



"El saber de mis hijos
hará mi grandeza"

UNIVERSIDAD DE SONORA

DEPARTAMENTO DE MATEMÁTICAS

Programa de Licenciatura en Matemáticas

Geometría Diferencial: un enfoque computacional con Python

T E S I S

Que para obtener el título de:

Licenciado en Matemáticas

Presenta:

Oscar Andrés Fraijo Domínguez

Director de tesis: Dr. Misael Avendaño Camacho

Hermosillo, Sonora, México, 17 de diciembre de 2025

SINODALES

Dr. Misael Avendaño Camacho

Universidad de Sonora, Hermosillo, México

Dr. José Crispín Ruiz Pantaleón

Universidad de Sonora, Hermosillo, México

Dr. Gerardo Mauricio Toledo Acosta

Universidad de Sonora, Hermosillo, México

Dr. Eduardo Velasco Barreras

Universidad de Sonora, Hermosillo, México

*A quienes me acompañaron en esta odisea.
Especialmente a mi padre, mi madre y mi hermana ...*

Agradecimientos

Indudablemente resulta natural agradecer primeramente a mis padres y mi hermana, que siempre me han apoyado y acompañado desde que tengo uso de razón. De no ser por ellos, no habría llegado hasta este punto de mi vida. No sólo este trabajo es fruto de mi esfuerzo, sino producto del suyo también.

Prosiguiendo con el cometido, expreso por este medio el más profundo agradecimiento y cariño que le tengo a la comunidad del departamento de matemáticas de la Universidad de Sonora, incluyendo así a los profesores, compañeros y resto del personal. A profesores como el Dr. José Montoya; Dr. Adrián Acuña, Dr. Héctor Hernández y la Dra. Marysol Navarro, quienes me aconsejaron, apoyaron y brindaron oportunidades de crecimiento. A estimados compañeros como José Hugo, Jesús Ángel, Sebastian, Saúl, Kevin, Alan, Mauricio, Victoria, Damaris, Jazmín y demás con quienes llevé clases, que de no ser por ustedes y la cálida cotidianidad en la que convivimos, los días durante la licenciatura no habrían sido tan amenos y buenos como lo fueron.

He de hacer especial reconocimiento a la Dra. Carolina Espinoza, quien me acogió y apoyó desde que tuve el honor de ser su alumno. Al Dr. Guillermo Dávila, por ser modelo de elegancia, claridad, y elocuencia; de quien aprendí el gusto por utilizar $\text{\LaTeX}$ y la finura de la escritura. A mi director de tesis, el Dr. Misael Avendaño, quien ha practicado la sagrada virtud de la paciencia durante este proceso y ha tenido la confianza en mis habilidades y la certeza en que yo terminaría este trabajo, legado de mis años de estudio durante la licenciatura.

Así como carecería de sentido y resultara inconcebible la travesía de Frodo, del Señor de los Anillos, sin sus amigos y compañía; seguramente así habría sido un imposible este trayecto, de no ser por el apoyo, ánimos y confianza que mis amigos me han brindado a lo largo de ésta. Agradecido de haber conocido a colegas matemáticos de otros lares durante las escuelas en las que tuve el agrado de asistir.

Casi para finalizar, agradecer a mis sinodales que revisaron esta tesis, tomaron el tiempo de leerla y corregirla. Agradecer al lector de ésta por su tiempo y curiosidad. Ahora sí, finalmente, gracias a Dios.

Índice

Agradecimientos	iii
Índice general	iii
Introducción	1
1. Un tutorial introductorio a Python	2
1.1. Recomendaciones	3
1.2. Instalación para Windows	3
1.3. Instalación para <i>macOS</i>	4
1.4. Estructura básica de Manim	5
1.5. Contenido adicional	6
2. Geometría de curvas en $\mathbb{R}^2$ y $\mathbb{R}^3$	10
2.1. Curvas parametrizadas diferenciales en $\mathbb{R}^n$	10
2.2. Longitud de arco y parámetro natural de curvas	12
2.2.1. Curvas dinámicas	15
2.3. Teoría local de curvas en $\mathbb{R}^2$	18
2.3.1. Teorema Fundamental de Curvas en $\mathbb{R}^2$	28
2.4. Teoría local de curvas en $\mathbb{R}^3$	30
2.4.1. Curvatura y torsión	31
2.4.2. Teorema Fundamental de Curvas en $\mathbb{R}^3$	34
2.4.3. Aparato de Frenet para curvas de parámetro arbitrario	36
2.4.4. Fórmulas para curvatura y torsión de parámetro arbitrario	37
3. Superficies regulares en $\mathbb{R}^3$	38
3.1. Definición de superficies regulares y ejemplos	38
3.2. El plano tangente de superficies regulares	46
3.3. Primera forma fundamental, longitudes, ángulos y áreas	50
3.4. La aplicación de Gauss y segunda forma fundamental	53
3.5. Superficies minimales o de curvatura media constante	62
4. Curvas geodésicas y sus propiedades	64
4.1. Isometrías y aplicaciones conformes	64

4.2. Curvatura geodésica y curvas geodésicas	73
4.3. Ecuaciones geodésicas y la relación de Clairaut	80

Bibliografía	84
---------------------	-----------

Introducción

Para todo matemático del siglo XXI, hoy en día resulta indispensable programar. El propósito de este trabajo es presentar y dotar de un enfoque computacional a la geometría diferencial, utilizando el lenguaje de programación *Python*. En esta tesis se presentan desarrollos teóricos de la teoría fundamental de curvas y superficies, los cuales se ilustran con el uso de *Manim*, una librería endémica a *Python*. Se ha de precisar que se desarrollaron algoritmos que permiten calcular, en algunos casos, mediante la resolución numérica de ecuaciones diferenciales, objetos geométricos fundamentales, como la curvatura y la torsión; las cuales se utilizan en los Teoremas Fundamentales de curvas (en el plano o en el espacio), por ejemplo.

La librería *Manim* no sólo sirve para generar imágenes en formato *.png*, sino que también puede producir videos en formato *.mp4*, además de permitir trabajar con otro tipo de recursos, como las imágenes vectoriales (*.svg*). El énfasis se debe a que estos pueden resultar de gran apoyo visual para la docencia, difusión e investigación de las matemáticas. Sólo en algunos casos se mostrará el código esencial detrás de la imagen presentada, ya que en ocasiones es extenso. Sin embargo, todos los códigos están disponibles en un repositorio de GitHub y quien desee consultarlos y ejecutarlos puede acceder desde la siguiente liga: <https://github.com/Fraijo/Geometria-Diferencial-un-enfoque-computacional-con-Python>

La estructura por capítulos del presente trabajo es la siguiente. En el Capítulo 1 se muestra un detallado tutorial de cómo realizar la instalación del software que se emplea en los sistemas operativos de *Windows* y *macOS*. En el Capítulo 2 trata de curvas en el plano y el espacio, su definición formal; propiedades, teoremas fundamentales. Se introduce a conceptos como curvatura y torsión. En el Capítulo 3 trata de las superficies y todo lo que nos sea posible abarcar respecto a ellas. Veremos que hay conceptos análogos entre estas y las curvas estudiadas en el Capítulo 2. Por último, en el Capítulo 4 se extiende la noción superficies y se estudian sus propiedades intrínsecas.

Capítulo 1

Un tutorial introductorio a Python

Python es un lenguaje de programación de alto nivel, lo que nos permite escribir código de manera sencilla con una sintaxis más amigable con principiantes.

Partiendo de lo anterior, *Manim* resulta ser una librería de *Python*, por lo que nos limitaremos a este lenguaje para programar. Esta librería es de código abierto y fue creada por Grant Sanderson, un matemático egresado de la Universidad de Stanford. Grant Sanderson publica videos en un canal de *YouTube* llamado *3B1B* y a raíz de la popularidad y apoyo que recibió de sus suscriptores, estos últimos lanzaron la versión de la librería que utilizaremos en esta tesis, llamada *Manim Community Edition*.

Existen tres versiones de *Manim* a la fecha de hoy:

- *Manim* ó *ManimCE*, la librería que mantiene la comunidad y la que usaremos en este trabajo.
- *ManimGL*, la cual es la librería que utiliza Grant Sanderson para sus videos y que suele traer elementos experimentales.
- *ManimCairo*, la versión previa a *ManimGL* y que resulta obsoleta, pues ha caído en desuso y principalmente sirve para correr códigos de los antiguos videos de 3B1B.

En general, no se requiere de un gran computador para correr *Manim*; aunque esto afecta el tiempo de generación de los archivos multimedia, ya sean imágenes o videos. También es posible utilizarlo en línea, en sitios donde pueda ejecutarse *Python*, por ejemplo en *Google Colab*. El código está hecho para la versión 18.0 de *Manim* y es esta versión la que usaremos para el tutorial.

1.1. Recomendaciones

Para mayor fluidez de trabajo y comodidad, se recomienda el siguiente software:

- *VSCode* es un entorno de desarrollo integrado, o por su abreviación en inglés, IDE (*Integrated Development Environment*), que nos permite personalizarlo a nuestro gusto. Se recomienda ampliamente, ya que cuenta con extensiones para *Python* y *Jupyter Notebook*; así como con atajos de teclado y fragmentos de código personalizables.
- *Jupyter Notebook* es un cuadernillo (similar a *R Markdown*) donde se puede combinar texto plano y código, el cual puede correrse dentro del mismo. Esto último facilita la interacción entre el código que se escribe y la salida del mismo.; así como agregar comentarios.

Son totalmente opcionales estos programas, aunque dentro del tutorial de instalación se hará específicamente para estos. Es decir, puede realizarse la instalación en otro IDE ó sin usar *Jupyter Notebook*.

1.2. Instalación para Windows

La versión de *Windows* utilizada es *Windows 10 Home*, aunque puede utilizarse para versiones posteriores. Para la instalación de *Manim* utilizaremos los siguientes.

Requisitos:

- *MiKTeX*,
- *Strawberry Perl*,
- *Python* 3.11.1 (incluyendo *pip*),
- *VSCode*,
- Extensión para *Python* en *VSCode*,
- Extensión para *Jupyter Notebook* en *VSCode*.

Supondremos que se ha instalado el software solicitado. Se recomienda buscar tutoriales de instalación en caso de no contar con los programas.

Instalación paso a paso:

1. En la terminal de *VSCode* escribimos (sin las comillas): `«pip install manim==0.18.0»`.

2. Esperamos a que se instalen los paquetes y las dependencias incluidas con *manim* 0.18.0.

A la fecha del 17 de diciembre de 2025, puede instalarse con mayor facilidad siguiendo las instrucciones en el siguiente enlace: <https://docs.manim.community/en/stable/installation.html>.

1.3. Instalación para *macOS*

Este tutorial funciona, por lo menos, para la versión de *macOS Sequoia 15.6.1*.

Requisitos:

- Homebrew
- *VSCode*
- *Python* 3.11.1

Instalación paso a paso:

1. Primero instalaremos *Homebrew* en caso de no tenerlo a través del siguiente enlace: <https://brew.sh/>. Para ello copiaremos la instrucción que viene ahí y la ejecutaremos en una terminal. Después, aparecen unas instrucciones para agregar *brew* al «PATH», por lo que hay que escribirlas y ejecutarlas una por una en la terminal.
2. Ahora escribimos en la terminal (sin las comillas): «*brew install py3cairo ffmpeg*» y esperamos a que se instale.
3. Hay un paso adicional y es que, dispositivos que cuenten con el chip M1 o similares, se tienen que instalar unas dependencias adicionales. Esto lo puede revisar viendo el menú de *Apple* y seleccionando *Acerca de esta Mac*. En caso de contar con ese chip, correr: «*brew install pango pkg-config scipy*».
4. Creamos una carpeta en donde guardaremos las animaciones. De ahí abrimos *VSCode* y con el teclado presionamos «*command + shift + p*». Le damos donde dice crear environment y seleccionamos *venv* y luego *python 3.11.1*

1.4. Estructura básica de Manim

A continuación explicaremos qué función cumple cada parte del código en un programa básico. El siguiente es un breve código para dibujar un cuadrado blanco de lado 2 centrado sobre un fondo negro, pero que nos servirá como punto de partida.

```
1 | from manim import*
2 |
3 | class HolaMundo(Scene):
4 |     def construct(self):
5 |         Cuadrado = Square()
6 |         self.add(Cuadrado)
```

Lo primero es importar nuestra librería *Manim* y esto lo hacemos con la siguiente línea.

```
1 | from manim import*
```

Es importante mencionar que utilizar «*» al final de *import* cargará **toda** la librería de *Manim* y que, si se trabaja con otras librerías, podría ser que colindasen algunas funciones y esto trajera problemas consigo; por lo que se recomienda que sólo se trabaje *Manim* en el archivo, aunque, como ya veremos, utilizando «*numpy import as np*» (por ejemplo) podemos evitarnos problemas.

```
3 | class HolaMundo(Scene):
```

Aquí *class* sería nuestro «lienzo». *HolaMundo* es simplemente el nombre que se desee y (*Scene*) define qué tipo de funciones podremos utilizar dentro. Hay diferentes tipos de «lienzos» (o escenas): la que empleamos es (*Scene*), que es trabajar en $\mathbb{R}^2$. Otros tipos de escenas serían: (*ThreeDScene*), que sería como trabajar en $\mathbb{R}^3$ y (*MovingCameraScene*), que permite mover la cámara tridimensionalmente.

Ahora, «debajo» de *class* sigue *def construct(self)*:. El término *def* se utiliza en *Python* para definir funciones; en este caso, *construct(self)*: contendrá todo lo que aparecerá en el *lienzo*. Propiamente de *Manim* —más que de *Python*— es utilizar *construct(self)*: siempre. Así fue diseñado.

```
4 |     def construct(self):
```

Dentro de *def construct(self)*: es donde pondremos los objetos o lo que queramos que se vea en la escena. En este caso, colocaremos un cuadrado centrado de color blanco sobre un fondo negro (viene por default). Más opciones, como color, tamaño, posición, etc. lo veremos en otro apartado de *kwargs*. Aquí *Square()* es una función nativa de *Manim*.

```
5 | Cuadrado = Square()
```

Por último, para que aparezcan los objetos tenemos dos opciones. En la primera es *self.add()*, que añade directamente los objetos a la escena; por otro lado, en la segunda que es *self.play()*, podemos decidir cómo es que se añadan los objetos a la escena. La gran diferencia entre estos es que *self.add()* nos genera una imagen *.png*, mientras que, si además utilizamos *self.play()*, nos produce un video en formato *.mp4*. En este caso primeramente utilizaremos *self.add()*.

```
6 | self.add(Cuadrado)
```

El resultado que obtendríamos sería la siguiente imagen *.png*.

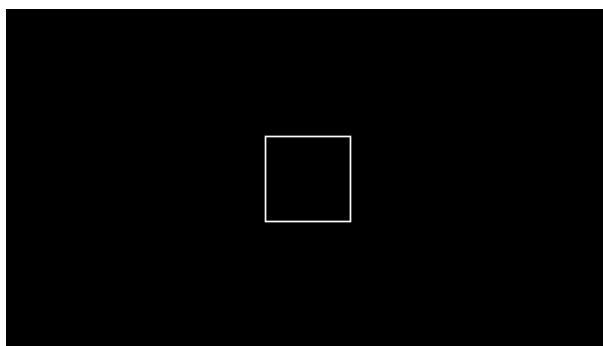


FIGURA 1.1: Imagen generada utilizando el código de arriba.

Dado que este es un *.pdf*, es imposible colocar videos en formato *.mp4*, por lo que sólo colocaremos código cuando se amerite.

1.5. Contenido adicional

Explicaremos algunos de los principales componentes de una escena en *Manim*. Estos componentes son varios y, por tanto, sólo haremos mención de los más afines al tema principal de esta tesis.

Scenes

Las escenas son nuestro «lienzo» y es donde colocaremos nuestros objetos, por lo que es importante conocer qué tipos existen y sus diferencias.

- **Scene**

Esta escena se utiliza para animaciones en dos dimensiones. La mayoría de funciones pueden utilizarse aquí, con ciertas excepciones como los objetos tridimensionales.

- **MovingCameraScene**

Muy similar a *Scene*, sólo que ésta permite desplazar la cámara o moverla. No permite emplear objetos tridimensionales.

- **ThreeDScene**

Con esta escena podemos utilizar funciones pensadas para tres dimensiones, ya sean superficies, como veremos más adelante. Aquí podemos rotar la cámara.

Mobjects

Los mobjects son los objetos con los que se trabajan en la escena, puede variar desde una figura predeterminada o alguna función o imagen. Son todo aquello que podemos manipular.

kwargs

Los *kwargs* se traducen como *keyword arguments* son comandos adicionales que uno le da a la función para modificar alguno de los parámetros que vienen por default. Normalmente se utilizan en los *Mobjects*, aunque también pueden utilizarse en otras funciones. A continuación se presentan algunos de ellos.

- *color* = aquí podemos utilizar los colores predeterminados de *Manim*, los cuales están en inglés y mayúsculas, además que cuentan con distintas versiones que se denotan por un guion bajo y una sola letra mayúscula que puede ir desde la A hasta la E. Por ejemplo: *color = 'RED'*
- *opacity* = nos permite modificar la opacidad de un objeto, es decir, que tan traslúcido es.
- *fill_opacity* = sirve para rellenar figuras, ya sean las que *Manim* trae por defecto o personalizadas.
- *stroke_width* = es el grosor del trazado de las líneas.

Animations

Cuando nos referimos a *Animations*, hablamos de las maneras en las que podemos agregar objetos a la escena mediante la función (que comunmente utilizaremos) *self.play()*. Existen más tipos de animaciones, no sólo las que se muestran a continuación.

- *animation*. Existen tres tipos para animar objetos: **Add**, **Animation** y **Wait**.
 - **Add**: Añade objetos a la escena sin más, sin animaciones. Podría decirse que de una manera inmediata y estática.
 - **Animation**: Añade los objetos a la escena mediante una animación; ésta puede ser personalizada o alguna de las predeterminadas.
 - **Wait**: Pone en pausa (estático) al objeto, lo deja estático aunque el tiempo siga avanzando.
- *changing*. Éstas nos permiten animar los rastros que dejan los objetos al desplazarse dentro de la escena. Hay dos tipos: **AnimatedBoundary** y **TracedPath**; éste último es el que nos permite dibujar la traza del objeto mediante una función.
- *creation*. Estas animaciones son las que nos permiten animar la creación de objetos dentro de nuestra escena. Hay distintos tipos; sin embargo, las que se utilizarán principalmente son las siguientes:
 - **Create**: Se construye poco a poco el objeto que queremos en la escena. Este proceso se realiza automáticamente, pues así ha sido programado.
 - **Write**: Es similar al anterior, sólo que en caso del texto «simula» como si una mano lo trazara.
- *transform*. Estas funciones ya implementadas nos permiten transformar un objeto en otro. Hay diversos métodos predefinidos; sin embargo, el que se utilizará principalmente es **Transform**, que directamente transforma un objeto en otro. Esta transformación no la «controlamos» con alguna función, es decir, lo realiza según ha sido programada.
- *updaters*. Estos nos permiten actualizar una función según un parámetro que demos. Sería un «equivalente» a un deslizador en *GeoGebra*. Como tal, una de las funciones es **update**, pero ésta se emplea de distinta manera a las anteriores. Se explicará con mayor detalle en la sección que se utilice.

Nota: algunas piezas de código que se repiten en varios códigos se dejan definidas a continuación; es decir, en los códigos que se presentan al final de cada sección se asumirá que se utilizan estos fragmentos y se omitirán como tal para efectos prácticos.

Se utilizará el fondo blanco en todas las escenas, por lo que se utiliza

```
1 | config.background_color = WHITE
```

LISTING 1: Código para cambiar el color de fondo de la escena.

Para los ejes en $\mathbb{R}^2$:

```
1 | L = 2 #longitud
2 | ax = Axes(
3 |     x_range=[-1 * L, L, 1], x_length=config.frame_height-1,
4 |     y_range=[-1 * L, L, 1], y_length=config.frame_height-1,
5 |     tips=True, #si queremos flechas al final de los ejes
6 |     ).set_color(BLACK) #color de los ejes
7 | axlbl = ax.get_axis_labels().set_color(BLACK) #si queremos que el eje muestre su etiqueta
8 | self.add(ax, axlbl)
```

LISTING 2: Código para agregar los ejes en el plano.

Para los ejes en $\mathbb{R}^3$:

```
1 | ax = ThreeDAxes(
2 |     x_range=[-1 * L, L, 1], x_length=config.frame_height-1,
3 |     y_range=[-1 * L, L, 1], y_length=config.frame_height-1,
4 |     z_range=[-1 * L, L, 1], z_length=config.frame_height-1,
5 |     tips=False, #si queremos flechas al final de los ejes
6 |     ).set_color(BLACK) #color de los ejes
7 | axlbl = ax.get_axis_labels().set_color(BLACK) #si queremos que el eje muestre su etiqueta
8 | self.add(ax, axlbl)
```

LISTING 3: Código para agregar los ejes en el espacio.

Para los anteriores ejemplos de los ejes, se abrevió como «ax» (*axes*, ejes), además de utilizar un parámetro L con el fin de facilitar la manipulación del eje. De igual forma, la longitud del eje se estableció respecto al tamaño del *frame*; esto nos permite visualizarlo en su totalidad.

