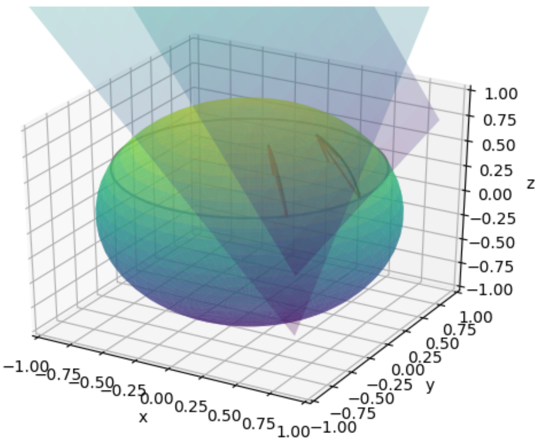
Práctica 8 - GCOMP

Celia Rubio Madrigal

13 de mayo de 2022

Índice



1. Introducción

En esta práctica estudiaremos y haremos gráficas de la traslación paralela de un vector sobre una superficie (pseudo)-Riemanniana suave: en nuestro caso, una esfera unitaria en el espacio tridimensional, $s_1^2 \subset \mathbb{R}^3$.

Una **traslación o transporte paralelo** de una sección X sobre una curva γ de una variedad como S_1^2 cumple que la derivada direccional de γ es 0. En este caso, un elemento de la fibra de esta variedad será un vector de un plano tangente sobre la esfera, y la curva escogida será un paralelo de la esfera.

2. Material usado y metodología

En coordenadas polares, si $q(\phi, \theta)$ es un punto de la esfera con $\phi \in [0, 2\pi)$ y $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, entonces la parametrización del paralelo θ_0 es $\gamma_{\theta_0}(\phi) = q(\phi, \theta_0)$.

El vector que vamos a transportar es $v_0 = (0, \frac{\pi}{5})$ colocado en el punto $q \in (0, \theta_0)$, con θ_0 variable.

Las ecuaciones del transporte paralelo de un vector v_0 pueden obtenerse de la condición, mencionada anteriormente, de que la derivada direccional sobre la curva γ es 0.

En función del ángulo ϕ, se obtiene un sistema lineal de ecuaciones diferenciales cuya solución es v, de componentes v^1 y v^2:

$$v^1(\phi) = v_0^2 \frac{\sin(\phi \sin(\theta_0))}{\cos(\theta_0)}$$

$$v^2(\phi) = v_0^2 \cos(\phi \sin(\theta_0))$$

2.1. Apartado *i*)

En el primer apartado, definimos una sucesión $f(t)$ en función de un parámetro temporal $t \in [0, 1]$, tales que $f(0) = v_0$ y $f(1)$ es la traslación de v_0 alrededor del paralelo entero ($\phi = 2\pi$). Además, la sucesión depende de t cuadráticamente.

Calcularemos cada elemento de la familia mediante la función `transp`, y adquiriremos, además, los puntos de origen donde colocar cada vector mediante la función `familia_param`, que llama a su vez a `transp`.

2.2. Apartado *ii*)

En el segundo apartado, escogeremos dos valores de θ_0. Esto define dos paralelos distintos, que dibujaremos en una gráfica mediante la función `paralelo`.

También colocaremos en la gráfica la malla de puntos que representa la esfera, como ya hemos hecho en prácticas previas. Es decir, discretizamos los puntos de la esfera $\in [0, 2\pi) \times [-\pi/2, \pi/2]$ en una malla de 60×30, y se realiza la siguiente transformación cartesiana:

$$x = \sin(u) \times \sin(v) \qquad y \quad = \sin(u) \times \cos(v) \qquad z = \cos(u) \times (1)_{i=1}^{|v|}$$

Con estos elementos constantes en cada fotograma, realizaremos una animación de la familia paramétrica $f(t)$ con los dos θ_0 escogidos, y el intervalo $t \in [0, 1]$ se discretizará de manera equidistante en 40 subintervalos. Para hacer la gráfica de cada vector, se llama a la función `get_bipunto`, y a la función, de la librería *matplotlib*, `quiver`.

