

PRÁCTICA 5: DEFORMACIÓN DE VARIEDADES DIFERENCIABLES

Autor

Compañera de código

Martín Fernández de Diego

Belén Sánchez Centeno

1. INTRODUCCIÓN

Sea S_1^2 la variedad diferencial correspondiente a una 2-esfera de radio unitario embebida en \mathbb{R}^3 . Se pretende representar gráficamente el difeomorfismo de una proyección estereográfica. Para ello, extraeremos el punto $e_3 := (0, 0, 1) \in S_1^2$ y utilizaremos las coordenadas cartesianas $(x, y, z) \in [-1, 1]^3$.

Apartado i) Estimar y representar una malla regular de puntos de S_1^2 con una resolución (paso de malla) que proporcione 30 valores de latitud ($\phi \in [0, \pi)$) y 60 valores de longitud ($\varphi \in [0, 2\pi)$). Estimar y representar la imagen de la proyección estereográfica $\Pi : S_1^2/e_3 \rightarrow \mathbb{R}^2$ con $\alpha = 1/2$.

Nota: Diseñar una curva sobre S_1^2 para comprobar cómo se deforma cuando se proyecta en \mathbb{R}^2 .

Apartado ii) Obtener una animación de al menos 20 fotografías (imágenes) de la siguiente familia paramétrica:

$$f_t : S_1^2/e_3 \rightarrow \mathbb{R}^3$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \frac{2}{2 - 2t + (1 - z)t} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -t + z(1 - t) \end{pmatrix}$$

con $t \in [0, 1]$, $\lim_{t \rightarrow 1} f_t = \Pi$ y siendo $p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, $f_0(p) = p$ función identidad.

2. MATERIAL USADO

A partir de la plantilla proporcionada por el profesor, son necesarios algunos cambios que se describen a continuación.

Sobre la superficie de la esfera se define una nueva curva

$$\begin{aligned} x(t) &= |t| \sin^2(40 \frac{t}{2}) \\ y(t) &= |t| \cos^2(40 \frac{t}{2}) \\ z(t) &= \sqrt{1 - x^2 - y^2} \end{aligned}$$

con $t \in [0, 1]$.

2.1. Apartado i)

Para calcular y representar la nueva malla regular se deben modificar las resoluciones de las coordenadas polares u y v de 25 a 30 puntos y de 50 a 60 respectivamente.

Inmediatamente, las coordenadas polares se transforman a cartesianas con

$$\begin{aligned} x &:= \sin \phi \times \sin \varphi \\ y &:= \cos \phi \times \cos \varphi \\ z &:= \cos \phi \times (1)_n \end{aligned}$$

con $n = \dim(\varphi)$.

Para calcular la proyección estereográfica de S_1^2 que manda el norte al infinito, se hace uso de la función $\text{proj}(x, z, z_0, \alpha)$ de la plantilla que realiza la transformación

$$\begin{aligned} x &\mapsto \frac{x}{|1 - z|^\delta} \\ y &\mapsto \frac{y}{|1 - z|^\delta} \\ z &\mapsto 1 \end{aligned}$$

con $\epsilon > 0$ para evitar dividir entre nulo y $\delta = \alpha \cdot \epsilon$.

Ahora se representa la gráfica de la malla esférica y de su proyección con la correspondiente curva de ayuda a la corrección.

2.2. Apartado ii)

La familia paramétrica de $f_t(x, y, z)$ define una transformación continua para $t \in [0, 1]$. Si $t = 0$, $f_0(x, y, z) = (x, y, z)$. Si $t \rightarrow 1$, $\lim_{t \rightarrow 1} f_t(x, y, z) = \Pi$ y, se calcula rápidamente, $\Pi = \{z = -1\}$.

La función `animate(t)` se encarga de generar los fotografías que formarán parte de la animación de la transformación. Recibe instantes de tiempo t y devuelve el estado de la transformación en ese punto.