Capítulo 2

Geometría de curvas en $\mathbb{R}^2$ y $\mathbb{R}^3$

La cotidianidad de las curvas a veces nos exime del cuestionarnos su naturaleza geométrica. Si observamos detenidamente, ¿podemos cuantificar qué tan curvada es una curva? La respuesta la daremos a continuación explorando las matemáticas detrás de las curvas; qué podemos decir de ellas y sus propiedades geométricas. Cabe mencionar que estudiarlas es clave para comprender a las superficies, pero esto queda para el siguiente capítulo.

2.1. Curvas parametrizadas diferenciales en $\mathbb{R}^n$

Empezaremos por definir lo que es una curva parametrizada. De ahora en adelante, se considerará el conjunto I como un intervalo abierto de $\mathbb{R}$, a menos que se indique lo contrario.

Definición 2.1.1. *Una curva parametrizada es una función, $\alpha : I \subset \mathbb{R} \rightarrow \mathbb{R}^n$,*

$$\alpha(t) = (x_1(t), x_2(t), \dots, x_n(t)),$$

con $x_i : I \subset \mathbb{R} \rightarrow \mathbb{R}$ las funciones componente; t es llamado el parámetro de la curva.

Una curva parametrizada, $\alpha : I \rightarrow \mathbb{R}^n$, se dice ser suave o diferenciable si sus funciones componente son C^∞ . Además, si $\alpha : I \rightarrow \mathbb{R}^n$ es una curva parametrizada, su imagen, $\alpha(I) \subset \mathbb{R}^n$, es llamada la traza de α . Si $\alpha : I \rightarrow \mathbb{R}^n$ es suave, su velocidad está dada por

$$\alpha'(t) = (x_1'(t), x_2'(t), \dots, x_n'(t)), \text{ con } x_i'(t) = \frac{dx_i}{dt}.$$

La rapidez de una curva parametrizada, $\alpha : I \rightarrow \mathbb{R}^n$, en $t \in I$, es

$$\|\alpha'(t)\| = \sqrt{(x'_1(t))^2 + (x'_2(t))^2 + \cdots + (x'_n(t))^2}.$$

La aceleración de una curva parametrizada, $\alpha : I \rightarrow \mathbb{R}^n$, es

$$\alpha''(t) = (x''_1(t), x''_2(t), \dots, x''_n(t)), \text{ donde } x''_i(t) = \frac{d^2 x_i}{dt^2}.$$

Una curva parametrizada suave, $\alpha : I \rightarrow \mathbb{R}^n$, se dice ser regular si para todo $t \in I$, $\alpha'(t) \neq 0$. En caso que $\alpha : I \rightarrow \mathbb{R}^n$ no sea regular, todos los $t_0 \in I$ donde $\alpha'(t_0) = 0$ son llamados puntos singulares de α .

Ejemplo 2.1.2. Sea $\alpha : \mathbb{R} \rightarrow \mathbb{R}^2$, $\alpha(t) = (\cos t, \sin t)$ es una curva parametrizada suave.

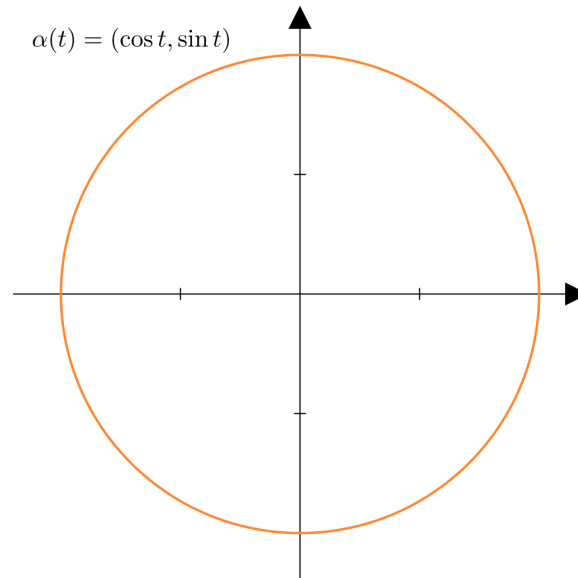


FIGURA 2.1: La traza de α .

En cada $t \in I$ tal que $\alpha'(t) \neq 0$, existe una recta en $\mathbb{R}^n$ que pasa por $\alpha(t)$ y paralelo a $\alpha'(t)$.

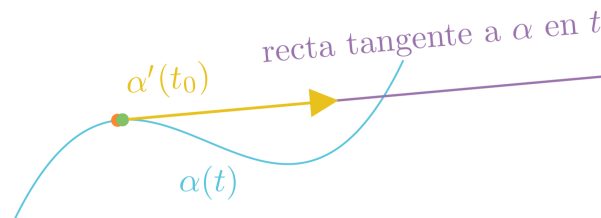


FIGURA 2.2: Ejemplo ilustrativo.

Definición 2.1.3. Una curva parametrizada regular, $\alpha : I \rightarrow \mathbb{R}^n$, se dice ser de rapidez unitaria si

$$\|\alpha'(t)\| = 1 \quad \forall t \in \mathbb{R}$$

Una curva parametrizada regular admite una recta tangente a α para cada $t \in \mathbb{R}$. La traza de $\alpha(t)$ no tiene esquinas ó cúspides.

2.2. Longitud de arco y parámetro natural de curvas

Definiremos la longitud de arco, la cual nos sirve para medir distancias entre dos puntos de una curva. Más aún, definiremos cómo reparametrizar una curva. Por último, se muestran ejemplos de curvas dinámicas, las cuales son generadas a partir de la traza o camino que recorre un punto fijo sobre una curva a lo largo de mover la curva, en este caso la curva es una circunferencia.

Definición 2.2.1. Sea $\alpha : (a, b) \subset \mathbb{R} \rightarrow \mathbb{R}^n$ una curva parametrizada regular y fijemos $t_0 \in (a, b)$. La función longitud de arco, $s : (a, b) \rightarrow \mathbb{R}$, se define por

$$s(t) = \int_{t_0}^t \|\alpha'(\tau)\| d\tau.$$

Por argumentos de cálculo diferencial e integral, se tiene que $s(t)$ nos mide la distancia orientada desde el punto $\alpha(t_0)$ hasta $\alpha(t)$ a lo largo de la curva. Es decir, si $t > t_0$, entonces $s(t)$

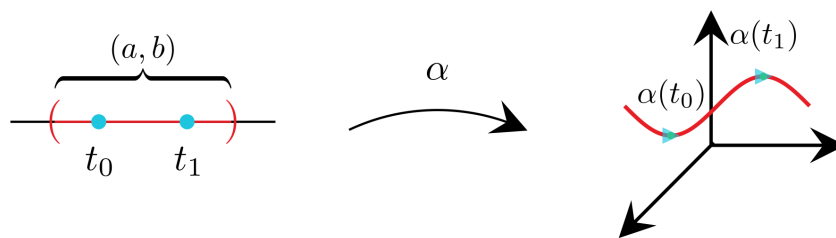


FIGURA 2.3: Dibujo ilustrativo.

- Si $t > t_0$, $s(t) > 0$.
- Si $t = t_0$, $s(t) = 0$.
- Si $t < t_0$, $s(t) < 0$.

Ejemplo 2.2.2. Consideremos $\alpha : \mathbb{R} \rightarrow \mathbb{R}^2$ como la elipse dada por $\alpha(t) = (a \cos t, b \sin t)$, con $a < b$; y calculemos su longitud de arco, la cual viene dada por

$$s(t) = \int_0^t \sqrt{a^2 + (b^2 - a^2) \cos^2 \tau} d\tau$$

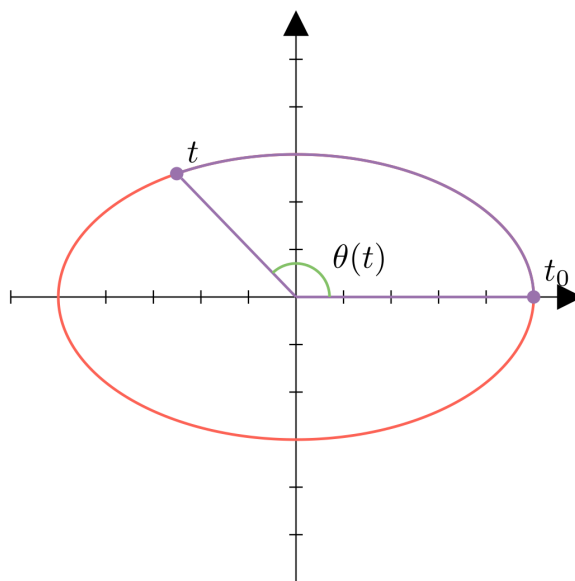


FIGURA 2.4: La traza de α , con $a > b$.

Para la implementación llevada a cabo en *Python* se utilizó este código

```

1      # cálculo de la curvatura
2      x = sy.Symbol('x')
3      c = 5
4      d = 3
5
6      fx = c*sy.cos(x)
7      fy = d*sy.sin(x)
8
9      fx1 = sy.diff(fx, x)
10     fy1 = sy.diff(fy, x)
11
12     fx2 = sy.diff(fx1, x)
13     fy2 = sy.diff(fy1, x)
14
15     num = sy.Abs(fx1*fy2 - fy1*fx2)
16     den = (fx1**2 + fy1**2)**sy.Rational(3,2)
17     k_sym = sy.simplify(num / den)
18
19     curvature = sy.lambdify(x, k_sym, "numpy")
20
21     # cálculo de longitud
22     def long(t):

```

```

23         c = 5
24         d = 3
25         if t <= 0:
26             return 0
27         n = 200
28         u = np.linspace(0, t, n)
29         speed = np.sqrt(c*c*np.sin(u)**2 + d*d*np.cos(u)**2)
30         return np.trapz(speed, u)

```

Ahora definiremos lo que es una reparametrización.

Definición 2.2.3. Sea $\alpha : (a, b) \rightarrow \mathbb{R}^n$ una curva parametrizada regular y $h : (c, d) \rightarrow (a, b)$ una función diferenciable. Si h es una función invertible con inversa diferenciable, la curva parametrizada regular $\beta : (c, d) \rightarrow \mathbb{R}^n$, dada por $\beta(\tau) = (\alpha \circ h)(\tau)$, es llamada una reparametrización de α .

La función $h : (c, d) \rightarrow (a, b)$ es llamada cambio de parametrización.

Observación 2.2.4. Si $h : (c, d) \rightarrow (a, b)$ es un cambio de parametrización:

- $\alpha(t)$ y $\beta(t)$ tienen la misma traza o trayectoria.
- $\frac{dh}{d\tau} > 0$ ó $\frac{dh}{d\tau} < 0$ para todo $\tau \in (c, d)$.
- Si $\frac{dh}{d\tau} > 0$, $\alpha(t)$ y $\beta(t)$ tienen la misma orientación.
- Si $\frac{dh}{d\tau} < 0$, $\alpha(t)$ y $\beta(t)$ tienen orientaciones opuestas.

Parámetro natural de una curva parametrizada

Si $\alpha : (a, b) \rightarrow \mathbb{R}^n$ es una curva parametrizada regular, la función longitud de arco, $s(t)$, es una función diferenciable por el Teorema Fundamental del cálculo. Calculando su derivada, obtenemos:

$$\frac{ds}{dt} = \|\alpha'(t)\| > 0 \quad \forall t \in (a, b),$$

puesto que $\|\alpha'(t)\| \neq 0$. Esto quiere decir que $s(t)$ es una función invertible con inversa diferenciable. Si $s(b) = l < \infty$, la inversa de la función longitud de arco, $h : (0, l) \rightarrow (a, b)$, $h(s)$, $h \circ s(t) = t$, es un cambio de parametrización de α . Por la regla de la cadena,

$$\frac{dh}{ds} = \frac{1}{\|\alpha'(t(s))\|}.$$

La parametrización por la inversa de la función longitud de arco es llamada el parámetro natural de la curva por la siguiente razón.

Teorema 2.2.5. Sea $\alpha : (a, b) \rightarrow \mathbb{R}^n$ una curva parametrizada regular y $h(s)$ la inversa de la función longitud de arco. La reparametrización $\tilde{\alpha}(s) = (\alpha \circ h)(s)$ es una curva de rapidez unitaria.

Demostración. Por definición, $\dot{\alpha}(s) = \frac{d\alpha}{ds}(s)$. Por la regla de la cadena, $\dot{\tilde{\alpha}}(s) = \frac{dh}{ds}(s)\alpha'(h(s))$. Luego,

$$\dot{\tilde{\alpha}}(s) \cdot \dot{\tilde{\alpha}}(s) = \left(\frac{dh}{ds}(s) \right)^2 (\alpha'(h(s)) \cdot \alpha'(h(s))) = \frac{1}{\|\alpha'(t)\|^2} (\|\alpha'(t)\|^2) = 1.$$

■

Ejemplo 2.2.6. El círculo de radio r centrado en $(0, 0)$, ¿cómo se puede parametrizar?

Sea $\alpha : (0, 2\pi) \rightarrow \mathbb{R}^2$, donde $\alpha(t) = (r \cos t, r \sin t)$. Derivando obtenemos $\alpha'(t) = (-r \sin t, r \cos t)$. Su longitud de arco entonces es $\|\alpha'(t)\| = r$. Consideremos $t_0 = 0$; luego,

$$s(t) = \int_0^t \|\alpha'(\tau)\| d\tau = \int_0^t r d\tau = rt$$

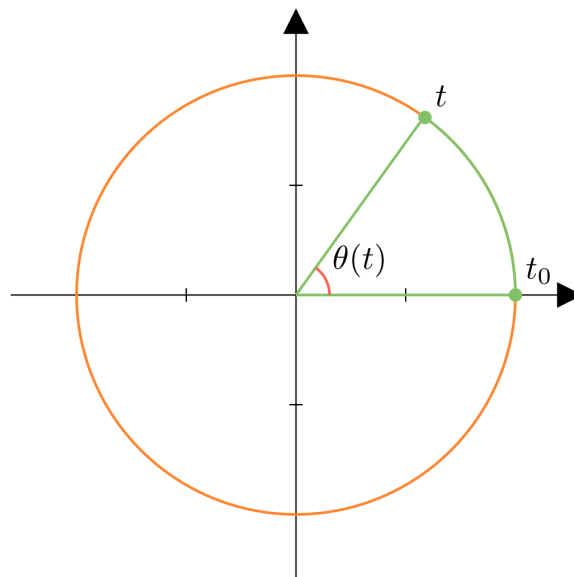


FIGURA 2.5: La traza de α y la longitud de arco entre t_0 y t .

2.2.1. Curvas dinámicas

Las curvas dinámicas son trayectorias descritas por objetos en movimiento, las cuales podemos parametrizar mediante una función.

Ejemplo 2.2.7 (Cicloide). *Supongamos que un círculo de radio a se encuentra en el eje x , haciendo contacto en el origen. Dejemos que el círculo ruede sin deslizamiento por todo el eje x positivo. La cicloide es la curva que describe la trayectoria que sigue el punto que originalmente hacía contacto con el eje x , y cuando la circunferencia rueda, está dada por*

$$\alpha(t) = (a(t - \sin t), a(1 - \cos t)),$$

donde t es el ángulo formado por el nuevo punto de contacto con el eje x , el centro y el punto original de contacto.

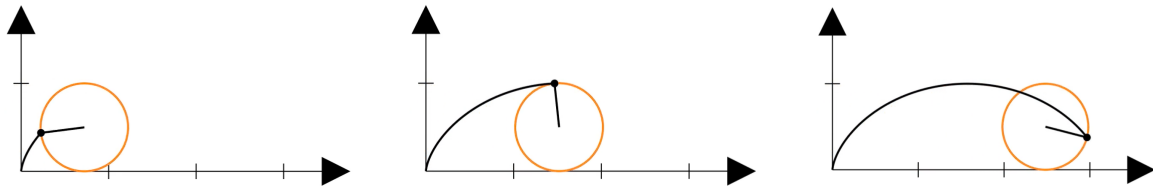


FIGURA 2.6: Cicloide

Ejemplo 2.2.8 (Hipocicloide). *Un círculo de radio $r < 1$ rueda dentro de un círculo de radio 1, centrado en el origen O . El centro C del círculo pequeño se encuentra inicialmente en $(1 - r, 0)$. Sea θ el ángulo subtendido por la línea OC y el eje x positivo. La posición del punto P , inicialmente en $(1, 0)$, traza una curva a la vez que el círculo pequeño rueda dentro del círculo unitario. La curva traza por el punto P tiene las siguientes ecuaciones paramétricas:*

$$x(\theta) = (1 - r) \cos \theta + r \cos \left(\frac{1 - r}{r} \theta \right),$$

$$y(\theta) = (1 - r) \sin \theta - r \sin \left(\frac{1 - r}{r} \theta \right).$$

A esta curva se le conoce como hipocicloide. En este caso, la hipocicloide formará un deltoide, que es una curva de tres cúspides con forma triangular.

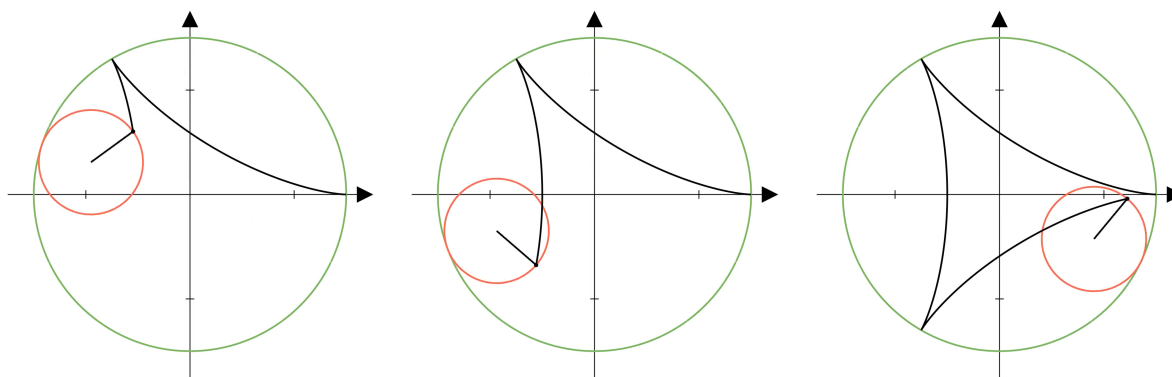


FIGURA 2.7: Deltoide

Ejemplo 2.2.9 (Astroide). La definición del astroide es similar a la de la cicloide. Sin embargo, un círculo rueda, no sobre una línea, sino dentro de otro círculo. Precisando aún más, un círculo de radio $a/4$ rueda dentro de un círculo mayor de radio a y centro en el origen. Supongamos que el pequeño círculo empieza en $(a, 0)$ y sigue el camino del punto originalmente en contacto con $(a, 0)$ a la par que el círculo va rodando. Si rodamos el círculo un poco, notaremos que una pieza de la longitud de arco s se ha usado.

Para t , el ángulo desde el centro del círculo mayor al nuevo punto de contacto y θ , el ángulo al cual el pequeño círculo ha rodado (medido desde el punto de contacto), tenemos que $s = at$ y $s = a\theta/4$. Por lo tanto, $\theta = 4t$.

Ahora podemos hacer dos cosas para parametrizar el punto que va rodando. Primero, podemos parametrizar el camino con respecto al centro del círculo pequeño. Esto es sencillo, pues estamos moviendo un ángulo θ al círculo. Notemos que aquí se puede utilizar la relación $\theta = 4t$. Las coordenadas con respecto al centro se vuelven entonces

$$x = (a/4) \cos 3t \quad y = (a/4) \sin 3t$$

En segundo lugar, el centro del pequeño círculo permanece de radio $3a/4$ centro en el origen, como el centro se ha movido $((3a/4) \cos t, (3a/4) \sin t)$. Luego, con respecto al origen, el punto rodante se ha movido

$$\alpha(t) = \left(\frac{3a}{4} \cos t + \frac{a}{4} \cos 3t, \frac{3a}{4} \sin t - \frac{a}{4} \sin 3t \right),$$

lo cual puede reducirse a

$$\alpha(t) = (a \cos^3 t, a \sin^3 t),$$

con forma implícita $x^{2/3} + y^{2/3} = a^{2/3}$.