3. Resultados y conclusiones

3.1. Apartado *i*)

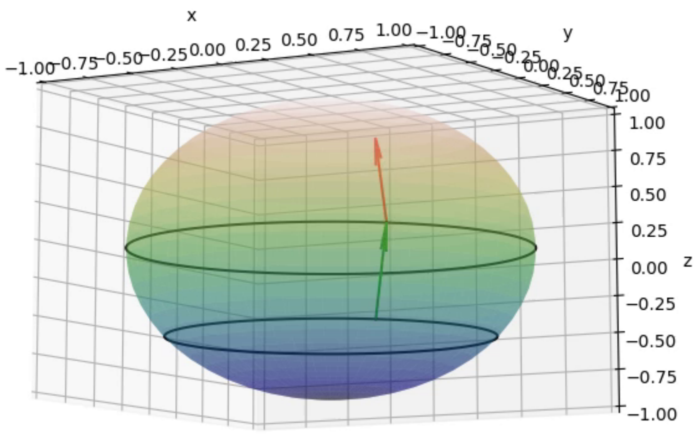
Como ya se ha descrito anteriormente, la familia escogida es $f(t) = v_t$, donde $v_t : [0, 2\pi] \to S_1^2$:

$$v_t \begin{pmatrix} \phi \\ \theta_0 \end{pmatrix} = v_0^2 \begin{pmatrix} \frac{\sin(\phi \sin(\theta_0) \cdot t^2)}{\cos(\theta_0)} \\ \cos(\phi \sin(\theta_0) \cdot t^2) \end{pmatrix}$$

3.2. Apartado *ii*)

Los dos valores de θ_0 escogidos son: $\theta_0^A = 0$ y $\theta_0^B = -0.2\pi$.

A continuación[1], y así como en los archivos adjuntos, se encuentra la animación de la sucesión de funciones paramétricas $f(t)$. El vector que recorre el paralelo θ_0^A es el rojo, y θ_0^B el azul. Como podemos observar, en el segundo caso el vector trasladado no se corresponde con el original.



Gracias a esta práctica hemos podido experimentar de primera mano cómo se define el transporte paralelo de vectores, que puede parecer trivial en el espacio $\mathbb{R}^n$, en otra variedad como es una esfera. Con ello hemos adquirido un mejor entendimiento de este objeto y de las curiosas propiedades que trae su curvatura.

[1]Para ver este gráfico en movimiento, se recomienda abrir este PDF en Adobe Reader u Okular.