3. RESULTADOS

3.1. Apartado i)

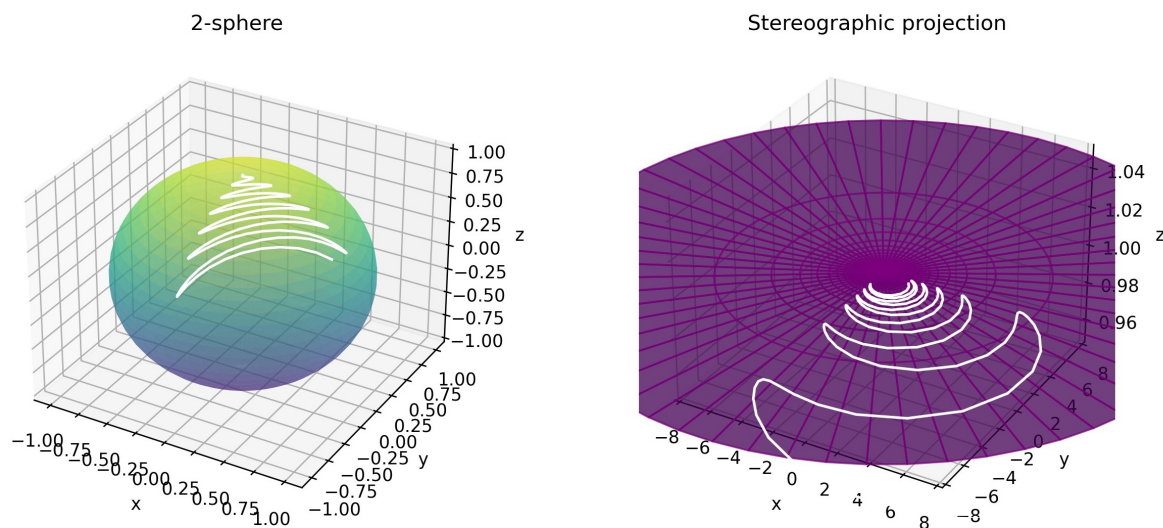


Fig. 1. Proyección estereográfica sobre el plano $z = 1$ envía el polo superior al infinito.

3.2. Apartado ii)

La animación correspondiente a la familia paramétrica $f_t(x, y, z)$ formada por 20 fotografías, aparece adjunta al documento.

4. CONCLUSIÓN

Véase cómo se deforma la curva de control de corrección. En la figura de la izquierda podemos observar cómo la amplitud de oscilación de dicha curva disminuye a medida que se acerca al polo superior. Sin embargo, en la figura de la derecha, con la deformación ya aplicada, la curva aumenta su amplitud a medida que se acerca al lugar que ocupaba el polo superior.

Es importante matizar que el polo superior no se encuentra en la figura de la derecha pero podría colocarse de manera intuitiva en el *infinito*.

Esta deformación es muy similar a la que sufren los mapas cartográficos terrestres.