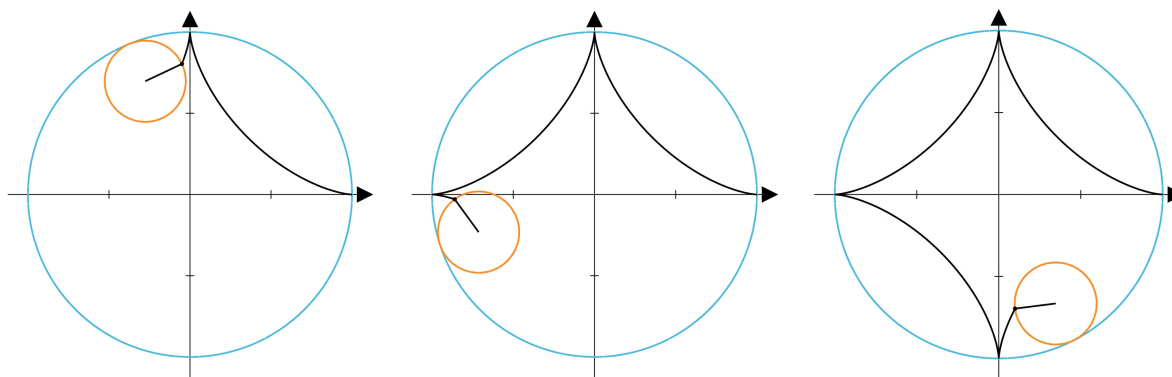


FIGURA 2.8: Astroide

Esta parte del código es fundamental para dibujar la traza que deja el punto al moverse en cada uno de los ejemplos.

```

1      A = always_redraw(
2          lambda: ParametricFunction(
3              lambda u: axes.c2p(*self.func2(u, s.get_value())),
4              t_range=[0, 2*PI], color = ORANGE
5          ))
6
7      ball = Dot(radius=0.05, color = BLACK)
8      ball.add_updater(lambda x: x.move_to(A))
9
10     path = VObject()
11     path.set_points_as_corners([ball.get_center(), ball.get_center()])
12
13     def update_path(path):
14         previous_path = path.copy()
15         previous_path.add_points_as_corners([ball.get_center()])
16         path.become(previous_path)
17         path.add_updater(update_path)

```

2.3. Teoría local de curvas en $\mathbb{R}^2$

La teoría local de curvas en el plano estudia las propiedades geométricas de una curva parametrizada regular, que son independientes de su posición en el plano e invariantes bajo transformaciones rígidas (traslaciones y transformaciones ortogonales). El concepto clave es la curvatura, una función que en cada punto de la curva mide la rapidez con la que cambia la dirección de la misma. Otra interpretación del concepto de curvatura es como el recíproco multiplicativo del radio de la circunferencia tangente en un punto dado.

Para estudiar el comportamiento local de una curva se utiliza el marco de Frenet-Serret, el cual a cada punto de una curva parametrizada regular le asigna una base ortonormal con orientación positiva: el vector tangente a la curva (T) y el vector normal (N). La evolución del marco de Frenet-Serret, con respecto al parámetro natural de una curva regular, está gobernado precisamente por la curvatura, lo que confirma su relevancia como objeto que determina la información geométrica de una curva plana.

Además, la curvatura define una relación de equivalencia en el espacio de curvas regulares planas: dos curvas planas con la misma función de curvatura difieren salvo por un movimiento rígido, es decir, una traslación seguida de una transformación ortogonal. Esto le otorga al concepto de curvatura la categoría de propiedad geométrica. Más aún, el Teorema Fundamental de Curvas establece que cualquier función continua, no negativa, determina una curva plana regular cuya curvatura es dicha función.

Curvatura y circunferencia osculadora

Consideremos una curva $r(t) = (f(t), g(t)) \in \mathbb{R}^2$. En el punto P sobre la curva, la circunferencia osculadora en P es una circunferencia que toca la curva en P , donde el círculo y la curva tienen la misma línea tangente. Puede ser vista como la mejor circunferencia que aproxima la curva en el punto P . Puede demostrarse que su centro y su radio están dados por:

$$R = \frac{1}{\pm\kappa(t)}, \quad C = (x_c(t), y_c(t)),$$

donde

$$x_c(t) = f(t) - \frac{g'(t)(f'(t)^2 + g'(t)^2)}{f'(t)g''(t) - g'(t)f''(t)},$$

$$y_c(t) = g(t) + \frac{f'(t)(f'(t)^2 + g'(t)^2)}{f'(t)g''(t) - g'(t)f''(t)},$$

y κ su curvatura.

A continuación se presentan algunas circunferencias osculadoras que giran alrededor de curvas dadas.

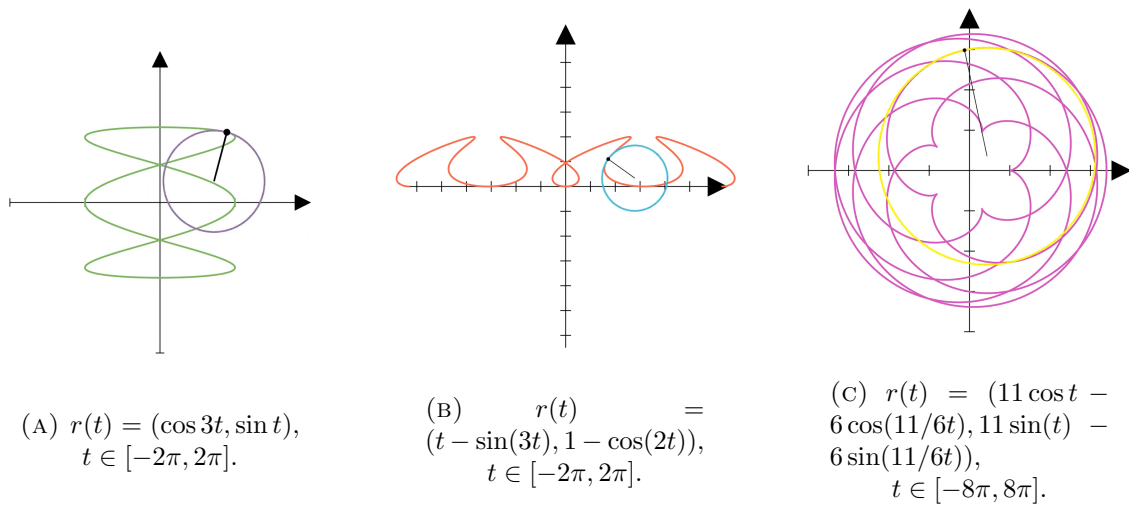


FIGURA 2.9: Curvas con sus respectivas circunferencias osculadoras.

Para las circunferencias osculadoras anteriores se utilizó cálculo simbólico con la librería de *Sympy* y luego se pasó a cálculo numérico con *Numpy*.

```

1  self.x_sym = sy.symbols('x')
2  fan_sym = (self.x_sym, sy.sin(self.x_sym), 0) #función
3
4  # Cálculo de derivadas simbólicas
5  d1_sym = [sy.diff(fan_sym[i], self.x_sym) for i in range(0,3)] #derivada primera
6  d2_sym = [sy.diff(fan_sym[i], self.x_sym, 2) for i in range(0,3)] #derivada segunda
7
8  self.func_x = sy.lambdify(self.x_sym, fan_sym[0], 'numpy')
9  self.func_y = sy.lambdify(self.x_sym, fan_sym[1], 'numpy')
10
11 self.d1_0 = sy.lambdify(self.x_sym, d1_sym[0], 'numpy')
12 self.d1_1 = sy.lambdify(self.x_sym, d1_sym[1], 'numpy')
13
14 self.d2_0 = sy.lambdify(self.x_sym, d2_sym[0], 'numpy')
15 self.d2_1 = sy.lambdify(self.x_sym, d2_sym[1], 'numpy')

```

Y con este otro código se hicieron los cálculos para la curvatura, radio y centro de la circunferencia osculadora.

```

1  d1_0 = self.d1_0(t)
2  d1_1 = self.d1_1(t)
3  d2_0 = self.d2_0(t)
4  d2_1 = self.d2_1(t)
5
6  # Cálculo de la curvatura (k) y radio (R)
7  num = (d1_0 * d2_1 - d1_1 * d2_0)
8  den = (d1_0**2 + d1_1**2)**(3/2)
9
10 # Curvatura
11 k = num / den if den != 0 else 0

```

```

12
13     if abs(k) < 1e-9:
14         k = np.inf
15
16     # Radio de curvatura (R = 1/k)
17     R = 1/k if k != 0 else np.inf
18
19     # Valores de la función en el punto t
20     x_at_t = self.func_x(t)
21     y_at_t = self.func_y(t)
22
23     if k == np.inf or R == np.inf:
24         xc, yc = x_at_t, y_at_t
25         return np.array([x_at_t, y_at_t, 0])
26     else:
27
28         r_val_sq = (d1_0**2 + d1_1**2)
29         xc = x_at_t - d1_1 * r_val_sq / num
30         yc = y_at_t + d1_0 * r_val_sq / num
31
32         return np.array([
33             R*np.cos(u) + xc,
34             R*np.sin(u) + yc,
35             0
36         ])

```

Consideremos $\alpha : I \rightarrow \mathbb{R}^2$, una curva parametrizada regular, parametrizada por longitud de arco. Fijemos $s_0 \in I$ y definamos $p_0 = \alpha(s_0)$. Supongamos que $\ddot{\alpha}(s) \neq 0$ para todo $s \in I$, donde $\ddot{\alpha}$ es la segunda derivada de la curva α . Tomemos $s_1, s_2 \in I$ tales que $s_1 < s_0 < s_2$ y $p_1 = \alpha(s_1)$, $p_2 = \alpha(s_2)$. Los puntos p_0 , p_1 y p_2 no son colineales y definen una circunferencia. Sea $C(s_0, s_1, s_2)$ el centro de la circunferencia que pasa por $\alpha(s_1)$, $\alpha(s_0)$ y $\alpha(s_2)$.

Usando argumentos del cálculo diferencial, se puede probar que $C(s_0, s_1, s_2)$ converge a un punto $C(s_0)$ si $s_1, s_2 \rightarrow s_0$. Si consideramos la circunferencia de centro $C(s_0)$ y radio $R(s_0) = \frac{1}{\|\ddot{\alpha}(s_0)\|}$, se tiene una circunferencia tangente a α en el punto $\alpha(s_0)$. Tal circunferencia es llamada circunferencia osculadora (círculo osculatriz).

Definición 2.3.1. *El punto $C(s)$ es llamado el centro de curvatura. La curvatura de $\alpha(s)$ se define como el recíproco del radio de la circunferencia osculadora,*

$$\kappa_{\pm}(\alpha, s) := \frac{1}{\|\alpha(s) - C_{\alpha(s)}\|}$$

Nota: La notación $\kappa_{\pm}(\alpha, s)$ para la curvatura es tomada de [6].

Proposición 2.3.2. *Para una curva plana, parametrizada por longitud de arco,*

$$\kappa_{\pm}(\alpha, s) := \|\alpha''(s)\|.$$

Demostración. Sea $\alpha : I \rightarrow \mathbb{R}^2$. $\forall s \in I$, $\alpha'(s)(\alpha(s) - C(s)) = 0$. Por otra parte, $\alpha'(s) \cdot \alpha''(s) = 0$. ■

Proposición 2.3.3. *La curvatura de una curva regular, $\alpha(t) = (x_1(t), x_2(t))$, está dada por*

$$\kappa_{\pm}(\alpha, t) = \left| \frac{x_1'(t)x_2''(t) - x_2'(t)x_1''(t)}{((x_1'(t))^2 + (x_2'(t))^2)^{3/2}} \right|$$

La prueba de la proposición anterior se puede consultar en [6].

Ejemplo 2.3.4. *Calcular la curvatura $\kappa_{\pm}$ de $\alpha : \mathbb{R} \rightarrow \mathbb{R}^2$, $\alpha(t) = (a \cos t, b \sin t)$, con $a > b > 0$ constantes.*

$$\begin{array}{lll} x_1(t) = a \cos t, & x_1'(t) = -a \sin t, & x_1''(t) = -a \cos t, \\ x_2(t) = b \sin t & x_2'(t) = b \cos t & x_2''(t) = -b \sin t \end{array}$$

$$\kappa_{\pm}(\alpha, t) = \left| \frac{x_1'(t)x_2''(t) - x_2'(t)x_1''(t)}{((x_1'(t))^2 + (x_2'(t))^2)^{3/2}} \right| = \left| \frac{ab}{(a^2 \sin^2 t + b^2 \cos^2 t)^{3/2}} \right|.$$

La siguiente imagen muestra una elipse, que es la traza de la curva α , junto con su función curvatura y longitud de arco. Como podemos apreciar de la gráfica, la curvatura es una función periódica con puntos críticos en los vértices de la elipse.

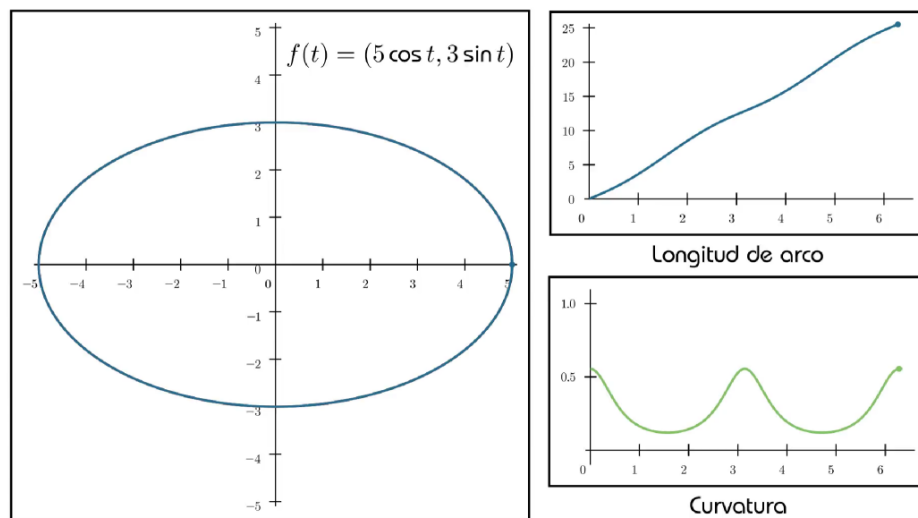


FIGURA 2.10: Traza de α con $a = 5$ y $b = 3$, su función curvatura y longitud de arco.

Curvatura orientada

Para definir el concepto de curvatura orientada vamos a presentar un breve recordatorio del concepto de orientación en $\mathbb{R}^2$. Se dice que una base $\{u = (u_1, u_2), v = (v_1, v_2)\}$ para $\mathbb{R}^2$ tiene orientación estándar o positiva si la matriz

$$\begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix}$$

tiene determinante positivo. Dado un vector $\vec{u} \in \mathbb{R}^2$, $\vec{u} \neq 0$, el vector $\vec{v}$ definido por $\vec{v} = \mathbb{J}\vec{u}$, con $\mathbb{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ (matriz de rotación por un ángulo $\frac{\pi}{2}$) determina una base $\{\vec{u}, \vec{v}\}$ con orientación positiva.

Sea $\alpha(t)$ una curva parametrizada regular. Se define el **vector tangente unitario** a α en $\alpha(t)$ por

$$T(t) := \frac{\alpha'(t)}{\|\alpha'(t)\|}$$

Si $\alpha(s)$ está parametrizada por longitud de arco, $T(s) = \dot{\alpha}(s)$, donde $\dot{\alpha}$ es la primera derivada de α . Definimos el vector normal unitario

$$N(t) = \mathbb{J}T(t)$$

Observemos que, para cada t , $\{T(t), N(t)\}$ son una base ortonormal de $\mathbb{R}^2$ con orientación positiva.

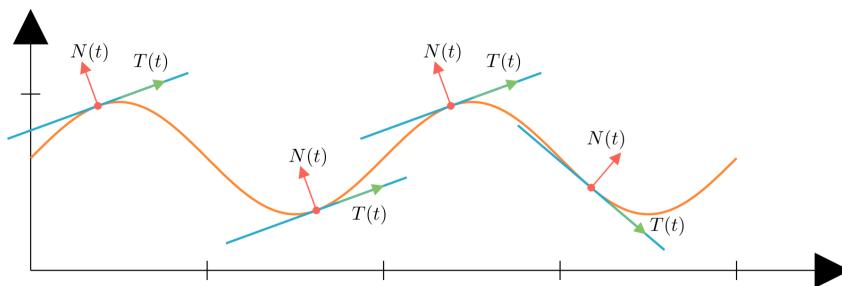


FIGURA 2.11: Dibujo ilustrativo del marco de Frenet-Serret.

Definición 2.3.5. La curvatura (orientada), $\kappa_\alpha(t)$, de $\alpha(t)$ se define por la identidad

$$T'(t) = (\kappa(t) \cdot \|\alpha'(t)\|) N(t).$$

Como $\|T(t)\| = 1 \ \forall t$, podemos expresar

$$T(t) = (\cos \theta(t), \sin \theta(t)),$$

donde $\theta(t)$ es llamado el «ángulo de giro». Entonces $N(t) = (-\sin \theta(t), \cos \theta(t))$. Calculando la derivada de $T(t)$,

$$T'(t) = \theta'(t) (-\sin \theta(t), \cos \theta(t)) = \theta'(t) N(t).$$

de donde se sigue que $\theta'(t) = \kappa_\alpha(t) \|\alpha'(t)\|$. Si el ángulo de giro es conocido y no cero,

$$\kappa_\alpha^\pm(t) = \frac{\theta'(t)}{\|\alpha'(t)\|}.$$

En caso de que α esté parametrizada por longitud de arco, $\alpha(s)$.

$$\begin{aligned} T(s) &= \dot{\alpha}(s) \\ \kappa_\alpha(s) &= \|\ddot{\alpha}(s)\| = \left\| \dot{T}(s) \right\| = \|\kappa_\alpha^\pm \|\dot{\alpha}(s)\| N(s)\| \\ &= |\kappa_\alpha^\pm(s)|. \end{aligned}$$

La curvatura κ_α y la curvatura orientada difieren a lo más en el signo para curvas de rapidez unitaria.

Teorema 2.3.6. Si $\alpha = (x(t), y(t))$ es una curva regular, entonces

$$\kappa_\alpha(t) = \frac{(x'y'' - x''y')}{((x')^2 + (y')^2)^{3/2}} = \frac{\langle \mathbb{J}\alpha'(t), \alpha''(t) \rangle}{\|\alpha'(t)\|^3}.$$

Demostración. Sea $T'(t) = \kappa_\alpha \|\alpha'(t)\| N(t)$, $N = \mathbb{J} \frac{\alpha'}{\|\alpha'(t)\|}$. Luego, $T(t) = \frac{\alpha'(t)}{\|\alpha'(t)\|}$, $T'(t) = \frac{\alpha''(t)}{\|\alpha'(t)\|} + \frac{\alpha'(t)\|\alpha'(t)\|}{\alpha' \cdot \alpha''}$. Además,

$$\frac{d}{dt} (\|\alpha'(t)\|) = \frac{\alpha' \cdot \alpha''}{\|\alpha'(t)\|}.$$

Así,

$$\begin{aligned} \kappa_\alpha \|\alpha'(t)\| &= N(t) \cdot T'(t) = \frac{\langle \mathbb{J}\alpha'(t), \alpha''(t) \rangle}{\|\alpha'(t)\|^2} \\ \kappa_\alpha(t) &= \frac{\langle \alpha''(t), \mathbb{J}\alpha'(t) \rangle}{\|\alpha'(t)\|^2} \end{aligned}$$

Si $\alpha(t) = (x(t), y(t))$,

$$\kappa_\alpha(t) = \frac{x'y'' - x''y'}{((x')^2 + (y')^2)^{3/2}}$$

■

Evolvente de una curva

Sea $\alpha : I \subset \mathbb{R} \rightarrow \mathbb{R}^2$ una curva parametrizada regular parametrizada por longitud de arco tal que $\alpha''(s) \neq 0$ para $s \in I$. Esto implica que existe la circunferencia osculadora cuyo centro define una curva para $\mathcal{C}(s)$ que se puede expresar en términos de $\alpha(s)$ y de la curvatura orientada, $\kappa(s)$.

$$\mathcal{C}(s) = \alpha(s) + \frac{1}{\kappa(s)^2} \ddot{\alpha}(s).$$

La curva $\mathcal{C}(s)$ definida por los centros de curvatura es llamada la curva evolvente de $\alpha(s)$. Para ilustrar este concepto, se realizó un programa que grafica la evolvente de una curva en un punto dado.

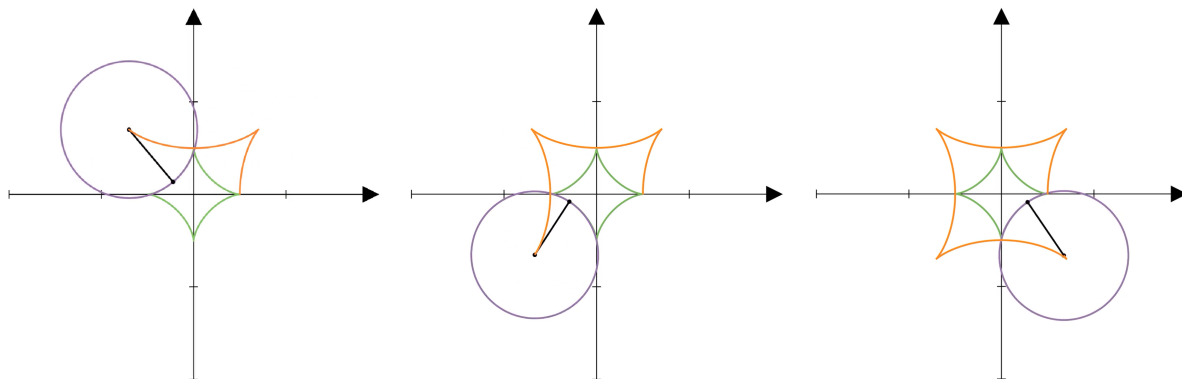


FIGURA 2.12: La evolvente de un astroide resulta ser otro astroide.

