

Transporte paralelo

Sergio González Montero (U2)

17 de mayo de 2024

1. Objetivo

Se pretende trasladar de forma paralela un vector dado tangente a la esfera unidad embebida en el espacio euclídeo. Se usarán longitudes $\phi = [0, 2\pi)$ y latitudes $\theta = [-\pi/2, \pi/2]$. El vector será $v_0 = (0, \pi/5) \in T_q(S_1^2)$, tangente a la variedad en $q = (0, \theta_0)$ a lo largo del paralelo $\theta = \theta_0$ en la esfera S_1^2 , trasladándolo de forma completa, es decir, $\phi \rightarrow 2\pi$.

2. Material y datos

Se usaron los siguientes apartados de las notas de Robert Monjo de la asignatura de Geometría Computacional, a día 17/05/2024:

- Sección 4.1.3. Espacio afín y difeomorfismos afines.
- Definición 4.2.3. Transporte paralelo.
- Ejemplo 4.2.1. Transporte paralelo en una esfera.
- Algoritmo 4.2.1. Transporte paralelo en $S_1^2 \subset \mathbb{R}^3$.

Las librerías usadas son las siguientes:

1. **numpy**: operaciones matemáticas y cálculos numéricos.
2. **matplotlib**: para visualización de gráficos y GIF.

3. Resultados

Se ha parametrizado la esfera de la forma $(x, y, z) = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)$. A partir de $q + v_0$, se traslada mediante la función:

$$f_t : [0, \frac{2}{\pi}] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \rightarrow S_1^2$$
$$\begin{pmatrix} \phi \\ \theta_0 \end{pmatrix} \mapsto \frac{\pi}{5} \begin{pmatrix} \frac{\sin(\sin(\theta_0)\phi t^2)}{\cos\theta_0} \\ \cos(\sin(\theta_0)\phi t^2) \end{pmatrix}$$

A continuación se presentan dos momentos de la animación del transporte paralelo:

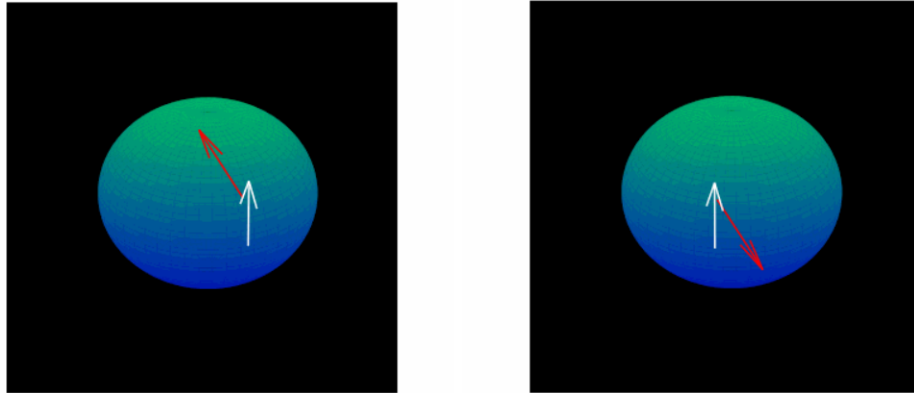


Figura 1: Distintos tiempos del transporte paralelo

4. Conclusión

Se observa que el desplazamiento del vector se ve acelerado conforme pasa el tiempo debido al parámetro cuadrático t^2 . Se puede apreciar también que el vector blanco del *GIF* adjunto tarda una revolución completa en volver a la posición inicial mientras que el vector rojo precisa del doble de revoluciones, variando su dirección en el transcurso del transporte.

