Práctica 5: Difeomorfismos de variedades diferenciables

Guillermo Martín Sánchez

11/04/20

1 Introducción

El objetivo de esta práctica es la representación gráfica de la variedad diferencial S_1^2: la 2-esfera de radio unidad embebida en $\mathbb{R}^3$. También queremos representar una curva γ sobre esta variedad y la imagen de ambas resultado de aplicar la proyección estereográfica $\Pi : S_1^2 \setminus e_3 (:= (0,0,-1)) \to \mathbb{R}^2$ con $\alpha = 1$. Finalmente, hacemos una animación en la que vemos las imágenes de la 2-esfera y la curva a través de las funciones de una familia paramétrica que cumple que $\lim_{t\to 1} f_t = \Pi$ y $f_0 = Id$.

2 Material usado

2.1 Ejercicio 1

Estima y representa una malla regular de puntos de S_1^2 con una resolución (paso de malla) que proporcione 25 valores de latitud ($\phi \in [0,\pi)$) y 50 valores de longitud ($\psi \in [0, 2\pi)$). Estima y representa la imagen de la proyección estereográfica $\Pi : S_1^2 \setminus e_3 \to \mathbb{R}^2$ con $\alpha = 1$ (ver teoría). Diseña una curva sobre S_1^2 para comprobar cómo se deforma cuando se proyecta en $\mathbb{R}^2$.

Para cada punto de la malla (ϕ, ψ) hemos calculado las coordenadas (x, y, z) de la siguiente manera:

$$S_1^2 : \begin{cases} x = sin(\phi) \cdot sin(\psi) \\ y = sin(\phi) \cdot cos(\psi) \\ z = cos(\phi) \end{cases} \qquad \phi \in [0,\pi) \quad \psi \in [0, 2\pi) \tag{1}$$

Las ecuaciones paramétricas de la curva γ son:

$$\gamma : \begin{cases} x(t) = 0.32 \cdot (2.1 \cdot cos(10t) - cos(21t)) \\ y(t) = 0.32 \cdot (2.1 \cdot sin(10t) - sin(21t)) \\ z(t) = -\sqrt{1 - (x^2(t) + y^2(t))} \end{cases} \qquad t \in [0, 2\pi) \tag{2}$$

Finalmente la proyección estereográfica Π con $\alpha = 1$ tiene la ecuación:

$$\Pi \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{x}{|-1-z|} \\ \frac{y}{|-1-z|} \\ 1 \end{pmatrix} \qquad (x, y, z) \in S_1^2 \setminus e_3 \tag{3}$$

2.2 Ejercicio 2

Obtén una animación de al menos 15 fotogramas (imágenes) de la familia paramétrica en Eq.4

La familia paramétrica son las funciones $f_t : S_1^2 \setminus e_3 \to \mathbb{R}^2$ definidas por

$$f_t \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{(1-t) + |-1-z|t} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ (-1)t + z(1-t) \end{pmatrix} \tag{4}$$

con $t \in [0, 1]$. Para 20 valores de t entre 0 y 1, representamos $f_t(S_1^2 \setminus e_3)$ y $f_t(\gamma)$ haciendo así una animación. Se puede observar que $\lim_{t\to 1} f_t = \Pi$ y $f_0 = Id$.

3 Resultados y discusión

3.1 Ejercicio 1

En la parte izquierda de la Figura 1 vemos la variedad diferencial S_1^2 y la curva γ en su superficie representadas en $\mathbb{R}^3$. En la derecha, vemos sus imágenes de la proyección estereográfica, es decir, $\Pi(S_1^2 \setminus e_3)$ y $\Pi(\gamma)$. Como se puede ver, $S_1^2 \setminus e_3$ es homeomorfo a $\mathbb{R}^2$ y resulta curioso ver la transformación de γ: los "picos" que antes apuntaban hacia e_3 ahora apuntan hacia afuera, hacia el infinito, y, la "circunferencia" que antes hacía la curva en el ecuador ahora es alrededor del punto central del plano correspondiente a $\Pi(0, 0, 1)$

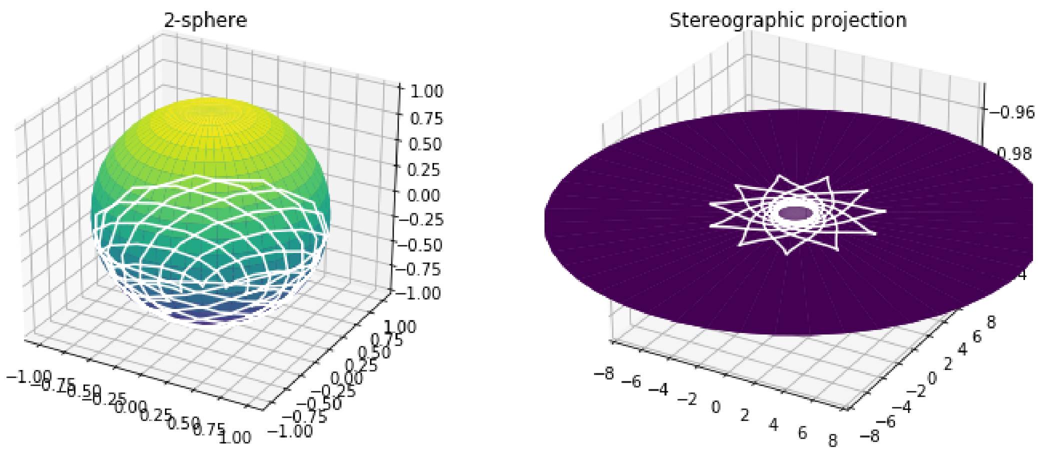

Figure 1: A la izquierda, S_1^2 y γ. A la derecha, $\Pi(S_1^2 \setminus e_3)$ y $\Pi(\gamma)$

3.2 Ejercicio 2

En la Figura anexa *exercise2.gif* vemos el resultado de animar las imágenes de la familia paramétrica. Como era de esperar en $t = 0$ obtenemos la identidad, es decir, una imagen igual que la izquierda de la Figura 1 y, conforme $t \to 1$ nos vamos acercando poco a poco a una imagen igual que la derecha de esa misma figura.