Con este código se realizaron los cálculos para la evolvente

```

1      x_sym = sy.symbols('x')
2
3      #función
4      fan_sym = sy.Matrix([
5          (sy.cos(x_sym))**3,
6          (sy.sin(x_sym))**3,
7          0])
8
9      d1_sym = sy.diff(fan_sym, x_sym, 1) # Derivada primera simbólica

```

```

10     d2_sym = sy.diff(fan_sym, x_sym, 2) # Derivada segunda simbólica
11
12     fan_func = sy.lambdify(x_sym, fan_sym, "numpy")
13     d1_func = sy.lambdify(x_sym, d1_sym, "numpy")
14     d2_func = sy.lambdify(x_sym, d2_sym, "numpy")
15
16     ft = fan_func(t).flatten()
17     d1 = d1_func(t).flatten()
18     d2 = d2_func(t).flatten()
19
20     num = np.float64(d1[0] * d2[1] - d1[1] * d2[0])
21     mod_d1_sq = d1[0]**2 + d1[1]**2
22     den = np.power(mod_d1_sq, 3/2)
23
24     epsilon = 1e-6
25     k = np.float64(num / (den if den > epsilon else epsilon))
26
27     r = np.float64(1.0 / k)
28
29     factor = mod_d1_sq / num
30
31     xc = np.float64(ft[0] - d1[1] * factor)
32     yc = np.float64(ft[1] + d1[0] * factor)
33
34     R_circulo = np.float64(abs(r))
35
36     # Ecuación de la circunferencia osculadora
37     circle_x = R_circulo * np.cos(u) + xc
38     circle_y = R_circulo * np.sin(u) + yc
39
40     return np.array([circle_x, circle_y, xc, yc])

```

Interpretación del signo de la curvatura orientada

Para dar una interpretación geométrica, necesitamos considerar el siguiente resultado.

Proposición 2.3.7. *Si $\kappa_{\pm}(s)$ y $\kappa(s)$ son respectivamente la curvatura con signo y la curvatura orientada de una curva $\alpha : I \rightarrow \mathbb{R}^2$ regular, parametrizada por longitud de arco, entonces*

$$\kappa(s) = \kappa_{\pm}(s) \cos(\theta(s)),$$

donde $\theta(s)$ es el ángulo entre $\ddot{\alpha}(s)$ y $N(s)$.

Demostración. Como $\dot{T}(s) = \kappa(s)N(s)$,

$$\kappa(s) = \langle \mathbb{J}\ddot{\alpha}(s), \ddot{\alpha}(s) \rangle = \langle \mathbb{J}T(s), \ddot{\alpha}(s) \rangle = \langle N(s), \ddot{\alpha}(s) \rangle = \|\ddot{\alpha}\|(s) \cos(\theta(s)) = \kappa_{\pm}(s) \cos(\theta(s)).$$

■

De la proposición anterior, se tiene que la curvatura orientada, $\kappa(s)$, tiene el mismo signo que $\cos(\theta(s))$. Como $N(s)$ y $\ddot{\alpha}(s)$ son perpendiculares a $T(s)$, entonces $\theta(s) = 0$ ó π . Además, dado que el vector $\ddot{\alpha}(s)$ siempre apunta en la dirección del centro de curvatura, tenemos que:

- i) $\kappa(s) > 0$ si $N(s)$ está del mismo lado que la circunferencia osculadora respecto a la recta tangente a α en $\alpha(s)$,
- ii) $\kappa(s) < 0$ si $N(s)$ y la circunferencia osculadora están en lados contrarios.

Para ejemplificar los puntos (i) e (ii), veamos qué pasa con la función seno.

Ejemplo 2.3.8. Consideremos la función seno y observemos cómo se comporta su circunferencia osculadora en diferentes puntos. Notemos que la circunferencia osculadora cambia de lado tras pasar por un punto donde su segunda derivada es equivalentemente 0.

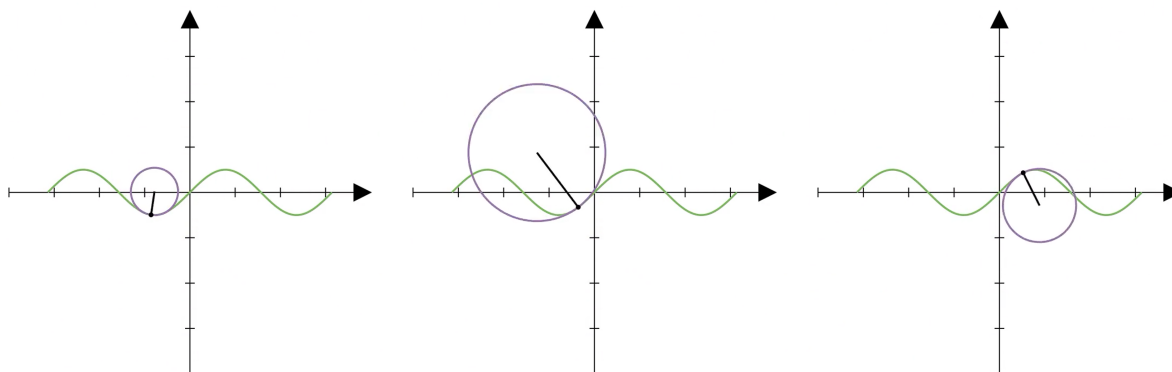


FIGURA 2.13: Circunferencia osculadora del seno en diversos puntos.

Tomando en cuenta el hecho anterior, podemos construir la evolvente para una curva $\alpha(t)$ de parametrización arbitraria:

$$\mathcal{C}(t) = \alpha(t) + \frac{1}{\kappa(t)}N(t).$$

Base móvil para una curva parametrizada

Sea $\alpha : I \rightarrow \mathbb{R}^2$ una curva parametrizada regular. Para $t \in I$, podemos asociarle al punto $\alpha(t)$ una base ortonormal de $\mathbb{R}^2$ definida por $\{T(t), N(t)\}$, donde

$$T(t) = \frac{\alpha'(t)}{\|\alpha'(t)\|} \quad \text{y} \quad N(t) = \mathbb{J}T(t), \quad \text{con } \mathbb{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Los vectores $T(t)$, $N(t)$ son diferenciales con respecto a t , de hecho, ya se tiene que

$$T'(t) = (\kappa(t) \|\alpha'(t)\|) N(t) \quad (2.1)$$

De aquí se sigue que:

$$\begin{aligned} N'(t) &= \mathbb{J}T'(t) = \kappa(t) \|\alpha'(t)\| \mathbb{J}N \\ &= (\kappa(t) \|\alpha'(t)\|) \mathbb{J}^2 T(t) \\ &= -(\kappa(t) \|\alpha'(t)\|) T(t) \end{aligned} \quad (2.2)$$

Si consideramos la función matricial que definen los vectores $T(t)$, $N(t)$, se tiene que ésta satisface la ecuación diferencial matricial

$$[T'(t), N'(t)] = [T(t), N(t)] \begin{bmatrix} 0 & -\kappa(t) \|\alpha'(t)\| \\ \kappa(t) \|\alpha'(t)\| & 0 \end{bmatrix}$$

Para una curva $\alpha(s)$ parametrizada por longitud de arco, la ecuación anterior se reduce a

$$\begin{bmatrix} \dot{T} \\ \dot{N} \end{bmatrix} = [T, N] \begin{bmatrix} 0 & -\kappa(s) \\ \kappa(s) & 0 \end{bmatrix} \quad (*)$$

Esta ecuación demuestra que la función curvatura caracteriza y define a una curva $\alpha(s)$ en el plano parametrizada por longitud de arco, salvo movimientos rígidos.

2.3.1. Teorema Fundamental de Curvas en $\mathbb{R}^2$

Teorema 2.3.9 (Teorema Fundamental de Curvas en $\mathbb{R}^2$). *Dada una función suave, $\kappa : (a, b) \rightarrow \mathbb{R}$, existe una curva, $\alpha : (a, b) \rightarrow \mathbb{R}^2$, parametrizada por longitud de arco cuya curvatura es precisamente κ . Más aún, si $\bar{\alpha} : (a, b) \rightarrow \mathbb{R}^2$ está parametrizada por longitud de arco con curvatura también κ , entonces α y $\bar{\alpha}$ difieren de una rotación seguida de una traslación.*

La prueba del teorema se basa en que podemos expresar el vector tangente

$$T(s) = (\cos \theta(s), \sin \theta(s)),$$

donde $\dot{\theta}(s) = \kappa(s)$. Por lo que

$$\theta(s) = \int \kappa(s) ds + \theta_0.$$

Entonces,

$$T(s) = (\cos(\theta(s) + \theta_0) + \sin(\theta(s) + \theta_0)).$$

Como $T(s) = (\dot{x}(s), \dot{y}(s))$, con $\alpha(s) = (x(s), y(s))$, se tiene que:

$$\begin{aligned} x(s) &= \int \cos(\theta(s) + \theta_0) + a_1 \\ y(s) &= \int \sin(\theta(s) + \theta_0) + a_2, \end{aligned}$$

con $a_1, a_2 \in \mathbb{R}$ arbitrarios, θ_0 un ángulo arbitrario y $\theta(s) = \int \kappa(s) ds$.

Para ilustrar el Teorema Fundamental de Curvas en el plano, se elaboró un programa que permite construir una curva a partir de una función dada que represente su curvatura. En las siguientes imágenes se muestra el resultado del programa.

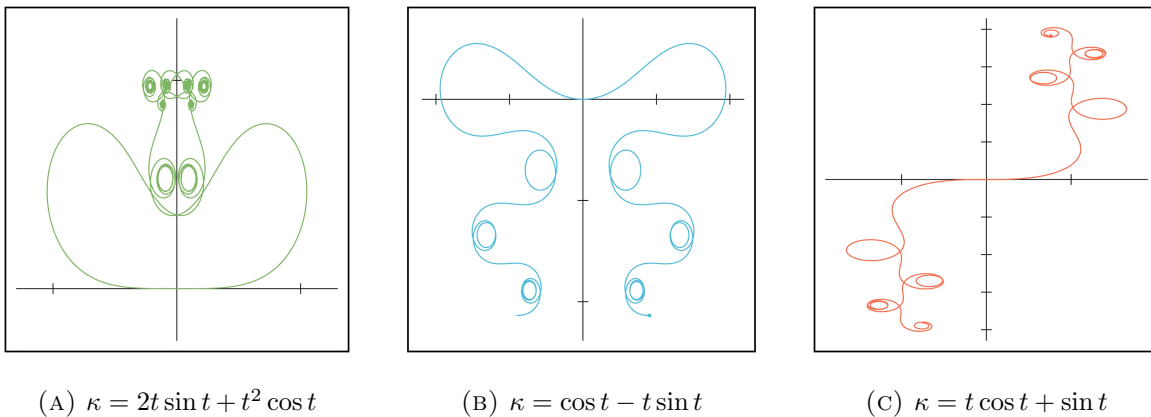


FIGURA 2.14: Distintas curvas generadas dada alguna curvatura κ .

Con este código se calculan los puntos de la curva y combinándose con las funciones de *Manim*, se obtienen las curvas anteriores.

```

1 import scipy.integrate as sc
2 from scipy.interpolate import interp1d
3
4 class tfcp(Scene):
5     def construct(self):
6         t_min = -4*np.pi
7         t_max = 4*np.pi
8         N = 10000 # más puntos -> curva suave
9         T = np.linspace(t_min, t_max, N)

```

```

10
11     #función curvatura
12     def k(x):
13         return x*(1-x)
14
15     P = np.array([sc.quad(k, 0, t)[0] for t in T])
16
17     dX = np.cos(P)
18     dY = np.sin(P)
19
20     X_raw = sc.cumulative_trapezoid(dX, T, initial=0)
21     Y_raw = sc.cumulative_trapezoid(dY, T, initial=0)
22
23     idx0 = np.argmin(np.abs(T - 0))
24
25     X = X_raw - X_raw[idx0]
26     Y = Y_raw - Y_raw[idx0]
27
28     fx = interp1d(T, X, kind="cubic")
29     fy = interp1d(T, Y, kind="cubic")

```

2.4. Teoría local de curvas en $\mathbb{R}^3$

De forma análoga al caso de curvas planas, el estudio local de curvas en el espacio determina las propiedades locales de una curva parametrizada que son invariantes bajo transformaciones rígidas. En el caso de curvas en el espacio, los objetos de estudio son la curvatura y torsión, las cuales miden la forma y el movimiento de una curva en un punto dado. La curvatura es una medida de la desviación de una curva en un punto respecto a su recta tangente, mientras que la torsión determina cuánto una curva se aleja de ser plana, midiendo la «desviación» de una curva respecto al plano que determinan su tangente y normal.

En el estudio del comportamiento local de curvas en espacio se utiliza el triedro móvil de Frenet-Serret; éste, en cada punto de una curva, coloca una base ortonormal en $\mathbb{R}^3$ positivamente orientada, determinada por el vector tangente (T), el vector normal (N) y el vector binormal (B). Al analizar la evolución de triedro móvil, respecto al parámetro natural, se obtienen las ecuaciones de Frenet-Serret, que es un sistema lineal no autónomo cuya matriz del sistema está determinada tanto por la curvatura y la torsión.

De manera recíproca, el Teorema Fundamental de Curvas en el espacio establece que, si fijamos dos funciones continuas, siendo una de ellas positiva, siempre existe una curva parametrizada regular cuya curvatura y torsión son precisamente esas funciones.

2.4.1. Curvatura y torsión

Definición 2.4.1. Sea $\alpha : I \rightarrow \mathbb{R}^3$ una curva regular en $\mathbb{R}^3$ parametrizada por longitud de arco. La curvatura de α es la función $\kappa : I \rightarrow \mathbb{R}$ definida por

$$\kappa(s) := \|\ddot{\alpha}(s)\|.$$

Como $\alpha(s)$ es de rapidez unitaria, el vector $T(s) := \dot{\alpha}(s)$ es el vector tangente unitario a α . Como $\dot{T}(s) \cdot T(s) = 0$ para toda $s \in I$, se define el vector normal unitario

$$N(s) = \frac{1}{\kappa(s)} \dot{T}(s) = \frac{1}{\kappa(s)} \ddot{\alpha}(s), \quad (2.2)$$

siempre que $\kappa(s) \neq 0$. El plano generado por los vectores $\{T(s), N(s)\}$ en el punto $\alpha(s)$ es llamado el plano osculador, el cual contiene a la circunferencia osculadora.

Circunferencia osculadora para curvas en el espacio

Sea $\alpha : I \rightarrow \mathbb{R}^3$ una curva regular en $\mathbb{R}^3$ parametrizada por longitud de arco. Por simplicidad, supongamos que $\|\ddot{\alpha}(s)\| \neq 0$ para todo $s \in I$. De manera similar al caso de la curvas planas, esto garantiza que, en cada punto $\alpha(s)$ con $s \in S$, existe una circunferencia tangente a la curva α cuyo radio es $R = \frac{1}{\kappa(s)}$.

Dicha circunferencia también es llamada *circunferencia osculadora* y está contenida en el plano que contiene al punto $\alpha(s)$; generado por el vector tangente $T(s)$ y el vector normal $N(s)$. Este plano es llamado el *plano osculador*. Una parametrización de la circunferencia osculadora está determinada por

$$\gamma : [0, 2\pi] \rightarrow \mathbb{R}^3$$

$$\gamma(\theta) = \mathcal{C} + \cos(\theta) T(\theta) + \sin(\theta) N(\theta),$$

donde $\mathcal{C}(s)$ es el centro de circunferencia osculadora que se define por:

$$\mathcal{C}(s) = \alpha(s) + \frac{1}{\kappa(s)^2} \ddot{\alpha}(s).$$

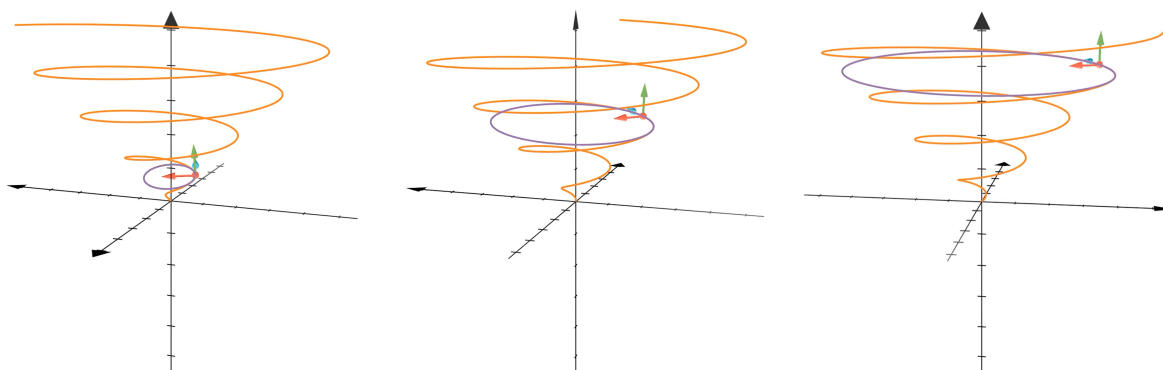


FIGURA 2.15: Circunferencia osculadora sobre una hélice.

El vector binormal a la curva α en $\alpha(s)$ se define por

$$B(s) = T(s) \times N(s),$$

donde $B(s)$ es ortogonal al plano osculador. El conjunto de vectores $\{T(s), N(s), B(s)\}$ es una base de $\mathbb{R}^3$ con orientación positiva, llamada triedro o base móvil de Frenet-Serret.

El vector binormal es diferenciable respecto a s , pues es producto de dos vectores diferenciales respecto a s . Como $B(s) \cdot B(s) = 0$, se sigue que $\dot{B}(s) \cdot B(s) = 0$. Por otro lado, $\dot{B}(s) = T(s) \times \dot{N}(s)$ (ya que $\dot{T}(s) = \kappa(s)N(s)$), lo que a su vez implica que $\dot{B}(s) \cdot T(s) = 0$. De aquí se concluye que $\dot{B}(s)$ es paralelo a $N(s)$.

Con esta función calculamos la circunferencia osculadora, así como el vector normal (U), tangente (v) y binormal (w).

```

1  def func1(self, u, t):
2      x = sy.symbols('x')
3      fan = (0.2 * x * sy.cos(x), 0.2 * x * sy.sin(x), 0.2 * x) #función
4      d1_num = np.array([sy.diff(fan[i], x).evalf(subs={x:t}) for i in range(3)], dtype=float)
5      d2_num = np.array([sy.diff(fan[i], x, 2).evalf(subs={x:t}) for i in range(3)], dtype=float)
6
7      r = np.array([fan[i].evalf(subs={x: t}) for i in range(3)], dtype=float)
8
9      norm_d1 = np.linalg.norm(d1_num)
10     U = d1_num / norm_d1
11     cross_d1_d2 = np.cross(d1_num, d2_num)
12     norm_cross = np.linalg.norm(cross_d1_d2)
13     if norm_cross == 0:
14         v = np.array([0., 0., 0.])
15         k = 0.0
16     else:
17         v = cross_d1_d2 / norm_cross
18         k = norm_cross / (norm_d1 ** 3)

```

```

19
20     w = np.cross(v, U)
21
22     if k == 0:
23         R = np.inf
24         C = np.array([np.inf, np.inf, np.inf])
25     else:
26         R = np.float64(1.0 / k)
27         C = r + R * w
28
29     xc = np.float64(C[0] + R * (np.cos(u) * U[0] + np.sin(u) * w[0]))
30     yc = np.float64(C[1] + R * (np.cos(u) * U[1] + np.sin(u) * w[1]))
31     zc = np.float64(C[2] + R * (np.cos(u) * U[2] + np.sin(u) * w[2]))
32
33     return [r, U, v, w, xc, yc, zc]

```

Definición 2.4.2. La torsión de α en s se define por el número $\tau(s)$ tal que

$$\dot{B}(s) = -\tau(s)N(s).$$

Se sigue de la definición $(B(s) \cdot N(s)) = 0$,

$$\tau(s) = -\langle B'(s), N(s) \rangle = \langle B(s), N'(s) \rangle$$

La torsión de α está relacionada con la variación del plano osculador. Si $\tau \equiv 0$, $B(s)$ es un vector constante y α está contenida en un plano.

Una curva $\alpha : I \rightarrow \mathbb{R}^3$ se dice ser plana si $\tau \equiv 0$. Por último, se puede probar que

$$\dot{N}(s) = -\kappa(s)T(s) + \tau(s)B(s).$$

Teorema 2.4.3 (Fórmulas de Frenet). Para una curva de velocidad unitaria, con $\kappa > 0$, las derivadas del triedro de Frenet son

$$\dot{T} = \kappa N$$

$$\dot{N} = -\kappa T + \tau B$$

$$\dot{B} = -\tau N$$

A continuación se muestra al triedro de Frenet sobre distintas curvas.