5. Código

Programa 1: transporte paralelo.py

```
1  # -*- coding: utf-8 -*-
2
3  # Sergio Gonzalez Montero
4
5  import numpy as np
6  from numpy import cos as cos, sin as sin, pi as pi
7  import matplotlib.pyplot as plt
8  from matplotlib import animation
9
10
11 def sph2cart(phi, theta):
12     """
13     phi : float; longitud de [0, 2*pi)
14     theta : float; latitud de [-pi/2, pi/2]
15     De coordenadas esfericas a cartesianas
16     """
17     x = cos(theta)*cos(phi)
18     y = cos(theta)*sin(phi)
19     z = sin(theta)
20
21     tocart = np.array([x,y,z])
```

```

22     return tocart
23
24 def cartTgPlanePt(o, o_phi, o_theta, v_phi, v_theta):
25     """
26     o : lst; punto en cartesianas
27     o_phi : float; longitud de o
28     o_theta : float; latitud de o
29     v_phi, v_theta : float; componentes tangenciales
30     Coordenadas cartesianas del punto o en el plano tangente
31     """
32
33     a = np.array([-cos(o_theta)*sin(o_phi),
34                  cos(o_theta)*cos(o_phi),
35                  0])
36     b = np.array([-sin(o_theta)*cos(o_phi),
37                  -sin(o_theta)*sin(o_phi),
38                  cos(o_theta)])
39     cart_coord = o + v_phi * a + v_theta * b
40
41     return cart_coord
42
43 #Familia parametrica
44 def paramFam(t, phi, theta0, v02):
45     """
46     t : float; parametro temporal
47     phi : float; angulo acimutal
48     theta0 : float; angulo polar
49     Familia param trica dependiente de t^2
50     """
51     p_phi = v02/cos(theta0) * sin(sin(theta0)*phi*t**2)
52     p_theta = v02 * cos(sin(theta0)*phi*t**2)
53
54     return p_phi, p_theta
55
56
57 def animate(t, theta01, theta02, v02, xsph, ysph, zsph):
58     ax = plt.axes(projection = '3d')
59
60     # Vector blanco
61     o_phi1 = 2*pi*t**2
62     o_theta1 = theta01
63     v_phi1, v_theta1 = paramFam(t, 2*pi, theta01, c0)
64
65     o1 = sph2cart(o_phi1, o_theta1)
66     p1 = cartTgPlanePt(o1, o_phi1, o_theta1, v_phi1, v_theta1)
67     X1, Y1, Z1, U1, V1, W1 = np.concatenate((o1, p1-o1))
68
69     # Vector rojo
70     o_phi2 = 2*pi*t**2
71     o_theta2 = theta02
72     v_phi2, v_theta2 = paramFam(t, 2*pi, theta02, c0)
73
74     o2 = sph2cart(o_phi2, o_theta2)
75     p2 = cartTgPlanePt(o2, o_phi2, o_theta2, v_phi2, v_theta2)
76     X2, Y2, Z2, U2, V2, W2 = np.concatenate((o2, p2-o2))
77

```

```

78     # Esfera y vectores
79     ax.plot_surface(xsph, ysph, zsph, cmap='winter',
80                    edgecolor='none',alpha=0.5)
81     ax.quiver(X1,Y1,Z1,U1,V1,W1,colors="white", zorder=3,
82              arrow_length_ratio=0.4)
83     ax.quiver(X2,Y2,Z2,U2,V2,W2,colors="red", zorder=3,
84              arrow_length_ratio=0.4)
85
86     ax.set_axis_off()
87     ax.set_facecolor('black')
88
89
90     # Sistema de referencia
91     phi = np.linspace(0, 2*pi, 3000)
92     theta = np.linspace(-pi/2, pi/2, 2000)
93     r = 1
94     x = r*np.outer(cos(theta), cos(phi))
95     y = r*np.outer(cos(theta), sin(phi))
96     z = r*np.outer(sin(theta), np.ones_like(phi))
97
98
99     # Animacion
100    theta01 = 0
101    theta02 = pi/6
102    v02 = pi/5
103    c0 = np.sqrt(pi/5)
104
105    fig = plt.figure(figsize=(6,6))
106    ani = animation.FuncAnimation(fig,animate,
107                                frames=np.arange(0,2.01,0.025),
108                                fargs=(theta01,theta02,v02,x,y,z),
109                                interval=20)
110    ani.save('transporte paralelo.gif',fps=10)
111    plt.show()

```