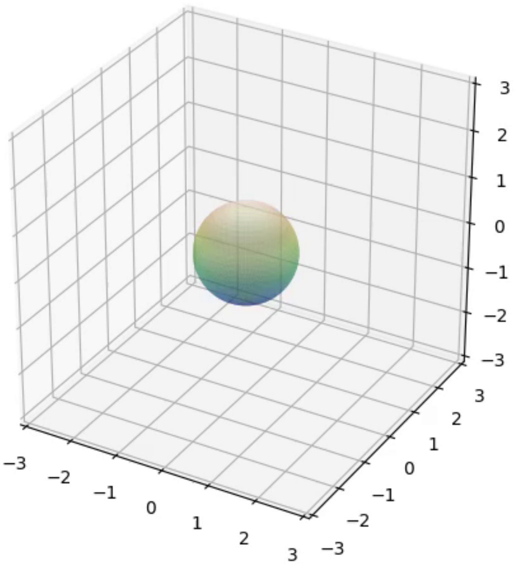
Práctica 5 - GCOMP

Celia Rubio Madrigal

6 de abril de 2022

Índice



1. Introducción

2. Material usado y metodología

2.1. Apartado *i*)

2.2. Apartado *ii*)

3. Resultados y conclusiones

3.1. Apartado *i*)

3.2. Apartado *ii*)

4. Código

```python
import os
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
import math

""" Apartado 1 """


def get_sphere(n, m):
    u = np.linspace(0, np.pi, n)
    v = np.linspace(0, 2 * np.pi, m)
    x = np.outer(np.sin(u), np.sin(v))
    y = np.outer(np.sin(u), np.cos(v))
    z = np.outer(np.cos(u), np.ones_like(v))
    return x, y, z


def plot_sphere(x, y, z, ax=None):
    if ax == None:
        plt.figure()
        ax = plt.axes(projection="3d")
    ax.set_xlim3d(-3, 3)
    ax.set_ylim3d(-3, 3)
    ax.set_zlim3d(-3, 3)
    ax.plot_surface(x, y, z, cmap="gist_earth", alpha=0.5, rstride=1, cstride=1)
    return ax


def plot_curve(x2, y2, z2, ax=None):
    if ax == None:
        plt.figure()
        ax = plt.axes(projection="3d")
    ax.set_xlim3d(-3, 3)
    ax.set_ylim3d(-3, 3)
    ax.set_zlim3d(-3, 3)
    ax.plot(x2, y2, z2, "-b", c="gray", zorder=3)
    ax.scatter(x2, y2, z2 + 0.01, c=x2 + y2, cmap="jet")
    return ax


x, y, z = get_sphere(30, 60)
ax = plot_sphere(x, y, z)

t2 = np.linspace(0.5, 1, 30)
x2 = abs(t2) * np.sin(t2 ** 3)
y2 = -abs(t2) * np.cos(2 * t2 ** 3)
```

```python
z2 = np.sqrt(1 - x2 ** 2 - y2 ** 2)
plot_curve(x2, y2, z2, ax)


def proj(x, y, z, alpha=1):
    eps = 1e-16
    aux = 1 / ((1 - z) ** alpha + eps)
    return aux * x, aux * y, 0 * z - 1


px, py, pz = proj(x, y, z, alpha=1 / 2)
px2, py2, pz2 = proj(x2, y2, z2, alpha=1 / 2)

plot_sphere(px, py, pz, ax)
plot_curve(px2, py2, pz2, ax)

plt.savefig("1")


""" Apartado 2 """

from matplotlib import animation


def proj2(x, y, z, t):
    aux = 2 / (2 * (1 - t) + (1 - z) * t)
    return aux * x, aux * y, -t + z * (1 - t)


def animate(t):
    xt, yt, zt = proj2(x, y, z, t)
    ax = plt.axes(projection="3d")
    plot_sphere(xt, yt, zt, ax)
    return (ax,)


def init():
    return (animate(0),)


fig = plt.figure(figsize=(6, 6))
ani = animation.FuncAnimation(
    fig, animate, np.arange(0, 1, 0.05), init_func=init, interval=20
)
ani.save("2.mp4", fps=5)
```