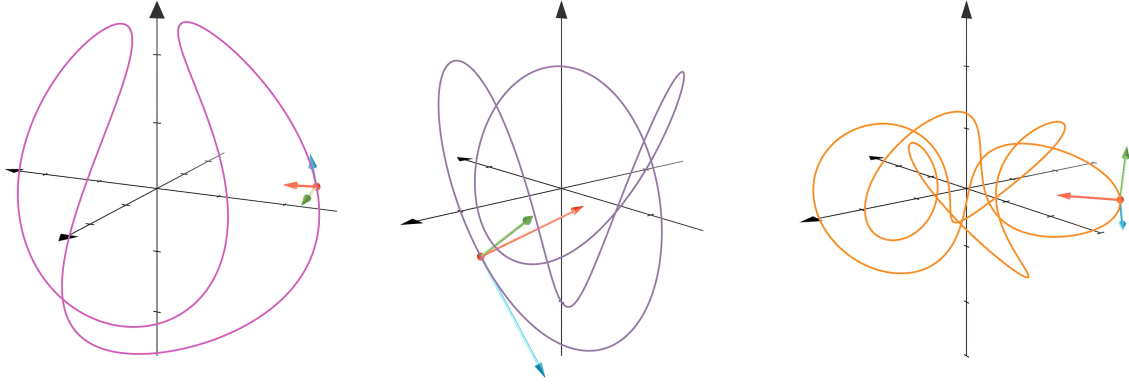


FIGURA 2.16: Triedro móvil sobre distintas curvas, descritas en el repositorio.

2.4.2. Teorema Fundamental de Curvas en $\mathbb{R}^3$

Dadas $f, g : I \rightarrow \mathbb{R}$, funciones $\mathcal{C}^\infty$, con $f > 0$, existe $\alpha : I \rightarrow \mathbb{R}^3$, una curva parametrizada por longitud de arco tal que $\kappa_\alpha(s) = f(s)$, $\tau(s) = g(s)$.

Demostración. Consideremos el sistema lineal

$$\dot{A}(s) = A(s) \begin{bmatrix} 0 & -f(s) & 0 \\ f(s) & 0 & -g(s) \\ 0 & g(s) & 0 \end{bmatrix}, \quad (2.3)$$

con $A(s) = \begin{bmatrix} T(s) & N(s) & B(s) \end{bmatrix}$. ■

Lema 2.4.4. Si $F : I \rightarrow \mathcal{M}_{n \times n}(\mathbb{R})$ es una función continua con valores en el espacio de matrices, donde $I = (a, b)$, entonces existe una solución al sistema de ecuaciones diferenciales (2.3), $A : (a, b) \rightarrow \mathbb{R}^n$, $\dot{A} = AF$, $A(s_0) = C$, $s_0 \in I$, $C \in \mathcal{M}_{n \times n}(\mathbb{R})$.

Lema 2.4.5. Si $F(s)$ es una función continua en (a, b) , con valores en el espacio de matrices asimétricas, $C \in SO(3)$ y $s_0 \in I$, entonces la solución, $A(s)$, al sistema lineal $\dot{A}(s) = A(s)F(s)$, $F(s_0) = C$, es una función diferenciable con valores en el grupo ortogonal $SO(3)$.

Demostración. Sea $A(s)$ la matriz solución de

$$\frac{dA}{ds} = A(s)F(s), \quad F(s)^T = -F(s).$$

Por demostrar, $A(s) \cdot A(s)^T = I$.

$$\frac{d}{ds}(A \cdot A^T) = \left(\frac{dA}{ds}\right) A^T + A \left(\frac{dA}{ds}\right)^T = (A \cdot F) A^T + A (AF)^T = AFA^T = \text{cte.}$$

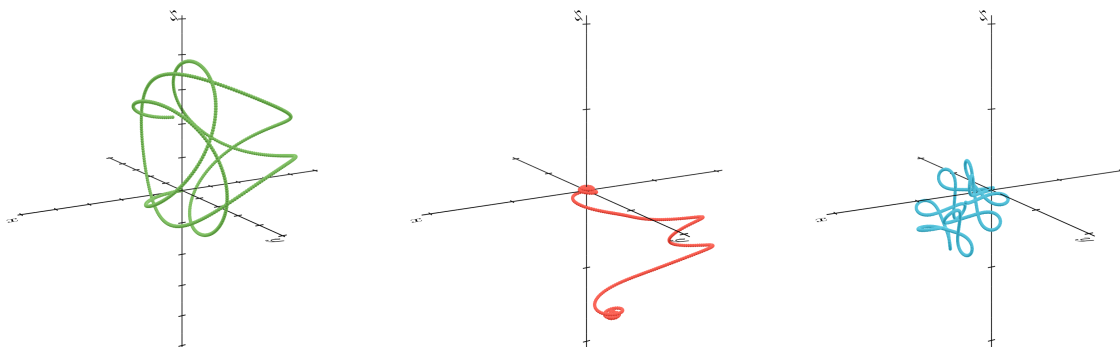
Luego, $A(s)A^T(s) = A(s_0)A^T(s_0) = C \cdot C^T = I$. ■

Por los lemas anteriores, existe $A(s) = \begin{bmatrix} T(s) & N(s) & B(s) \end{bmatrix}$ que satisface (2.3). Definamos

$$\alpha(s) = \int_{s_0}^s T(r) dr, \quad T(s) = A(s) \cdot e_1, \text{ donde } e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Supongamos que $\alpha : I \rightarrow \mathbb{R}^3$ y $\bar{\alpha} : I \rightarrow \mathbb{R}^3$ son curvas tales que $\kappa_\alpha = \kappa_{\bar{\alpha}}$, $\tau_\alpha = \tau_{\bar{\alpha}}$, $s_0 \in I$ fijo.

Ejemplo 2.4.6. Con ayuda de Python, ilustraremos el Teorema Fundamental de curvas en $\mathbb{R}^3$.



(A) $\kappa = 1/2$ y $\tau = 3/8 \sin(s/2)$.

(B) $\kappa = s^2 - 4$ y $\tau = \cos(s)$.

(C) $\kappa = 10 \sin^2(\pi s)$ y $\tau = 3 \cos(\pi s)$.

FIGURA 2.17: Resultado del programa con distintas curvaturas y torsiones.

Con este código se resuelve el sistema matricial de ecuaciones diferenciales. Se proporciona tanto la curvatura como la torsión. Se guarda en un archivo `.csv` en el que después el programa utiliza para colocar los puntos en el espacio y construir la gráfica de la curva.

```

1  from scipy.integrate import solve_ivp
2  import pandas as pd
3
4  def frenet_serret(s, y):
5      T = y[0:3]
6      N = y[3:6]
```

```

7   B = y[6:9]
8   r = y[9:12]
9
10  k = 10 * (np.sin(np.pi * s))**2 #curvatura
11  t = 3 * np.cos(np.pi * s) #torsión
12
13  #sistema de ecuaciones
14  dTds = k * N
15  dNds = -k * T + t * B
16  dBds = -t * N
17  drds = T
18
19  return np.concatenate([dTds, dNds, dBds, drds])
20
21  # Condiciones iniciales
22  T0 = np.array([1, 0, 0])
23  N0 = np.array([0, 1, 0])
24  B0 = np.array([0, 0, 1])
25  r0 = np.array([0, 0, 0])
26  y0 = np.concatenate([T0, N0, B0, r0])
27
28  # Intervalo de integración
29  s_span = (-2 * np.pi, 2 * np.pi)
30  s_eval = np.linspace(*s_span, 700)
31
32  # Resolver el sistema
33  sol = solve_ivp(frenet_serret, s_span, y0, t_eval=s_eval, method='RK45')
34
35  # Extraer la curva
36  r = sol.y[9:12].T # Transpuesta para tener las filas como puntos
37
38  # Guardar en CSV
39  df = pd.DataFrame(r, columns=["x", "y", "z"])
40  df.to_csv("curva_frenet_serret.csv", index=False)
41
42  print("Archivo 'curva_frenet_serret.csv' guardado.")

```

Y con estas líneas lo agregamos en *Manim*.

```

1  df = pd.read_csv("curva_frenet_serret.csv")
2  dots = [Dot3D(ax.c2p(x, y, z), color= BLUE, radius=0.035) for x, y, z in df.values]
3  for dot in dots:
4      #self.add(dot)
5      self.play(Create(dot))

```

2.4.3. Aparato de Frenet para curvas de parámetro arbitrario

Sea $\alpha : I \rightarrow \mathbb{R}^3$ una curva parametrizada regular, no necesariamente parametrizada por longitud de arco. Ante la dificultad técnica y práctica para obtener, en la mayoría de

los casos, la reparametrización de α por longitud de arco, es necesario derivar fórmulas para el cálculo del triedro móvil de Frenet, la curvatura y torsión en términos de $\alpha(t)$.

Para calcular el vector tangente unitario,

$$T(t) := \frac{\alpha'(t)}{\|\alpha'(t)\|}.$$

El vector $\alpha''(t)$ no es ortogonal a $\alpha'(t)$, como ocurre cuando t es el parámetro natural, pero sí está contenido en el plano osculador. Esto permite definir el vector binormal de α en $\alpha(t)$,

$$B(t) = \frac{\alpha' \times \alpha''}{\|\alpha' \times \alpha''\|}(t),$$

siempre que $\alpha''(t) \neq 0$. Por lo que podemos obtener el vector normal por

$$N(t) = B(t) \times T(t).$$

2.4.4. Fórmulas para curvatura y torsión de parámetro arbitrario

Para la curvatura:

$$\kappa(t) = \frac{\|\alpha'(t) \times \alpha''(t)\|}{\|\alpha'(t)\|^3} \quad (2.4)$$

Para la torsión:

$$\tau(t) = \frac{\alpha'(t) \cdot (\alpha''(t) \times \alpha'''(t))}{\|\alpha'(t) \times \alpha''(t)\|^2}. \quad (2.5)$$

La derivación detallada de las fórmulas anteriores, (2.4) y (2.5), puede consultarse en las siguientes referencias: [3], [6], [7].

Capítulo 3

Superficies regulares en $\mathbb{R}^3$

Vivimos en un mundo tridimensional, así que es indiscutible que se tenga la noción común de superficie. Además, como podría uno imaginarse, algunas de los conceptos son análogos o muy similares a los que hemos definido previamente en el capítulo 2, que va de curvas.

3.1. Definición de superficies regulares y ejemplos

En esta sección definiremos a las superficies regulares, que en este caso son las superficies de revolución y regladas. También presentamos a las curvas coordenadas y valores regulares de funciones diferenciables; las anteriores surgen como necesidad de estudiar a las superficies regulares.

Definición 3.1.1. *Un subconjunto, $S \subset \mathbb{R}^3$, es una superficie regular si, para cada $p \in S$, existe una vecindad, $V \subset \mathbb{R}^3$, y una función, $\mathbf{x} : U \rightarrow V \cap S$, de un conjunto abierto, $U \subset \mathbb{R}^2$, sobre $V \cap S \subset \mathbb{R}^3$ tal que*

1. *$\mathbf{x}$ es diferenciable (suave).*

$$\mathbf{x}(u, v) = (x_1(u, v), x_2(u, v), x_3(u, v)), \quad (u, v) \in U,$$

las funciones $x_1(u, v), x_2(u, v), x_3(u, v)$ tienen derivadas parciales continuas de cualquier orden.

2. *$\mathbf{x}$ es un homeomorfismo.*

Esto es, $\mathbf{x}$ es continua, invertible y $\mathbf{x}^{-1} : V \cap S \rightarrow U$ es continua. Lo anterior se debe a que existe una función continua, $F : W \subset \mathbb{R}^3 \rightarrow \mathbb{R}^2$, definida en un abierto, $V \cap S \subset W$, tal que $F|_{V \cap S} = \mathbf{x}^{-1}$.

3. (Condición de regularidad) Para cada $(u, v) \in U$, el diferencial $D\mathbf{x}_{(u,v)} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ es uno a uno, donde

$$D\mathbf{x}_{(u,v)} = \begin{pmatrix} \frac{dx_1}{du}(u, v) & \frac{dx_1}{dv}(u, v) \\ \frac{dx_2}{du}(u, v) & \frac{dx_2}{dv}(u, v) \\ \frac{dx_3}{du}(u, v) & \frac{dx_3}{dv}(u, v) \end{pmatrix}$$

Observación 3.1.2.

- (i) La condición 2 ($\mathbf{x}$ es homeomorfismo) tiene el propósito de evitar autointersecciones. También permite definir conceptos geométricos usando una parametrización sin que dependan de la parametrización.
- (ii) La condición 3 implica la existencia de un «plano tangente» en cada punto de S .

Una función $\mathbf{x}$ que satisface las condiciones 1, 2 y 3 es llamada una parametrización ó un sistema (local) de coordenadas en (una vecindad de) p . La vecindad $V \cap S$ de p en S es llamada una *vecindad coordenada*.

Curvas coordenadas

Denotemos por $\{e_1, e_2\}$ a la base canónica en $\mathbb{R}^2$. Fijemos $q = (u_0, v_0) \in U$, las curvas: $\beta_1 : u \mapsto (u, v_0)$ y $\beta_2 : v \mapsto (u_0, v)$ en U . Supongamos que $p = \mathbf{x}(q)$ y consideremos las imágenes bajo $\mathbf{x}$ de β_1 y β_2 :

$$\gamma_1 : u \mapsto \mathbf{x}(u, v_0) \in \mathbb{R}^3, \text{ curva coordenada } v = v_0$$

$$\gamma_2 : v \mapsto \mathbf{x}(u_0, v) \in \mathbb{R}^3, \text{ curva coordenada } u = u_0$$

Notemos que γ_1, γ_2 son curvas regulares contenidas en S ,

$$\mathbf{x}(u, v) = (x_1(u, v), x_2(u, v), x_3(u, v)).$$

Derivando γ_1 en $u = u_0$, $\gamma_1(u) = \mathbf{x} \circ \beta_1(u)$,

$$\frac{d\gamma_1}{du}(u_0) = \left(\frac{\partial x_1}{\partial u}(u_0, v_0), \frac{\partial x_2}{\partial u}(u_0, v_0), \frac{\partial x_3}{\partial u}(u_0, v_0) \right) = \frac{\partial \mathbf{x}}{\partial u}(u_0, v_0)$$

Por otra parte,

$$\frac{d\gamma_1}{du}(u_0) = D_q \mathbf{x}(e_1).$$

Y de manera análoga,

$$D_q \mathbf{x} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$D_q \mathbf{x}(e_2) = \frac{\partial \mathbf{x}}{\partial v}(u_0, v_0)$$

Ejemplo 3.1.3. *Un ejemplo de superficie regular es la esfera unitaria,*

$$\mathbb{S}^2 = \{(x, y, z) \in \mathbb{R}^3 | x^2 + y^2 + z^2 = 1\}.$$

Consideremos $U = \{(x, y) \in \mathbb{R}^2 | x^2 + y^2 < 1\}$. La función $\mathbf{x} : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$, $\mathbf{x}(x, y) = (x, y, \sqrt{1 - x^2 - y^2})$, es una parametrización de $\mathbb{S}^2$ que cubre los puntos $p = (x, y, z) \in \mathbb{S}^2$, con $z > 0$.

$W = \{(x, y, z) \in \mathbb{R}^3 | z > 0\}$, $\mathbf{x} : U \rightarrow W \cap \mathbb{S}^2$, sobre.

1. $\mathbf{x}$ es diferenciable.

$$\mathbf{x}(x, y) = (x, y, \sqrt{1 - x^2 - y^2})$$

2. $\mathbf{x}$ es inyectiva. $\mathbf{x}(x_1, y_1) = \mathbf{x}(x_2, y_2)$ implica

$$(x_1, y_1, \sqrt{1 - x_1^2 - y_1^2}) = (x_2, y_2, \sqrt{1 - x_2^2 - y_2^2})$$

implica que $x_1 = x_2$ y $y_1 = y_2$.

$\mathbf{x}$ es sobre, $W \cap \mathbb{S}^2$. Para $p = (x, y, z) \in W \cap \mathbb{S}^2$, $z > 0$ y $x^2 + y^2 + z^2 = 1$, que implica $z = \sqrt{1 - x^2 - y^2}$, $\mathbf{x}(x, y) = p$, con $(x, y) \in U$.

3. Condición de regularidad.

$$D\mathbf{x}_{(x,y)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{x}{\sqrt{1-x^2-y^2}} & -\frac{y}{\sqrt{1-x^2-y^2}} \end{pmatrix}$$

Ahora cubriremos a la esfera con parametrizaciones similares. Definamos $\mathbf{x}_2 : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ como

$$\mathbf{x}_2(x, y) = \left(x, y, -\sqrt{1 - (x^2 + y^2)} \right);$$

cerciorándonos que $\mathbf{x}_2$ es una parametrización, y observando que $\mathbf{x}_1(U) \cup \mathbf{x}_2(U)$ cubre S^2 menos el ecuador

$$\{(x, y, z) \in \mathbb{R}^3; x^2 + y^2 = 1, z = 0\}.$$

De manera análoga, utilizando los planos xz y zy , definimos estas parametrizaciones

$$\mathbf{x}_3(x, z) = \left(x, +\sqrt{1 - (x^2 + z^2)}, z \right),$$

$$\mathbf{x}_4(x, z) = \left(x, -\sqrt{1 - (x^2 + z^2)}, z \right),$$

$$\mathbf{x}_5(y, z) = \left(+\sqrt{1 - (y^2 + z^2)}, y, z \right),$$

$$\mathbf{x}_6(y, z) = \left(-\sqrt{1 - (y^2 + z^2)}, y, z \right),$$

que, junto con $\mathbf{x}_1$ y $\mathbf{x}_2$, cubren S^2 completamente, por lo que es una superficie regular.

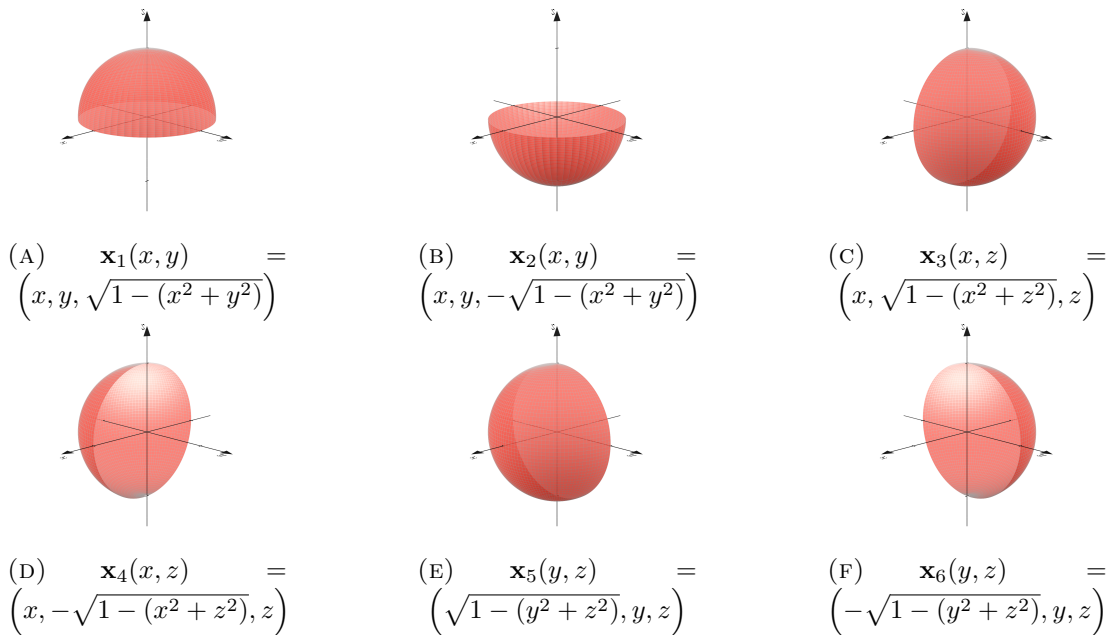


FIGURA 3.1: Parametrizaciones de la esfera.

Definición 3.1.4 (Superficies de Monge). Dada una función diferenciable $f : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, su gráfica $\text{graph}(f) = \{(x, y, f(x, y)) | (x, y) \in U\}$ es una superficie regular parametrizada por:

$$\mathbf{x} : U \rightarrow \mathbb{R}^3, \quad \mathbf{x}(u, v) = (u, v, f(u, v)).$$

A dichas superficies regulares parametrizadas de esta forma se les conoce como superficies de Monge.

Ejemplo 3.1.5. Consideremos $\mathbf{x}(u, v) = (u, v, u^2 + v^2)$, con $-\frac{1}{2} \leq u, v \leq \frac{1}{2}$. A continuación su gráfica.

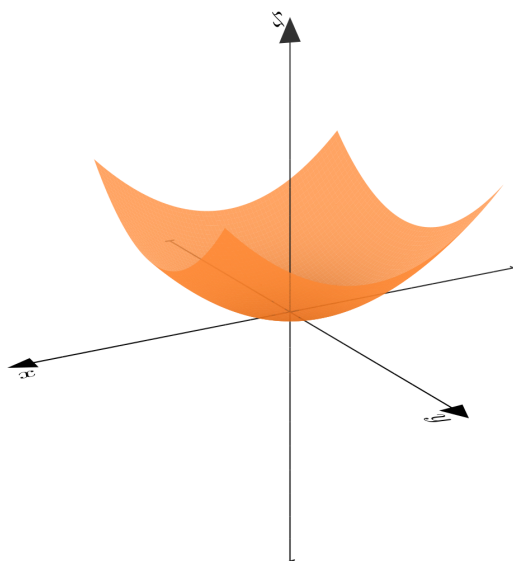


FIGURA 3.2: Superficie de Monge.

