ S$  et le pôle Sud comme étant l'opposé du pôle Nord.

Voici ci-dessous, une figure illustrant cette projection :

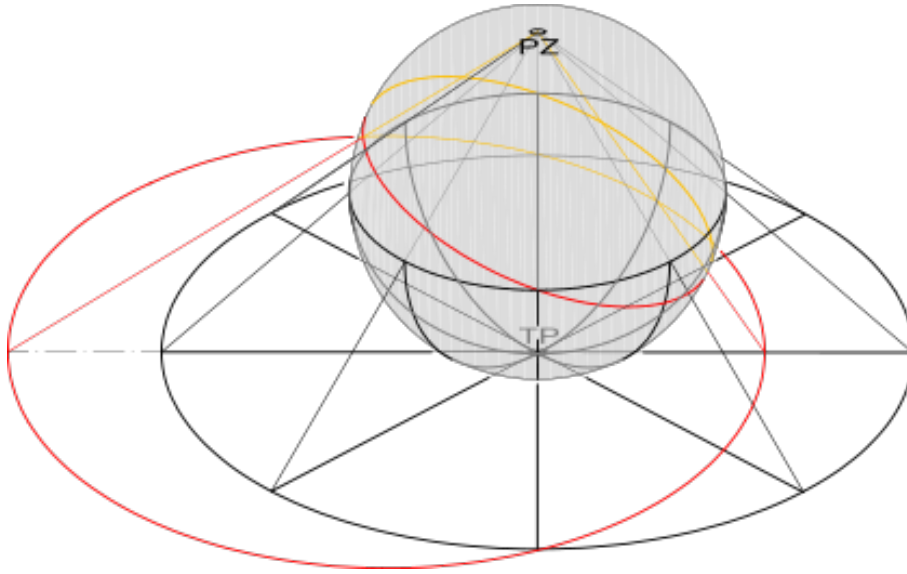


FIGURE 1 – Projection stéréographique - Principe

Dans le cadre de notre projet, la projection stéréographique nous a permis de tracer la courbe sur la sphère  $S$  qui a pour rôle de définir la trajectoire de notre petite sphère  $S_0$ .

Pour réaliser cette projection, nous avons créé un plan tangent à la sphère  $S$  au pôle Sud de cette sphère.

Nous avons ensuite tracé une courbe de Bézier avec des points de contrôle aléatoires appartenant à ce plan tangent qui nous servira ensuite pour réaliser notre projection. Il y a donc une contrainte sur la composante  $z$  des points définissant cette courbe de Bézier pour que celle-ci appartienne toujours à notre plan tangent. Les coordonnées  $(x, y)$  des points de contrôle sont modifiables dans le menu GUI.

Nous avons choisi de récupérer 100 points de cette courbe de Bézier pour ensuite déterminer les points d'intersections avec la sphère  $S$ . En effet, pour chacun des points de la courbe de Bézier, nous avons déterminé un point d'intersection dont les coordonnées sont stockées dans un tableau regroupant l'ensemble des points d'intersection trouvés.

Les points d'intersections obtenus forment ainsi la courbe qui servira de trajectoire à la petite sphère  $S_0$ .

## 2.6 Animation de la petite sphère $S_0$

Dans la fonction `renduAnim()`, nous avons modifié la position du centre de la petite sphère  $S_0$  au cours du temps. Son centre suit la trajectoire décrite par la courbe obtenue précédemment (affichée en rouge dans notre visualisation).

## 2.7 Approximation de la courbe projetée à la surface de la sphère $S$ par une courbe de Bézier

Nous n'avons pas réussi à réaliser cette partie mais nous avons néanmoins effectué des recherches et tenter certaines méthodes pour essayer d'y parvenir. La méthode qui semble la plus cohérente avec notre cas consiste en une approche basée sur la minimisation des erreurs. Elle consiste à trouver les points de contrôles qui minimisent la somme des erreurs entre les points obtenus précédemment formant la courbe sur la sphère  $S$  et les points de la courbe de Bézier que nous cherchons à déterminer.

Voici ci-dessous, une figure illustrant cette méthode :

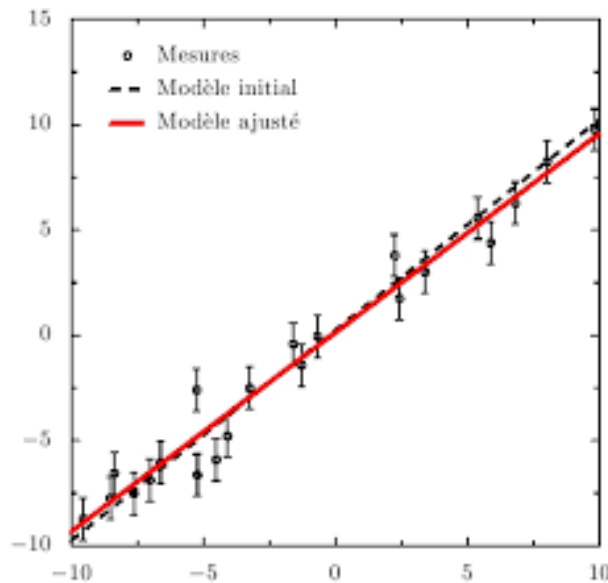


FIGURE 2 – Minimisation des erreurs - Principe

## 2.8 Difficultés rencontrées

Comme précisé dans la section précédente, nous avons eu des difficultés pour approximer la courbe obtenue par une courbe de Bézier.

De plus, nous avons éprouvé des difficultés lors de la réalisation du menu GUI pour modifier les paramètres précisés dans l'énoncé. En effet, nous avons écrit notre programme avec des valeurs fixes au départ, ainsi notre code ne s'adapte pas à toutes les configurations possibles. Par exemple, nous n'avons pas réussi à faire en sorte que la courbe de Bézier appartienne toujours au plan tangent de la sphère  $S$  qu'importe son équation. Nous nous sommes donc placés dans un cas particulier pour faciliter la programmation.