5. ANEXO CON EL SCRIPT Y CÓDIGO UTILIZADO

5.1. Código

```
1  """
2  PRÁCTICA 5: DEFORMACIÓN DE VARIEDADES DIFERENCIABLES
3  Belén Sánchez Centeno
4  Martín Fernández de Diego
5  """
6
7  #import os
8  import numpy as np
9  import matplotlib.pyplot as plt
10 #from mpl_toolkits.mplot3d import axes3d
11
12 from matplotlib import animation
13 #from mpl_toolkits.mplot3d.axes3d import Axes3D
14
15 """
16 Dadas las coordenadas de x y z
17 devuelve la proyección de x sobre el eje z
18 z0 = 1 porque el polo extraído es el (0,0,1)
19 """
20 def proj(x,z,z0=1,alpha=1):
21     z0 = z*0+z0
22     eps = 1e-16
23     x_trans = x/(abs(z0-z)**alpha+eps)
24     return(x_trans)
25     # Nótese que añadimos un épsilon para evitar dividir entre 0
26
27 """
28 Animación del APARTADO ii)
29 """
30 def animate(t):
31     xt = 2/(2*(1-t) + (1-z)*t + eps)*x
32     yt = 2/(2*(1-t) + (1-z)*t + eps)*y
33     zt = (-1)*t + z*(1-t)
34     x2t = 2/(2*(1-t) + (1-z2)*t + eps)*x2
35     y2t = 2/(2*(1-t) + (1-z2)*t + eps)*y2
36     z2t = (-1)*t + z2*(1-t)
37
38     ax = plt.axes(projection='3d')
39     ax.set_zlim3d(-1,1)
40     ax.set_xlabel('x')
41     ax.set_ylabel('y')
42     ax.set_zlabel('z')
43     ax.plot_surface(xt, yt, zt, rstride=1, cstride=1, alpha=0.5, cmap='viridis',
44                     edgecolor='none')
45     ax.plot(x2t, y2t, z2t, '-b', c="white", zorder=3)
46     return ax
47
48 def init():
49     return animate(0)
50
51
52 # FORMATO
53 class Formato:
54     BOLD = "\033[1m"
55     RESET = "\033[0m"
56
57
58
59 # APARTADO i)
60 print("\n" + Formato.BOLD + "Apartado i)" + Formato.RESET)
61
62 """
63 2-esfera
64 definición en polares
65 """
```

```

66 u = np.linspace(0, np.pi, 30)
67 v = np.linspace(0, 2 * np.pi, 60)
68
69 """
70 2-esfera
71 transformación en paramétricas
72 """
73 x = np.outer(np.sin(u), np.sin(v))
74 y = np.outer(np.sin(u), np.cos(v))
75 z = np.outer(np.cos(u), np.ones_like(v))
76
77 """
78 gamma-curva
79 definición en paramétricas
80 """
81 t2 = np.linspace(0.001, 1, 200)
82
83 x2 = abs(t2) * np.sin(40 * t2/2)**2
84 y2 = abs(t2) * -np.cos(40 * t2/2)**2
85 z2 = np.sqrt(1-x2**2-y2**2)
86
87 # Representación: 2-esfera + 2-esfera proyectada
88 z0 = 1
89
90 fig = plt.figure(figsize=(12,12))
91 fig.subplots_adjust(hspace=0.4, wspace=0.2)
92
93 ax = fig.add_subplot(2, 2, 1, projection='3d')
94 ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='viridis', alpha=0.5, edgecolor='
    none')
95 ax.plot(x2, y2, z2, '-b', c="white", zorder=3)
96
97 ax.set_title('2-sphere');
98 ax.set_xlabel('x')
99 ax.set_ylabel('y')
100 ax.set_zlabel('z')
101
102 ax = fig.add_subplot(2, 2, 2, projection='3d')
103 ax.set_xlim3d(-8,8)
104 ax.set_ylim3d(-8,8)
105
106 ax.plot_surface(proj(x, z, z0), proj(y, z, z0), z*0+z0, rstride=1, cstride=1, cmap='
    viridis', alpha=0.5, edgecolor='purple')
107 ax.plot(proj(x2, z2, z0), proj(y2, z2, z0), z0, '-b', c="white", zorder=3)
108
109 ax.set_title('Stereographic projection');
110 ax.set_xlabel('x')
111 ax.set_ylabel('y')
112 ax.set_zlabel('z')
113
114 plt.show()
115 fig.savefig('stereo2.png', dpi=250)
116 plt.close(fig)
117
118
119
120 # APARTADO ii)
121 print("\n" + Formato.BOLD + "Apartado ii)" + Formato.RESET)
122
123 """
124 2-esfera proyectada - familia paramétrica
125 """
126 t = 0.1
127 eps = 1e-16
128
129 # Representación: animación
130 fig = plt.figure(figsize=(6, 6))
131 ani = animation.FuncAnimation(fig, animate, np.arange(0, 1, 0.05), init_func=init,
    interval=20)
132 ani.save("ejemplo.gif", fps = 5)

```

```
133 plt.close(fig)
```