Valores regulares de funciones diferenciables

Dada una función, $f : V \subset \mathbb{R}^3 \rightarrow \mathbb{R}$, diferenciable, el conjunto solución de la ecuación

$$f(x, y, z) = 0 \quad (3.1)$$

determina una superficie en $\mathbb{R}^3$. La pregunta natural es: ¿bajo qué condiciones podemos garantizar que la superficie determinada por (3.1) es una superficie regular?

Definición 3.1.6. Sea $f : V \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ una función diferenciable. Un número, $a \in \mathbb{R}$, se dice ser un valor regular de f si para todo $(x, y, z) \in f^{-1}(a)$ se tiene que $\nabla f(x, y, z) \neq 0$. En caso contrario, a se dice valor crítico o singular de f .

Proposición 3.1.7. Si $a \in \mathbb{R}$ es un valor regular de $f : V \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ tal que $f^{-1}(a) \neq \emptyset$, entonces $S = f^{-1}(a)$ es una superficie regular.

Ejemplo 3.1.8. El toro es una superficie generada al rotar un círculo S^1 de radio r alrededor de una línea recta perteneciente al plano del círculo y a una distancia $a > r$ alejada del centro del círculo. Veámoslo.

Sea S^1 el círculo en el plano yz con centro el punto $(0, a, 0)$. Luego, S^1 está dado por $(y - a)^2 + z^2 = r^2$. Los puntos del toro se obtienen al rotar este círculo alrededor del eje z , satisfaciendo la ecuación

$$\left(\sqrt{x^2 + y^2} - a\right)^2 + z^2 = r^2.$$

Ahora, consideremos $f(x, y, z) = \left(\sqrt{x^2 + y^2} - a\right)^2 + z^2$. Por lo que

$$\frac{\partial f}{\partial x} = \frac{2x \left(\sqrt{x^2 + y^2} - a\right)}{\sqrt{x^2 + y^2}}, \quad \frac{\partial f}{\partial y} = \frac{2y \left(\sqrt{x^2 + y^2} - a\right)}{\sqrt{x^2 + y^2}}, \quad \frac{\partial f}{\partial z} = 2z.$$

Entonces, $(f_x, f_y, f_z) \neq (0, 0, 0)$ en $f^{-1}(r^2)$, por lo que r^2 es un valor regular. Por lo tanto, el toro es una superficie regular.

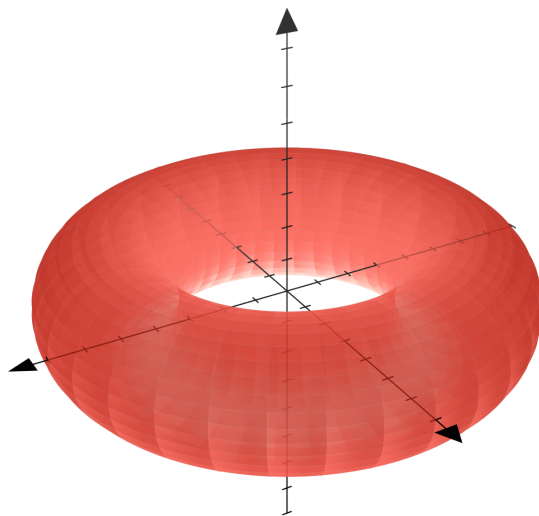


FIGURA 3.3: El toro también es una superficie de revolución.

Una parametrización para el toro anterior está dada por

$$\mathbf{x}(u, v) = ((r \cos u + a) \cos v, (r \cos u + a) \sin v, r \sin u),$$

con $u, v \in (0, 2\pi)$, $a > r > 0$. De la parametrización anterior, se puede concluir que el toro también es una superficie de revolución, como veremos a continuación.

Superficies de revolución

Las superficies de revolución se generan a partir de rotar una curva plana alrededor de un eje de revolución contenido en el mismo plano de la curva. Consideremos (x_1, x_2, x_3) coordenadas cartesianas arbitrarias en $\mathbb{R}^3$ y una parametrización de una curva regular $\alpha(t) = (f(t), 0, g(t))$ para $t \in (a, b) \subset \mathbb{R}$, con $f(t) > 0$ contenida en el plano x_1x_3 . Al rotar la curva sobre el eje x_3 , obtenemos el subconjunto S de $\mathbb{R}^3$ determinado por la imagen de la siguiente función

$$S = \{(f(v) \cos u, f(v) \sin u, g(v)) \in \mathbb{R}^3 : u \in [0, 2\pi] ; v \in (a, b)\},$$

con $v \in (a, b)$ y $u \in (0, 2\pi)$. Vamos a mostrar que este subconjunto es una superficie regular en la siguiente proposición.

Proposición 3.1.9. *Una superficie de revolución es una superficie regular.*

Demostración. Para verificarlo, definamos la función

$$\mathbf{x} : (0, 2\pi) \times (a, b) \rightarrow \mathbb{R}^3, \quad \mathbf{x}(u, v) = (f(v) \cos u, f(v) \sin v, g(v))$$

es una parametrización cuya imagen es la superficie de revolución que se obtiene al rotar la curva γ alrededor del eje z . ■

Ejemplo 3.1.10. *Algunos ejemplos de superficies de revolución son los siguientes*

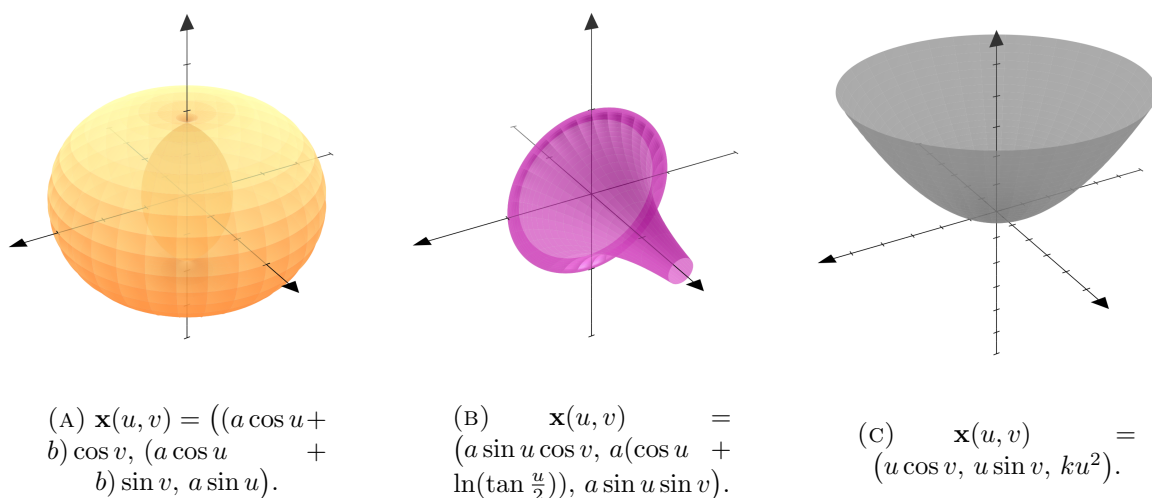


FIGURA 3.4: Ejemplos de superficies de revolución.

Con este código podemos graficar superficies en *Manim*,

```

1      surface = Surface(
2          lambda u, v: axes.c2p(*self.func(u, v)),
3          u_range=[0, 2*PI], v_range=[0, 2*PI],
4          resolution=32, checkerboard_colors = [RED],
5          stroke_width= 0.05, fill_opacity = 0.7
6      )

```

donde `*self.func(u, v)` es la función que buscamos graficar. En este caso es para graficar un toro.

```

1      def func(self, u, v):
2          R = 5
3          r = 2
4          return np.array([
5              (R + r * np.sin(u)) * np.cos(v),
6              (R + r * np.sin(u)) * np.sin(v),
7              r * np.cos(u)
8          ])

```

Lema 3.1.11. Sea S una superficie. Si $\alpha : I \rightarrow \mathbf{x}(D) \subseteq S$ es una curva en $\mathbb{R}^3$ contenida en la imagen de una parametrización $\mathbf{x}$ en S , entonces, para funciones suaves únicas $u(t)$, $v(t) : I \rightarrow \mathbb{R}$,

$$\alpha(t) = \mathbf{x}(u(t), v(t)).$$

Superficies regladas

Definición 3.1.12. Una superficie es **reglada** si tiene una parametrización de la forma

$$\mathbf{x}(u, v) = \beta(u) + v\delta(u),$$

donde β y δ son curvas. Esto es, la superficie entera está cubierta por esta parametrización que consiste de líneas que emanan de una curva $\beta(u)$ yendo en la dirección $\delta(u)$. La curva $\beta(u)$ es llamada *directriz* de la superficie y una línea teniendo a $\delta(u)$ como vector de dirección se le llama *regla*.

Una superficie es *doblemente reglada* si tiene dos parametrizaciones diferentes $\mathbf{x}(u, v) = \beta(u) + v\delta(u)$ y $\mathbf{y}(u, v) = \alpha(u) + v\gamma(u)$.

Ejemplo 3.1.13. Algunos ejemplos de superficies regladas son:

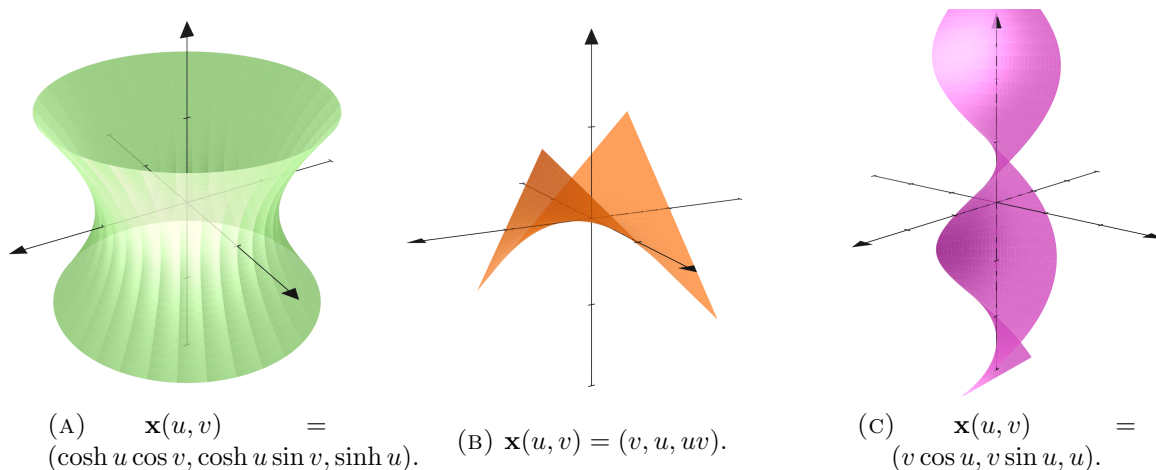


FIGURA 3.5: Ejemplos de superficies regladas.

3.2. El plano tangente de superficies regulares

Mostraremos que la definición de superficie regular implica la existencia de un plano tangente en cada punto de la superficie. También hablaremos de cuándo una función se dice ser suave y sus connotaciones.

El plano tangente a una superficie regular

Sea S una superficie regular y $p \in S$. Un vector $w \in \mathbb{R}^3$ se dice ser tangente a S en p si existe $\varepsilon > 0$ y una curva parametrizada, $\gamma : (-\varepsilon, \varepsilon) \rightarrow S$, diferenciable tal que:

- (i) $\gamma(0) = p$,
- (ii) $\gamma'(0) = w$.

El conjunto de vectores tangentes a S en p es denotado por $T_p S$. La condición 3 (regularidad) de la definición de superficie regular garantiza que el conjunto de vectores tangentes a S en $p \in S$ es un espacio vectorial de dimensión 2.

Proposición 3.2.1. *El subespacio vectorial de dimensión 2, $D_q \mathbf{x}(\mathbb{R}^2) \subset \mathbb{R}^3$, coincide con $T_p S$, i.e., $D_q \mathbf{x}(\mathbb{R}^2) = T_p S$.*

Demostración. Sea $h \in \mathbb{R}^2$ un vector tangente a q en $\mathbb{R}^2$, es decir, existe $\beta : (-\delta, \delta) \rightarrow U \subset \mathbb{R}^2$, diferenciable, tal que $\beta(0) = q$ y $\beta'(0) = h$. Definamos $\alpha : (-\delta, \delta) \rightarrow S$, $\alpha(t) = \mathbf{x} \circ \beta(t)$; α es diferenciable y $\alpha(0) = \mathbf{x}(\beta(0)) = \mathbf{x}(q) = p$. Entonces, α es una curva diferenciable en S que pasa por p , $\alpha'(0) \in T_p S$. Por la regla de la cadena, $\alpha'(t) = D_{\beta(t)} \mathbf{x}(\beta'(t))$, y así,

$$\alpha'(0) = D_q \mathbf{x}(h) \in T_p S \quad \forall h \in \mathbb{R}^2.$$

Por otra parte, sea $w \in T_p S$, es decir, existe $\alpha : (-\delta, \delta) \rightarrow S$, diferenciable, tal que $\alpha(0) = p$, $\alpha'(0) = w$. Sin pérdida de generalidad, $\alpha(-\delta, \delta) \subset \mathbf{x}(U)$. Ahora, definamos $\beta := \mathbf{x}^{-1} \circ \alpha : (-\delta, \delta) \rightarrow U$ es diferenciable. $\beta(0) = \mathbf{x}^{-1}(\alpha(0)) = \mathbf{x}^{-1}(p) = q$, $h := \beta'(0)$. $D_q \mathbf{x}(h) = w$; $\mathbf{x} : (\beta(t) = \mathbf{x}^{-1} \circ \alpha(t))$. Luego, $\mathbf{x} \circ \beta(t) = \alpha(t)$. Por la regla de la cadena, $D_q(\beta'(0)) = \alpha'(0) = w$. ■

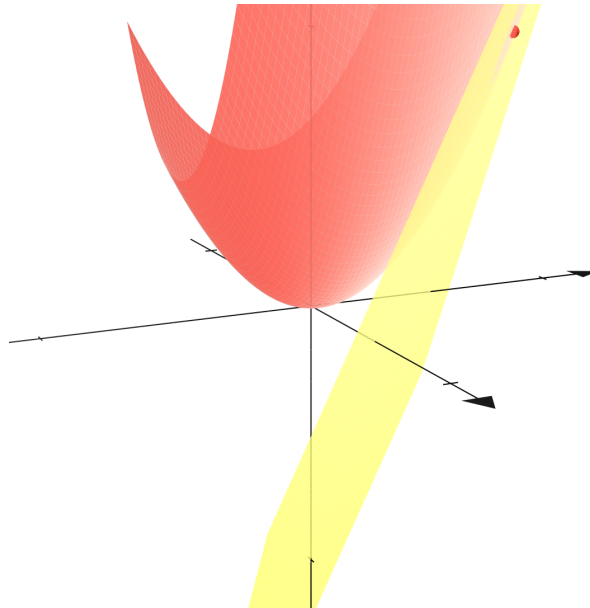
Proposición 3.2.2. *Una parametrización $\mathbf{x} : U \subset \mathbb{R}^2 \rightarrow S$ de S en p induce una base en $T_p S$*

Demostración. Supongamos que $\mathbf{x}(u_0, v_0) = p$ para algún $(u_0, v_0) \in U$. Sea $w \in \mathbb{R}^3$ un vector tangente a S en p , es decir, existe una curva $\gamma : (-\varepsilon, \varepsilon) \rightarrow S$, diferenciable, tal que $\gamma(0) = p$, $\gamma'(0) = w$. Entonces, existe $\beta : (-\varepsilon, \varepsilon) \rightarrow U$, una curva diferenciable, $\beta(t) = (u(t), v(t))$ tal que $\beta(0) = (u_0, v_0)$ y $\gamma(t) = \mathbf{x} \circ \beta(t)$. En virtud de la regla de la cadena,

$$w = \gamma'(0) = \left. \frac{d}{dt} \right|_{t=0} (\mathbf{x}(u(t), v(t))) = u'(0) \mathbf{x}_u(u_0, v_0) + v'(0) \mathbf{x}_v(u_0, v_0).$$

De aquí se sigue que todo vector $w \in T_p S$ se puede expresar como una combinación lineal de los vectores $\mathbf{x}_u(u_0, v_0)$, $\mathbf{x}_v(u_0, v_0)$. Como estos vectores son linealmente independientes (condición de regularidad), entonces $\{\mathbf{x}_u(u_0, v_0), \mathbf{x}_v(u_0, v_0)\}$ es una base de $T_p S$. ■

Ejemplo 3.2.3. *Mostramos un ejemplo de la visualización del plano tangente a una superficie definida por $\mathbf{x}(u, v) = (u, v, u^2 + v^2)$ en el punto P con coordenadas $(1, 1, 2)$.*

FIGURA 3.6: Plano tangente a la superficie dada en el punto P .

Con el siguiente código calculamos el plano tangente; primero simbólicamente y después numéricamente.

```

1  def plano_tangente():
2      u, v = sy.symbols('u v')
3      f_expr = u**2 + v**2
4
5      fu = sy.diff(f_expr, u)
6      fv = sy.diff(f_expr, v)
7
8      u0, v0 = 1, 1
9      f0 = f_expr.subs({u: u0, v: v0})
10     fu0 = fu.subs({u: u0, v: v0})
11     fv0 = fv.subs({u: u0, v: v0})
12
13     z_tangent_expr = f0 + fu0 * (u - u0) + fv0 * (v - v0)
14     z_tangent_func = sy.lambdify((u, v), z_tangent_expr, 'numpy')
15
16     def plano_func(u_vals, v_vals):
17         return np.array([u_vals, v_vals, z_tangent_func(u_vals, v_vals)])
18
19     return plano_func

```

Funciones suaves

Definición 3.2.4. Sea $f : V \subset S \rightarrow \mathbb{R}$, una función definida en un subconjunto abierto V de la superficie regular S . Se dice que f es una función suave en $p \in V$ si existe una parametrización $\mathbf{x} : U \rightarrow \mathbb{R}^3$ de S , con $p \in \mathbf{x}(U) \subset V$, de tal manera que la función

$f \circ \mathbf{x} : U \rightarrow \mathbb{R}$ es suave en el punto $\mathbf{x}^{-1}(p)$. La función f es suave en todo el conjunto V si es suave en cada punto de V .

Ahora definiremos el concepto de funciones suaves entre superficies regulares.

Definición 3.2.5. Una función continua $F : V_1 \subset S_1 \rightarrow S_2$, de un abierto V_1 de la superficie regular S_1 en la superficie regular S_2 se dice ser diferenciable en el punto $p \in V_1$ si dadas las parametrizaciones

$$\mathbf{x}_1 : U_1 \rightarrow S_1, \quad \mathbf{x}_2 : U_2 \rightarrow S_2,$$

con $p \in \mathbf{x}_1(U_1)$ y $F(\mathbf{x}_1(U_1)) \subset \mathbf{x}_2(U_2)$, la función

$$\mathbf{x}_2^{-1} \circ F \circ \mathbf{x}_1 : U_1 \rightarrow U_2$$

es diferenciable en $q = \mathbf{x}_1^{-1}(p)$.

La diferencial de una función suave

Definición 3.2.6. Sean S_1 y S_2 dos superficies regulares en $\mathbb{R}^3$ y $F : S_1 \rightarrow S_2$ una función suave. La diferencial de F en p , denotada por $d_p F$, es la transformación lineal de $d_p F : T_p S_1 \rightarrow T_{F(p)} S_2$ definida por

$$d_p F(\alpha'(0)) = (F \circ \alpha)'(0),$$

donde $\alpha : (-\epsilon, \epsilon) \rightarrow S_1$ es una curva suave en S_1 tal que $\alpha(0) = p$.

El diferencial de F está bien definido, pues si α, β son dos curvas en S_1 tales que $\alpha(0) = \beta(0) = p$ y $\alpha'(0) = \beta'(0)$, y por la regla de la cadena, $(F \circ \alpha)'(0) = (F \circ \beta)'(0)$.

Por otro lado, el diferencial $d_p F$ es una transformación lineal y, al fijar una base para el espacio $T_p S_1$, es posible representarla de forma matricial. Sea $\alpha : (-\epsilon, \epsilon) \rightarrow S_1$ una curva parametrizada con $\alpha(0) = p$. Definimos la curva $\beta = F \circ \alpha$, la cual es una curva en la superficie S_2 y se tiene $\beta(0) = F(p)$. Sean $\mathbf{x}(u, v)$ y $\mathbf{y}(w, z)$ parametrizaciones de S_1 y S_2 respectivamente, tales que $p \in \mathbf{x}(U)$ y $F(p) \in \mathbf{y}(V)$. Supongamos que F se expresa en coordenadas locales por $F(u, v) = (F_1(u, v), F_2(u, v))$, y que la curva α está dada por

$$\alpha(t) = (u(t), v(t)), \quad t \in (-\epsilon, \epsilon).$$

Entonces

$$\beta(t) = (F_1(u(t), v(t)), F_2(u(t), v(t))),$$

y la expresión de $\beta'(0)$ en la base $\{\mathbf{y}_w, \mathbf{y}_z\}$ es

$$\beta'(0) = \left(\frac{\partial F_1}{\partial u} u'(0) + \frac{\partial F_1}{\partial v} v'(0), \frac{\partial F_2}{\partial u} u'(0) + \frac{\partial F_2}{\partial v} v'(0) \right). \quad (3.2)$$

La ecuación (3.2) muestra que $\beta'(0)$ depende sólo de la función F y de las coordenadas $(u'(0), v'(0))$ del vector $\alpha'(0)$ en la base $\{\mathbf{x}_u, \mathbf{x}_v\}$. En forma matricial se tiene

$$\beta'(0) = d_p F(\alpha'(0)) = \begin{bmatrix} \frac{\partial F_1}{\partial u} & \frac{\partial F_1}{\partial v} \\ \frac{\partial F_2}{\partial u} & \frac{\partial F_2}{\partial v} \end{bmatrix} \begin{pmatrix} u'(0) \\ v'(0) \end{pmatrix}, \quad (3.3)$$

que es la representación matricial de la transformación lineal $d_p F$ (3.3).