4. Código

```
import numpy as np
from numpy import pi, cos, sin, sqrt, outer, ones
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import math

"""
2-esfera
"""
# latitudes
u = np.linspace(0, np.pi, 30)
# longitudes
v = np.linspace(0, 2 * np.pi, 60)

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))


"""
Vector en la variedad de la 2-esfera
Definimos un punto origen (o) y una dirección (p)
"""

o_phi_0 = 0 # longitud
o_theta_A = 0 # latitud del punto original. Por lo tanto es
    THETHA0 (!!)
o_theta_B = -0.2*pi # latitud del punto original. Por lo tanto es
     THETHA0 (!!)

p_phi_0 = 0 # longitud
p_theta_0 = pi / 5 # latitud
p_norm_0 = sqrt((p_phi_0 ** 2) * cos(p_theta_0) ** 2 + p_theta_0
    ** 2)

"""
Trasladamos paralelamente el bipunto anterior
"""

def transp(th0, ph, th, t, alpha=2):
    ph2 = th * np.sin((np.sin(th0)) * (ph) * (t**alpha)) / np.cos(
        th0)
    th2 = th * np.cos((np.sin(th0)) * (ph) * (t**alpha))
    return ph2, th2

def familia_param(o_theta_C, t, alpha=2):
    Dphi = math.tau
    o_phi2 = p_phi_0 + Dphi * (t**alpha)
```

```python
    o_theta2 = o_theta_C
    p_phi2, p_theta2 = transp(th0=o_theta_C, ph=Dphi, th=p_theta_0
        , t=t, alpha=alpha)
    return o_phi2, o_theta2, p_phi2, p_theta2


"""
CAMBIAMOS EL SISTEMA DE REFERENCIA PARA LA REPRESENTACIÓN
"""

phi0 = np.pi / 4

"""
VISTO COMO BIPUNTO
"""

def get_bipunto(o_phi2, o_theta2, p_phi2, p_theta2):
    o2 = np.array(
        [
            np.cos(o_theta2) * np.cos(o_phi2 - phi0),
            np.cos(o_theta2) * np.sin(o_phi2 - phi0),
            np.sin(o_theta2),
        ]
    )
    p2 = np.array(
        [
            np.cos(o_theta2 + p_theta2) * np.cos(o_phi2 + p_phi2 -
                phi0),
            np.cos(o_theta2 + p_theta2) * np.sin(o_phi2 + p_phi2 -
                phi0),
            np.sin(o_theta2 + p_theta2),
        ]
    )
    return np.concatenate((o2, p2 - o2))


"""
Curva (el paralelo) donde queremos transladar
"""


def paralelo(o_theta_C):
    phis = np.linspace(0, 2 * pi, 100)
    thetas = np.ones_like(phis) * o_theta_C
    return np.array(
        [
            np.cos(thetas) * np.cos(phis - phi0),
            np.cos(thetas) * np.sin(phis - phi0),
            np.sin(thetas),
        ]
```

```
    )


gamma_A = paralelo(o_theta_A)
gamma_B = paralelo(o_theta_B)

"""
FIGURA
"""

from matplotlib import animation


def animate(t):
    ax = plt.axes(projection="3d")
    ax.clear()
    fig.tight_layout()
    ax.plot_surface(x, y, z, rstride=1, cstride=1, alpha = 0.4,
                    cmap='gist_earth', edgecolor='none')
    ax.set_xlabel("x")
    ax.set_ylabel("y")
    ax.set_zlabel("z")
    ax.plot(gamma_A[0], gamma_A[1], gamma_A[2], "-b", c="black",
        zorder=3)
    ax.plot(gamma_B[0], gamma_B[1], gamma_B[2], "-b", c="black",
        zorder=3)
    ax.set_xlim(-1, 1)
    ax.set_ylim(-1, 1)
    ax.set_zlim(-1, 1)
    ax.view_init(-10, 300)

    o_phi_a, o_theta_a, p_phi_a, p_theta_a = familia_param(
        o_theta_A, t)
    o_phi_b, o_theta_b, p_phi_b, p_theta_b = familia_param(
        o_theta_B, t)
    X_A, Y_A, Z_A, U_A, V_A, W_A = get_bipunto(o_phi_a, o_theta_a,
         p_phi_a, p_theta_a)
    X_B, Y_B, Z_B, U_B, V_B, W_B = get_bipunto(o_phi_b, o_theta_b,
         p_phi_b, p_theta_b)
    plt.quiver(
        X_A,
        Y_A,
        Z_A,
        U_A,
        V_A,
        W_A,
        colors="red",
        zorder=3,
        color="red",
        arrow_length_ratio=0.3,
```

```
    )
    plt.quiver(
        X_B,
        Y_B,
        Z_B,
        U_B,
        V_B,
        W_B,
        colors="green",
        zorder=3,
        color="green",
        arrow_length_ratio=0.3,
    )
    return (ax,)


fig = plt.figure()
fig.tight_layout()
ax = plt.axes(projection="3d")


def init():
    ax = plt.axes(projection="3d")
    return (ax,)

ani = animation.FuncAnimation(
    fig, animate, np.linspace(0, 1, 40), init_func=init, interval
        =10
)
ani.save("2.mp4", fps=5)
```