4 Conclusión

Hemos visto como se puede representar gráficamente una variedad diferencial a partir de sus coordenadas y un mallado. Hemos visto también un ejemplo de difeomorfismo entre variedades diferenciales y como animar una familia paramétrica de transformaciones de estas variedades.

5 Código

```
"""
Coursework 5: Diffeomorphisms
"""

import os
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import animation
from mpl_toolkits.mplot3d import Axes3D
```

```python
vuestra_ruta = "C:/Users/guill/Documents/Carrera/GEOComp/5-Diffeomorphism"

os.getcwd()
os.chdir(vuestra_ruta)

#%%
"""
Exercise 1: 2-sphere projected
"""

# Stereographic projection to plane z0 (=-1)
def proj(x,z,z0=-1,alpha=1):
    z0 = z*0+z0
    eps = 1e-16
    x_trans = x/(abs(z0-z)**alpha+eps)
    return(x_trans)


u = np.linspace(0, np.pi, 25)
v = np.linspace(0, 2 * np.pi, 50)

# Parametric equations of 2-sphere
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))

# Parametric equations of curve
t2 = np.linspace(0, 2*np.pi, 200)
x2 = 0.32*(2.1*np.cos(10*t2)-np.cos(21*t2))
y2 = 0.32*(2.1*np.sin(10*t2)-np.sin(21*t2))
z2 = -np.sqrt(1-(x2**2+y2**2))


fig = plt.figure(figsize=(12,12))
fig.subplots_adjust(hspace=0.4, wspace=0.2)

# Plot the 2-sphere and the curve
ax = fig.add_subplot(2, 2, 1, projection='3d')
ax.plot_surface(x, y, z, rstride=1, cstride=1,
                cmap='viridis', edgecolor='none')
ax.plot(x2, y2, z2, '-b',c="white",zorder=3)
ax.set_title('2-sphere');

# Plot the projection of the 2-sphere and the curve
ax = fig.add_subplot(2, 2, 2, projection='3d')
```

```
ax.set_xlim3d(-8,8)
ax.set_ylim3d(-8,8)
ax.plot_surface(proj(x,z), proj(y,z), z*0-1, rstride=1, cstride=1,
                cmap='viridis', edgecolor='none')
ax.plot(proj(x2,z2), proj(y2,z2), -1, '-b',c="white",zorder=3)
ax.set_title('Stereographic projection');

plt.show()
plt.close(fig)

#%%
"""
Exercise 2: Parametric transformation
"""

# Plot a frame of the parametric transformation at time t
def animate(t):
    frac = 1/((1-t) + np.abs(-1-z)*t)
    xt = frac*x
    yt = frac*y
    zt = -t + z*(1-t)

    frac2 = 1/((1-t) + np.abs(-1-z2)*t)
    x2t = frac2*x2
    y2t = frac2*y2
    z2t = -t + z2*(1-t)

    ax = plt.axes(projection='3d')
    ax.set_xlim3d(-2,2)
    ax.set_ylim3d(-2,2)
    ax.set_zlim3d(-2,2)
    ax.plot_surface(xt, yt, zt, rstride=1, cstride=1,
                    cmap='viridis', edgecolor='none')
    ax.plot(x2t,y2t, z2t, '-b',c="white",zorder=3)
    return ax,

# Plot the first frame
def init():
    return animate(0),

# Save a .gif with the animation of the parametric transformation
fig = plt.figure(figsize=(6,6))
ani = animation.FuncAnimation(fig, animate, np.arange(0,1,0.05), init_func=init,
                              interval=20)
ani.save("exercise2.gif", fps = 5)
```

```python
#%%
"""
Exercise 3: Another parametric transformation
"""

# Plot a frame of the parametric transformation at time t
def animate(t):
    frac = np.tan(np.arctan(1-t) + t*(np.pi*(-z+1)/4))
    xt = frac*x
    yt = frac*y
    zt = -t + z*(1-t)

    frac2 = np.tan(np.arctan(1-t) + t*(np.pi*(-z2+1)/4))
    x2t = frac2*x2
    y2t = frac2*y2
    z2t = -t + z2*(1-t)

    ax = plt.axes(projection='3d')
    ax.set_xlim3d(-2,2)
    ax.set_ylim3d(-2,2)
    ax.set_zlim3d(-2,2)
    ax.plot_surface(xt, yt, zt, rstride=1, cstride=1,
                    cmap='viridis', edgecolor='none')
    ax.plot(x2t,y2t, z2t, '-b',c="white",zorder=3)
    return ax,

# Plot the first frame
def init():
    return animate(0),

# Save a .gif with the animation of the parametric transformation
fig = plt.figure(figsize=(6,6))
ani = animation.FuncAnimation(fig, animate, np.arange(0,1,0.05), init_func=init,
                              interval=20)
ani.save("exercise3.gif", fps = 5)
```