3.3. Primera forma fundamental, longitudes, ángulos y áreas

Desde Euclides, hablar de geometría es hablar de longitudes, ángulos y áreas; es decir, de cuestiones métricas. ¿Cómo podemos estudiar estas propiedades geométricas en distintas superficies? A continuación veremos que la primera forma fundamental resulta natural para dilucidar esta cuestión.

El producto interior usual en $\mathbb{R}^3$ induce en cada plano tangente, $T_p S$, de una superficie regular un producto interior, denotado por $\langle \cdot, \cdot \rangle_p$,

$$\begin{aligned} \langle \cdot, \cdot \rangle_p : T_p S \times T_p S &\rightarrow \mathbb{R} \\ \langle v, w \rangle_p &:= \langle \cdot, \cdot \rangle = v_1 w_1 + v_2 w_2 + v_3 w_3, \end{aligned}$$

con $v = (v_1, v_2, v_3)$ y $w = (w_1, w_2, w_3)$.

Definición 3.3.1. La forma cuadrática I_p en $T_p S$, definida por

$$\begin{aligned} I_p : T_p S \times T_p S &\rightarrow \mathbb{R} \\ I_p &:= \langle \cdot, \cdot \rangle = v_1 w_1 + v_2 w_2 + v_3 w_3, \end{aligned}$$

es llamado la primera forma fundamental de una superficie regular S en p .

Primer forma fundamental en coordenadas locales

Sea $S \subset \mathbb{R}^3$ una superficie regular, $p \in S$ y $\mathbf{x} : U \subset \mathbb{R}^2 \rightarrow S$ una parametrización que cubre a p . Es decir, existe (u_0, v_0) tal que $\mathbf{x}(u_0, v_0) = p$.

Sea $w \in T_p S$ un vector tangente a S en p . Como analizamos previamente, una base de $T_p S$ es $\{\mathbf{x}_u(u_0, v_0), \mathbf{x}_v(u_0, v_0)\}$. Por lo que existen números reales a, b tales que

$$w = a\mathbf{x}_u + b\mathbf{x}_v \quad (3.4)$$

Por simplicidad, omitiremos la evaluación de $\mathbf{x}_u, \mathbf{x}_v$ en el punto (u_0, v_0) . Si calculamos el valor de la forma cuadrática I_p en el vector w utilizando la representación (3.4) se obtiene

$$\begin{aligned} I_p(w) &= \langle w, w \rangle_p \\ &= \langle a\mathbf{x}_u + b\mathbf{x}_v, a\mathbf{x}_u + b\mathbf{x}_v \rangle_p \\ &= a^2 \langle \mathbf{x}_u, \mathbf{x}_u \rangle_p + 2ab \langle \mathbf{x}_u, \mathbf{x}_v \rangle_p + b^2 \langle \mathbf{x}_v, \mathbf{x}_v \rangle_p \\ &= a^2 E + 2ab F + b^2 G. \end{aligned}$$

Aquí las funciones E, F, G son definidas por $E = \langle \mathbf{x}_u, \mathbf{x}_u \rangle$, $F = \langle \mathbf{x}_u, \mathbf{x}_v \rangle$, $G = \langle \mathbf{x}_v, \mathbf{x}_v \rangle$ son llamados los coeficientes de la Primera Forma Fundamental asociados a la parametrización $\mathbf{x} : U \subset \mathbb{R}^2 \rightarrow S$.

La primera forma fundamental nos permitirá hacer «mediciones» sobre la superficie regular.

$$\begin{aligned} \gamma : (a, b) &\rightarrow \mathbb{R}^3 \quad \text{diferenciable,} \\ \text{long}(\gamma) &= \int_a^b \|\gamma'(t)\| dt \\ \gamma : (a, b) &\rightarrow S \subset \mathbb{R}^3, \quad \text{superficie regular,} \\ \text{long}(\gamma) &= \int_a^b \|\gamma'(t)\| dt = \int_a^b I_{\gamma(t)}(\gamma'(t)) dt \end{aligned}$$

Sea $v \in \mathbb{R}^3$ tangente a una superficie regular S en p si existe $\gamma : (-\varepsilon, \varepsilon) \rightarrow S$, $\mathcal{C}^\infty$, tal que $\gamma(0) = p$ y $\gamma'(0) = v$.

Teorema 3.3.2. *El área de la porción de una superficie parametrizada por $\mathbf{x}(u, v)$ correspondiente a $R \subseteq \mathbb{R}^2$ está dada por*

$$\text{Área} = \iint_R \left| \frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v} \right| du dv. \quad (3.5)$$

De manera equivalente, el área también puede ser expresada como

$$\text{Área} = \iint_R \sqrt{EG - F^2} du dv, \quad (3.6)$$

donde $E = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial \mathbf{x}}{\partial u}$, $F = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial \mathbf{x}}{\partial v}$ y $G = \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial \mathbf{x}}{\partial v}$.

A continuación se presenta un ejemplo de la aplicación del teorema anterior.

Ejemplo 3.3.3. *Consideremos la esfera unitaria parametrizada por*

$$\mathbf{x}(u, v) = (\cos u \sin v, \sin u \sin v, \cos v),$$

donde $u \in [0, 2\pi]$ y $v \in [0, \pi/2]$.

Calculando el área utilizando el teorema anterior, tenemos que

$$\frac{\partial \mathbf{x}}{\partial u} = (-\sin u \sin v, \cos u \sin v, 0) \quad \frac{\partial \mathbf{x}}{\partial v} = (\cos u \sin v, \sin u \cos v, -\sin v),$$

con $E = \sin^2 v$, $F = 0$, $G = 1$, por lo que

$$\text{Área} = \int_{u=0}^{2\pi} \int_{v=0}^{\pi/2} \sin v dv du = 2\pi. \quad (3.7)$$

Podemos realizar los cálculos de manera más rápida con Python, así como después implementarlo y visualizarlo con Manim. Con este código se realizan los cálculos del área.

```

1  # Variables
2  u, v = sy.symbols('u v')
3
4  # Parametrización
5  x = sy.Matrix([
6      sy.cos(u) * sy.sin(v),
7      sy.sin(u) * sy.sin(v),
8      sy.cos(v)
9  ])
10

```

```

11 # Derivadas parciales
12 xu = x.diff(u)
13 xv = x.diff(v)
14
15 # Producto cruz y su norma
16 cross = xu.cross(xv)
17 norm = sy.sqrt(cross.dot(cross))
18
19 # Integrando
20 area = sy.integrate(sy.integrate(norm, (v, 0, sy.pi / 2)), (u, 0, 2 * sy.pi))
21 print(f"Área de la superficie: {area.evalf():.6f}")

```

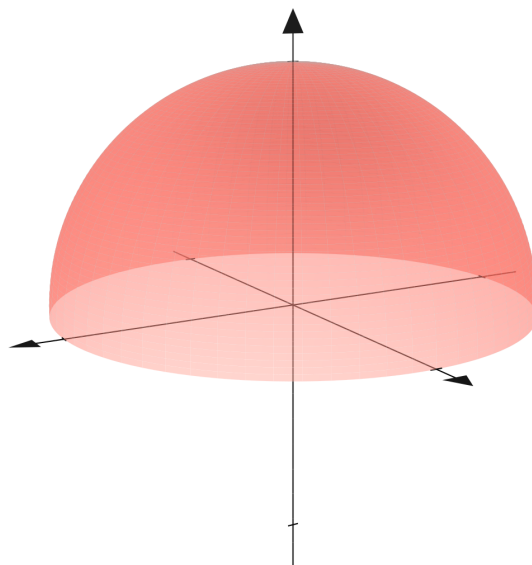


FIGURA 3.7: Semiesfera unitaria.

3.4. La aplicación de Gauss y segunda forma fundamental

Para comenzar esta sección, necesitamos introducir el concepto de superficie regular orientable.

Definición 3.4.1. Una superficie regular $S \subset \mathbb{R}^3$ se dice ser orientable si existe una función suave $\mathbf{N} : S \rightarrow \mathbb{S}^2$ con la propiedad de que para cada $p \in S$, $\mathbf{N}(p)$ es ortogonal al plano $T_p S$. Si tal función $\mathbf{N}$ existe, ésta es llamada una orientación para S .

Nota: Si S es orientable con orientación $N : S \rightarrow \mathbb{S}^2$, la función $\bar{\mathbf{N}} = -\mathbf{N}$ también es una orientación.

Observación 3.4.2. *Una definición equivalente del concepto de orientabilidad para superficies regulares puede consultarse en [3], [7], [6].*

Toda superficie S es localmente orientable. Si fijamos una parametrización $\mathbf{x} : U \rightarrow S$, de S , podemos definir la función $\mathbf{N}(p) : S \rightarrow \mathbb{S}^2$ por

$$\mathbf{N}(p) := \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}(u, v), \quad \text{donde } \mathbf{x}(u, v) = p, \quad (3.8)$$

define un campo de vectores normales unitarios en cada punto de $\mathbf{x}(U)$.

Definición 3.4.3. *Sea $S \subset \mathbb{R}^3$ una superficie orientable. La aplicación de Gauss es una orientación $\mathbf{N} : S \rightarrow \mathbb{S}^2$ para S .*

Observación 3.4.4. *Si S no es orientable, la aplicación de Gauss se puede definir localmente utilizando una parametrización.*

Como $\mathbf{N}$ es diferenciable, para cada $p \in S$, el diferencial

$$d_p \mathbf{N} : T_p S \rightarrow T_{\mathbf{N}(p)} \mathbb{S}^2$$

está bien definido. Además, notemos que los planos $T_p S$ y $T_{\mathbf{N}(p)} \mathbb{S}^2$ son paralelos, pues ambos son ortogonales a $\mathbf{N}(p)$. Por lo que, sin pérdida de generalidad, podemos asumir que $d_p \mathbf{N}$ es operador lineal del espacio $T_p S$ en sí mismo,

$$d_p \mathbf{N} : T_p S \rightarrow T_p S.$$

Proposición 3.4.5. *El diferencial de la aplicación de Gauss, $d\mathbf{N}_p : T_p S \rightarrow T_p S$ (??), es un operador lineal autoadjunto, esto es,*

$$\langle d\mathbf{N}_p(v), w \rangle = \langle v, d\mathbf{N}_p(w) \rangle, \quad (3.9)$$

para todo $v, w \in T_p S$. También se le conoce como el operador de forma de la superficie S en p .

Demostración. Puesto que el diferencial de $\mathbf{N}$ en p , $d\mathbf{N}_p$, es lineal; basta probar que

$$\langle d\mathbf{N}_p(v), w \rangle = \langle v, d\mathbf{N}_p(w) \rangle$$

para una base $\{v, w\}$ de $T_p(S)$.

Sea $x(u, v)$ una parametrización de S en p y $\{\mathbf{x}_u, \mathbf{x}_v\}$ las bases asociadas de $T_p(S)$. Si $\alpha(t) = x(u(t), v(t))$ es una curva parametrizada en S , con $\alpha(0) = p$, tenemos que

$$\begin{aligned} d\mathbf{N}_p(\alpha'(0)) &= d\mathbf{N}_p(\mathbf{x}_u u'(0) + \mathbf{x}_v v'(0)) \\ &= \frac{d}{dt} N(u(t), v(t))|_{t=0} \\ &= \mathbf{N}_u u'(0) + \mathbf{N}_v v'(0) \end{aligned}$$

En particular, $d\mathbf{N}_p(\mathbf{x}_u) = \mathbf{N}_u$ y $d\mathbf{N}_p(\mathbf{x}_v) = \mathbf{N}_v$, probándose así que

$$\langle \mathbf{N}_u, \mathbf{x}_v \rangle = \langle \mathbf{x}_u, \mathbf{N}_v \rangle$$

es decir, el diferencial de aplicación de Gauss es una aplicación autoadjunta. ■

A continuación, vamos a introducir la segunda forma fundamental

Definición 3.4.6. Sea $S \subset \mathbb{R}^3$ una superficie regular orientable con orientación $N : S \rightarrow \mathbb{S}^2$. La función $\mathbb{I}_p : T_p S \rightarrow \mathbb{R}$, dada por

$$\mathbb{I}_p(w) = -\langle d\mathbf{N}_p(w), w \rangle, \quad w \in T_p S,$$

y es llamada la segunda forma fundamental de S en p .

En la siguiente sección vamos a dar una interpretación geométrica de la segunda forma fundamental.

Curvatura normal y segunda forma fundamental.

Sea $S \subset \mathbb{R}^3$ una superficie regular orientable y fijemos una orientación $\mathbf{N} : S \rightarrow \mathbb{S}^2$. Dado un punto $p \in S$, un número real $\delta > 0$ y $\alpha : (-\delta, \delta) \rightarrow S$ una curva regular en S parametrizada por longitud de arco tal $\alpha(0) = p$. Definamos $\mathbf{N}(t) := \mathbf{N}(\alpha(t))$. La curvatura normal de α en t se define por

$$\kappa_n(t) := \langle \ddot{\alpha}(t), \mathbf{N}(t) \rangle = \kappa(t) \cdot \cos \theta(t), \quad (3.10)$$

donde $\kappa(t)$ es la curvatura de $\alpha(t)$ y $\theta(t)$ el ángulo que forma $\ddot{\alpha}(t)$ con $\mathbf{N}(t)$. En otras palabras, la curvatura normal es la proyección del vector aceleración $\ddot{\alpha}(t)$ sobre el vector normal.

Como $\langle \dot{\alpha}(t), \mathbf{N}(t) \rangle = 0$,

$$\kappa_n(t) = \langle \ddot{\alpha}(t), \mathbf{N}(t) \rangle = -\langle \dot{\alpha}(t), d_p \mathbf{N}(\dot{\alpha}(t)) \rangle = \mathbb{I}_p(\dot{\alpha}(t)).$$

De la ecuación anterior tenemos dos conclusiones

- i) El valor de la curvatura normal depende del vector tangente a la curva. Es decir, dos curvas en una superficie regular que tienen el mismo vector tangente en un punto, tienen la misma curvatura normal.
- ii) En un punto $p \in S$ y $w \in T_p S$ tal que $\|w\|_p = 1$, $\mathbb{I}_p(w)$ es el valor de la curvatura normal de toda curva en S que pasa por p cuyo vector tangente en p es w .

Si restringimos la segunda forma fundamental $\mathbb{I}_p$, a vectores en $T_p S$ de norma uno, tenemos que $\mathbb{I}_p$ alcanza su máximo y su mínimo valor. Más aún, como el operador de forma $d_p \mathbf{N}$ es un operador autoadjunto, los valores propios de $d_p \mathbf{N}$ son los máximos y mínimos de $\mathbb{I}_p$ y los vectores propios son ortogonales.

Lo anterior motiva la siguiente definición.

Definición 3.4.7. Sea $S \subset \mathbb{R}^3$ una superficie regular en $\mathbb{R}^3$, orientable con orientación $\mathbf{N} : S \rightarrow \mathbb{S}^2$. Para un punto $p \in S$, las curvaturas principales en p son el máximo y el mínimo de $\mathbb{I}_p$. Las direcciones principales en p son los vectores tangentes unitarios donde se alcanza el máximo y el mínimo de $\mathbb{I}_p$.

Segunda forma fundamental en coordenadas locales

Sea $\mathbf{x}(u, v)$ una parametrización de la superficie S ; $p \in S$ y $\alpha(t) = \mathbf{x}(u(t), v(t))$ una curva parametrizada en S , con $\alpha(0) = p$. El vector tangente a $\alpha(t)$ en p es

$$\alpha' = u' \mathbf{x}_u + v' \mathbf{x}_v.$$

Consideremos la aplicación de Gauss

$$\mathbf{N}(u, v) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}$$

Aplicando el diferencial de la aplicación de Gauss al vector tangente α' , tenemos

$$d\mathbf{N}_p(\alpha') = \frac{d\mathbf{N}}{dt}(u(t), v(t)) = u' \mathbf{N}_u + v' \mathbf{N}_v, \quad (3.11)$$

donde $\mathbf{N}_u = \frac{\partial \mathbf{N}}{\partial u}$ y $\mathbf{N}_v = \frac{\partial \mathbf{N}}{\partial v}$.

Como $\langle \mathbf{N}, \mathbf{N} \rangle = 1$, se sigue que $\mathbf{N}_u$ y $\mathbf{N}_v$ son ambos ortogonales a $\mathbf{N}$, y por ende pertenecen a $T_p S$. Lo anterior nos lleva a que $\mathbf{N}_u$ y $\mathbf{N}_v$ se pueden expresar como una combinación lineal de $\mathbf{x}_u$ y $\mathbf{x}_v$; así,

$$\mathbf{N}_u = a_{11}\mathbf{x}_u + a_{21}\mathbf{x}_v \quad (3.12)$$

$$\mathbf{N}_v = a_{12}\mathbf{x}_u + a_{22}\mathbf{x}_v. \quad (3.13)$$

Sustituyendo (3.12) y (3.13) en (3.11), se tiene

$$d\mathbf{N}_p(\alpha') = (a_{11}u' + a_{12}v')\mathbf{x}_u + (a_{21}u' + a_{22}v')\mathbf{x}_v, \quad (3.14)$$

que expresándolo de manera matricial es

$$d\mathbf{N} \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix}. \quad (3.15)$$

Ahora, para la segunda forma fundamental utilizando la base $\{\mathbf{x}_u, \mathbf{x}_v\}$ tenemos

$$\begin{aligned} \mathbb{I}_p(\alpha') &= -\langle d\mathbf{N}_p(\alpha'), \alpha' \rangle \\ &= -\langle u'\mathbf{N}_u + v'\mathbf{N}_v, u'\mathbf{x}_u + v'\mathbf{x}_v \rangle \\ &= e(u')^2 + 2fu'v' + g(v')^2, \end{aligned}$$

con los coeficientes e, f, g , están dados por:

$$e = -\langle \mathbf{N}_u, \mathbf{x}_u \rangle = \langle \mathbf{N}, \mathbf{x}_{uu} \rangle, \quad (3.16)$$

$$f = -\langle \mathbf{N}_v, \mathbf{x}_u \rangle = \langle \mathbf{N}, \mathbf{x}_{uv} \rangle = \langle \mathbf{N}, \mathbf{x}_{vu} \rangle = -\langle \mathbf{N}_u, \mathbf{x}_v \rangle, \quad (3.17)$$

$$g = -\langle \mathbf{N}_v, \mathbf{x}_v \rangle = \langle \mathbf{N}, \mathbf{x}_{vv} \rangle. \quad (3.18)$$

Más aún, podemos expresar los coeficientes a_{ij} de la representación matricial del operador de forma en (3.15). Combinando (3.12) y (3.13) con (3.16), (3.17) y (3.18) obtenemos las

siguientes ecuaciones

$$\begin{aligned} -e &= a_{11}E + a_{21}F, \\ -f &= a_{11}F + a_{21}G, \\ -f &= a_{12}E + a_{22}F, \\ -g &= a_{21}F + a_{22}G, \end{aligned}$$

A partir de estas ecuaciones, tenemos

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = -\frac{1}{EG - F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix} \begin{bmatrix} e & f \\ f & g \end{bmatrix} \quad (3.19)$$

Curvatura media y curvatura Gaussiana

Sea $S \subset \mathbb{R}^3$ una superficie regular orientable y fijemos una aplicación de Gauss $\mathbf{N} : S \rightarrow \mathbb{S}^2$. Recordemos que, para cada $p \in S$, $d_p\mathbf{N} : T_pS \rightarrow T_pS$ es un operador lineal autoadjunto.

Como todo operador lineal tiene con invariantes a su traza y determinante, vamos a concentrarnos en estos invariantes para el caso de $d_p\mathbf{N}$ (operador de forma) e interpretarlos geométricamente.

Definición 3.4.8. Sea $S \subset \mathbb{R}^3$ una superficie regular en $\mathbb{R}^3$, orientable, con orientación $\mathbf{N} : S \rightarrow \mathbb{S}^2$. Definimos:

(i) La curvatura gaussiana de S por la función

$$\begin{aligned} K &: S \rightarrow \mathbb{R}, \\ K(p) &:= \det(d_p\mathbf{N}). \end{aligned}$$

(ii) La curvatura media de S por

$$\begin{aligned} H &: S \rightarrow \mathbb{R}, \\ H(p) &:= -\frac{1}{2} \operatorname{tr}(d_p\mathbf{N}). \end{aligned}$$

Recordemos que una superficie regular orientable tiene dos orientaciones, es decir, si $\mathbf{N} : S \rightarrow \mathbb{S}^2$ es una orientación de S , entonces $\bar{\mathbf{N}} = -\mathbf{N}$ es también una orientación.

Como

$$\operatorname{tr}(d\bar{\mathbf{N}}) = -\operatorname{tr}(d\mathbf{N}) \quad \text{y} \quad \det(d\bar{\mathbf{N}}) = \det(d\mathbf{N}),$$

tenemos que la curvatura Gaussiana es independiente de la elección de la orientación, mientras que la curvatura media es independiente salvo signo.

En la siguiente proposición se expresan la curvatura media y Gaussiana en coordenadas locales. Además de su utilidad para hacer cálculos explícitos, estas fórmulas muestran que las curvaturas media y Gaussiana son funciones C^∞ .

Proposición 3.4.9. *Sea $S \subset \mathbb{R}^3$ una superficie regular orientable con orientación $\mathbf{N} : S \rightarrow \mathbb{S}^2$. Si $\mathbf{x} : U \rightarrow \mathbb{R}^3$ es una parametrización de S tal que*

$$\mathbf{N} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|},$$

entonces,

$$K = \frac{eg - f^2}{EG - F^2} \quad \text{y} \quad H = -\frac{eG - 2fF + gE}{EG - F^2}.$$

Demostración. Notemos que siempre existe una parametrización tal que $\mathbf{N} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}$. Pues en caso de que no se cumpla para una parametrización dada, solo se deben intercambiar variables. Ahora si consideramos la representación matricial de $d_p \mathbf{N}$ con respecto a la base $\{\mathbf{x}_u, \mathbf{x}_v\}$ de $T_p S$, obtenemos las fórmulas deseadas utilizando la ecuación (3.19). ■

La curvatura Gaussiana K se puede definir también para superficies no orientables utilizando parametrizaciones a través de la fórmula de la proposición anterior. Como la curvatura Gaussiana no depende de la orientación, tenemos una función C^∞ bien definida globalmente.

Finalmente, si κ_1 y κ_2 representan las curvaturas principales de S en p , entonces

$$K(p) = \kappa_1 \cdot \kappa_2 \quad \text{y} \quad H(p) = \frac{1}{2}(\kappa_1 + \kappa_2).$$

Ejemplo 3.4.10. *El cilindro, con la parametrización*

$$\mathbf{x}(u, v) = (\cos(u), \sin(u), v).$$

Sea $\mathbf{x}(u, v) = (\cos(u), \sin(u), v)$. Luego, derivando a $\mathbf{x}(u, v)$ con respecto a u y v ,

$$\mathbf{x}_u(u, v) = (-\sin(u), \cos(u), 0);$$

$$\mathbf{x}_v(u, v) = (0, 0, 1).$$

Procedemos ahora a calcular los coeficientes de la primera forma fundamental

$$E = \langle \mathbf{x}_u, \mathbf{x}_u \rangle = 1, \quad F = \langle \mathbf{x}_u, \mathbf{x}_v \rangle = 0, \quad G = \langle \mathbf{x}_v, \mathbf{x}_v \rangle = 1.$$

Por otro lado,

$$\mathbf{x}_u \times \mathbf{x}_v = (\cos(u), \sin(u), 0)$$

y $\|\mathbf{x}_u \times \mathbf{x}_v\| = 1$. Así,

$$\mathbf{N}(u, v) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|} = (\cos(u), \sin(u), 0).$$

Derivando $\mathbf{N}(u, v)$ respecto a u y v ,

$$\mathbf{N}_u(u, v) = (-\sin(u), \cos(u), 0)$$

$$\mathbf{N}_v(u, v) = (0, 0, 0).$$

Calculemos ahora los coeficientes de la segunda forma fundamental

$$e = -\langle \mathbf{x}_u, n_u \rangle = -1, \quad f = -\langle \mathbf{x}_u, n_v \rangle = 0, \quad g = -\langle \mathbf{x}_v, n_v \rangle = 0.$$

Para la representación matricial del operador de forma se tiene

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = -\frac{1}{EG - F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix} \begin{bmatrix} e & f \\ f & g \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

La curvatura de Gauss es

$$K = \frac{eg - f^2}{EG - F^2} = 0,$$

la curvatura media es

$$H = \frac{1}{2} \frac{eG - 2fF + gE}{EG - F^2} = -0.5,$$

y las curvaturas principales son

$$k_1 = -H + \sqrt{H^2 - K} = 1$$

$$k_2 = -H - \sqrt{H^2 - K} = 0$$

En el cilindro todos los puntos son parabólicos, pues $K = 0$.

Los cálculos detrás del ejemplo anterior han sido programados para automatizarse en *Python*. También puede utilizarse para el caso de la silla del mono, aunque el programa no siempre hace la mejor simplificación de los resultados; por lo que se omitió por cuestiones de espacio. Aun así, he aquí el cómo se ve la silla del mono y el código detrás.

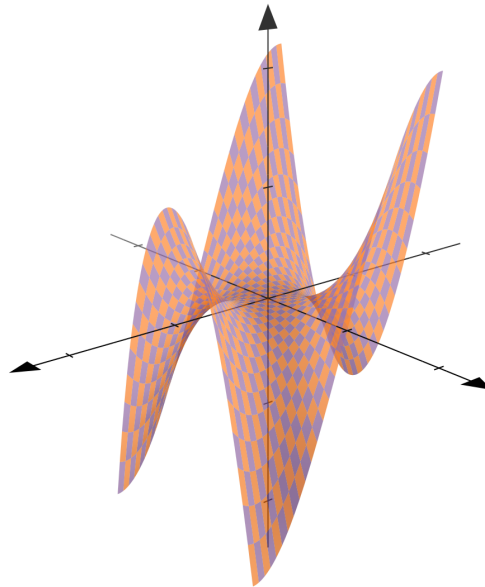


FIGURA 3.8: Silla del mono.

Código:

```

1  from sympy import *
2
3  u, v = symbols('u v')
4  f_u_v = (cos(u), sin(u), v)
5
6  x_u = tuple(diff(f, u) for f in f_u_v)
7  x_v = tuple(diff(f, v) for f in f_u_v)
8
9  E = Matrix([x_u]).dot(Matrix([x_u]))
10 F = Matrix([x_u]).dot(Matrix([x_v]))
11 G = Matrix([x_v]).dot(Matrix([x_v]))
12

```

```

13 cruz = Matrix([x_u]).cross(Matrix([x_v]))
14 norma_cruz = sqrt(sum(comp**2 for comp in cruz))
15
16 n_u_v = (cruz)/(simplify(norma_cruz))
17 n_u = tuple(diff(f, u) for f in n_u_v)
18 n_v = tuple(diff(f, v) for f in n_u_v)
19
20 e = -1*Matrix([x_u]).dot(Matrix([n_u]))
21 f = -1*Matrix([x_u]).dot(Matrix([n_v]))
22 g = -1*Matrix([x_v]).dot(Matrix([n_v]))
23
24 m0 = simplify(E)*simplify(G)-simplify(F)**2
25 m1 = Matrix([[simplify(G), -simplify(F)], [-simplify(F), simplify(E)]])
26 m2 = Matrix([[simplify(e), simplify(f)], [simplify(f), simplify(g)]])
27 mf = (-1/m0)*m1*m2
28
29 K = (simplify(e)*simplify(g)-simplify(f)**2)/(simplify(E)*simplify(G)-simplify(F)**2)
30 H = (1/2) * (simplify(e)*simplify(G)-2*simplify(f)*simplify(F)+simplify(g)*simplify(E)) /
31 (simplify(E)*simplify(G)-simplify(F)**2)
32 k_1 = -1 * simplify(H) + sqrt(simplify(H)**2 - simplify(K))
33 k_2 = -1 * simplify(H) - sqrt(simplify(H)**2 - simplify(K))

```

3.5. Superficies minimales o de curvatura media constante

Tras haber definido la curvatura media, veremos que, si ésta se anula, entonces la superficie es minimal en el sentido de que minimiza localmente su área.

Definición 3.5.1. Una superficie S se dice ser plana si $K(p) = 0$ para cada $p \in S$ y se dice ser minimal si $H(p) = 0$ para cada $p \in S$.

A continuación se muestran ejemplos de superficies minimales.

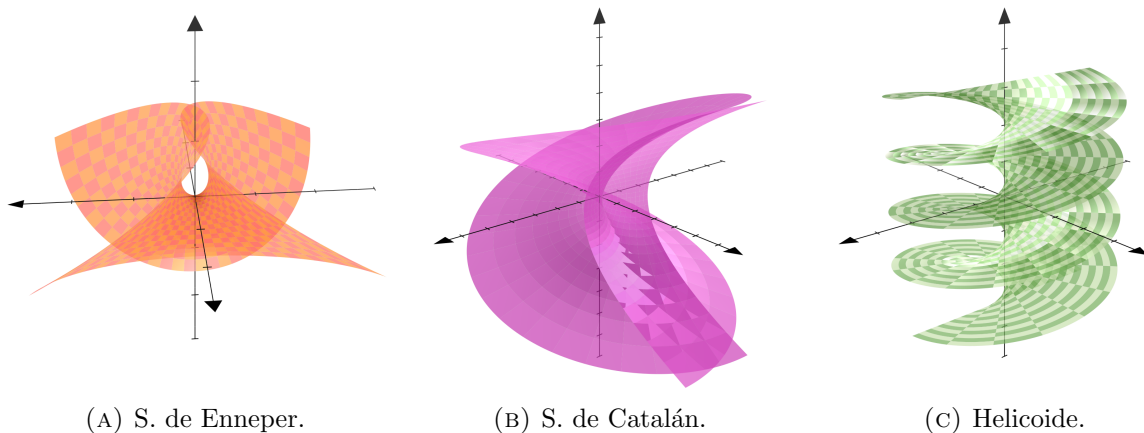


FIGURA 3.9: Ejemplos de superficies minimales. Aquí «S.» abrevia superficie.

He aquí la parametrización de cada una de las anteriores:

(A) $\mathbf{x}(u, v) = ((1/3)u(1 - (1/3)u^2 + v^2), (1/3)v(1 - (1/3)v^2 + u^2), (1/3)(u^2 - v^2)).$

(B) $\mathbf{x}(u, v) = (u - \sin u \cosh v, 1 - \cos u \cosh v, 4 \sin(u/2) \sinh(v/2)).$

(C) $\mathbf{x}(u, v) = u \cos v, u \sin v, cv.$

Nota: Si una superficie de revolución M es minimal, entonces M está contenida, ya sea, en un plano o en un catenoide.

Capítulo 4

Curvas geodésicas y sus propiedades

Los Elementos de Euclides es quizá uno de los escritos más importantes dentro de las matemáticas, si bien por su carácter histórico así como su impacto en las matemáticas del siglo XVIII y XIX, pues trajo consigo el desarrollo de geometrías no euclidianas. Extenderemos conceptos de la geometría diferencial a la geometría no necesariamente euclidiana y veremos que resulta natural hacerlo.

4.1. Isometrías y aplicaciones conformes

En esta sección veremos cuándo dos superficies preservan la distancia entre dos puntos tras haberse aplicado algún difeomorfismo, es decir, cuándo son isométricas. Primero introducimos el concepto de difeomorfismo. Para ello, sean S_1 y S_2 superficies regulares y $\varphi : S_1 \rightarrow S_2$, que es $\mathcal{C}^\infty$. Consideremos $p \in S_1$, $v \in T_p S$ y $\gamma : (-\varepsilon, \varepsilon) \rightarrow S_1$, también $\mathcal{C}^\infty$, con $\gamma(0) = p$ y $\gamma'(0) = v$.

Definición 4.1.1. Sean S_1 y S_2 superficies regulares en $\mathbb{R}^3$. Una función, $\varphi : S_1 \rightarrow S_2$ es llamada un difeomorfismo si

- (i) φ es $\mathcal{C}^\infty$.
- (ii) φ sea invertible.
- (iii) $\varphi : S_2 \rightarrow S_1$ es $\mathcal{C}^\infty$.

En este caso, se dice que S_1 y S_1 son superficies difeomorfas.

Ejemplo 4.1.2. Sea $\varphi : S^2 \rightarrow S^2$; $\varphi(P) = -P$. φ es un difeomorfismo. En efecto:

- (i) φ es $\mathcal{C}^\infty$.
- (ii) $\varphi \circ \varphi(P) = P$.
- (iii) $\varphi^{-1} = \varphi$ es $\mathcal{C}^\infty$.

Definición 4.1.3. Sea $\varphi : S_1 \rightarrow S_2$, $\mathcal{C}^\infty$. φ se dice ser un difeomorfismo local en $p \in S_1$, fijo, si existe una vecindad abierta, U , de p en $\mathbb{R}^3$ y una vecindad abierta, V , de $\varphi(p)$ en $\mathbb{R}^3$ tal que $\varphi : S_1 \cap U \rightarrow S_2 \cap V$ es un difeomorfismo.

Isometría en superficies regulares

Definición 4.1.4. Un difeomorfismo $\phi : S_1 \rightarrow S_2$ entre dos superficies regulares es una isometría si, para cada $p \in S_1$ y $w \in T_p S_1$, se tiene que

$$I_p(w) = I_{\phi(p)}(d_p \phi(w)).$$

Aquí I_p denota la primera forma fundamental.

Dos superficies se dicen ser isométricas, si hay una isometría entre ellas.

Proposición 4.1.5. Sea $\phi : S_1 \rightarrow S_2$ una isometría. Entonces para cada $p \in S_1$, $d_p \phi : T_p S_1 \rightarrow T_{\phi(p)} S_2$, es una isometría entre espacios vectoriales con producto interior, es decir,

$$\langle v, w \rangle_p = \langle d_p \phi(v), d_p \phi(w) \rangle,$$

para todo $v, w \in T_p S_1$.

Demostración. Esta proposición se sigue inmediatamente de la identidad de polarización: para $v, w \in T_p S_1$

$$\begin{aligned} \langle v, w \rangle_p &= \frac{1}{2} (\langle v + w, v + w \rangle_p - \langle v, v \rangle_p - \langle w, w \rangle_p) \\ &= \frac{1}{2} (I_p(v + w) - I_p(v) - I_p(w)) \\ &= \frac{1}{2} (I_{\phi(p)}(d_p \phi(v + w)) - I_{\phi(p)}(d_p \phi(v)) - I_{\phi(p)}(d_p \phi(w))) \\ &= \langle d_p \phi(v), d_p \phi(w) \rangle_{\phi(p)} \end{aligned}$$

■

Para estudiar propiedades locales de las isometrías, es útil el uso de coordenadas locales. Haciendo uso de ellas, se tiene el siguiente resultado

Proposición 4.1.6. *Si $\phi : S_1 \rightarrow S_2$ es una isometría y $p \in S_1$, entonces existen parametrizaciones $\mathbf{x} : U \rightarrow S_1$, con $p \in \mathbf{x}(U)$ y $\bar{\mathbf{x}} : U \rightarrow S_2$ con $\phi(p) \in \bar{\mathbf{x}}(U)$ tal que los coeficientes de la primera forma fundamental de cada parametrización son iguales, es decir, $E = \bar{E}$, $F = \bar{F}$ y $G = \bar{G}$.*

Demostración. Fijemos cualquier parametrización $\mathbf{x} : U \rightarrow S_1$, tal que $p \in \mathbf{x}(U)$. Definamos

$$\bar{\mathbf{x}} : U \rightarrow S_2$$

$$\bar{\mathbf{x}} = \phi \circ \mathbf{x}.$$

Probemos que $\bar{\mathbf{x}}$ tiene las propiedades deseadas. Notemos que $\bar{\mathbf{x}}_u = d_p\phi(\mathbf{x}_u)$ y $\bar{\mathbf{x}}_v = d_p\phi(\mathbf{x}_v)$. Como ϕ es isometría,

$$\begin{aligned} E(u, v) &= \langle \mathbf{x}_u, \mathbf{x}_u \rangle_p = \langle d_p\phi(\mathbf{x}_u), d_p\phi(\mathbf{x}_u) \rangle_{\phi(p)} \\ &= \langle \bar{\mathbf{x}}_u, \bar{\mathbf{x}}_u \rangle = \bar{E}(u, v) \end{aligned}$$

De forma análoga, se demuestra que $F = \bar{F}$ y $G = \bar{G}$. ■

De la proposición anterior podemos concluir que longitudes, ángulos y áreas son preservadas bajo isometrías.

La noción de isometrías entre superficies regulares es un concepto global. Para analizar propiedades locales de funciones que preserven la primera forma fundamental, se requiere una versión local del concepto de isometrías.

Definición 4.1.7. *Sea $\phi : S_1 \rightarrow S_2$ una función C^∞ . Se dice que ϕ es una isometría local en un punto $p \in S_1$ si existen abiertos V de p en S_1 y w de $\phi(p)$ en S_2 tal que $\phi : V \rightarrow w$ es una isometría. Dos superficies S_1, S_2 son localmente isométricas si existe una función $\phi : S_1 \rightarrow S_2$ que es una isometría local para cada $p \in S_1$.*

Nota: Una isometría $\phi : S_1 \rightarrow S_2$ es una isometría local pero, no toda isometría local es una isometría.

Ejemplo 4.1.8. *El cilindro y el plano son superficies localmente isométricas pero no isométricas.*

Consideremos el plano $S_1 = \{(x, y, 0) | z \in \mathbb{R}\}$ y el cilindro $S_2 = \{(x, y, z) | x^2 + y^2 = 1\}$. La función

$$\phi : S_1 \rightarrow S_2$$

$$\phi(x, y, 0) = (\cos x, \sin x, y).$$

Como ϕ no es inyectiva, entonces no puede ser un isometría. Probemos que ϕ es un isomorfismo local. Para cada punto $p_0 = (x_0, y_0, 0) \in S_1$, consideremos la vecindad abierta

$$V = \{(x, y, z) \in \mathbb{R}^3 | x \in (x_0 - \pi, x_0 + \pi)\}.$$

En esta vecindad, ϕ es un difeomorfismo sobre su imagen

$$\phi(V) = \{(\cos(u + x_0), \sin(u + x_0), v) | u \in (-\pi, \pi), v \in \mathbb{R}\}$$

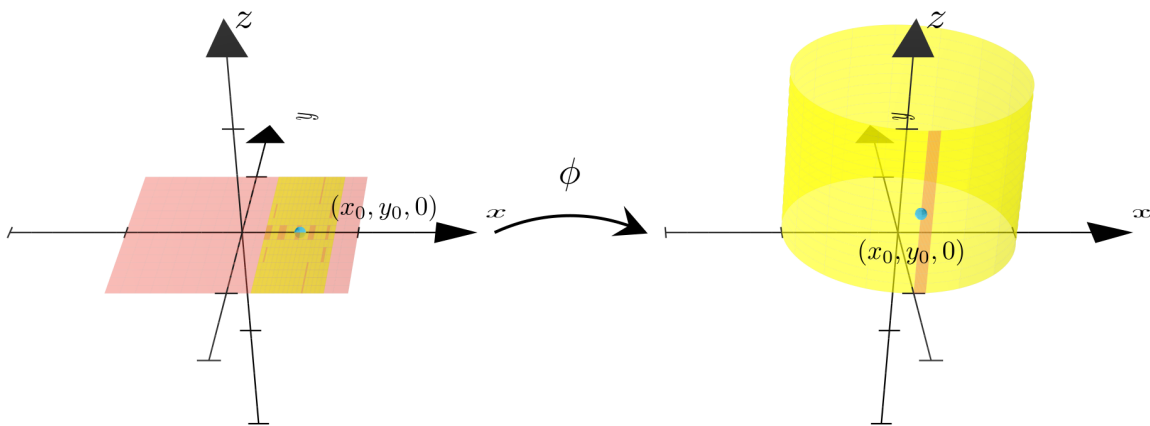


FIGURA 4.1: Representación visual.

Sobre el conjunto abierto $U = \{(u, v) \in \mathbb{R}^2 | u \in (x_0 - \pi, x_0 + \pi)\}$ de $\mathbb{R}^2$ definamos las siguientes parametrizaciones

$$\mathbf{x} : U \rightarrow V$$

$$\mathbf{x}(u, v) = (u, v, 0)$$

$$\bar{\mathbf{x}} : U \rightarrow \phi(V)$$

$$\bar{\mathbf{x}}(u, v) = (\cos u, \sin u, v)$$

Notemos que $E(u, v) = \bar{E}(u, v) = 1$, $F(u, v) = \bar{F}(u, v) = 0$ y $G(u, v) = \bar{G}(u, v) = 1$.

Para $w \in T_{p_0}S$, $w = u'(0)\mathbf{x}_u(u(0), v(0)) + v'(0)\mathbf{x}_v(u(0), v(0))$, para alguna curva $\beta : (-\delta, \delta) \rightarrow U$, $\beta(t) = (u(t), v(t))$ tal que $\beta(0) = (x_0, y_0)$. Entonces,

$$\begin{aligned} d\phi_{p_0}(w) &= \left. \frac{d}{dt} \right|_{t=0} (\phi(\mathbf{x}(u(t), v(t)))) = \left. \frac{d}{dt} \right|_{t=0} (\bar{\mathbf{x}}(u(t), v(t))) \\ &= u'(0)\bar{\mathbf{x}}_u(u_0, v_0) + v'(0)\bar{\mathbf{x}}_v(u_0, v_0). \end{aligned}$$

Por cálculo directo,

$$\begin{aligned} I_{p_0}(w) &= I_{p_0}(u'(0)\mathbf{x}_u(u_0, v_0) + v'(0)\mathbf{x}_v(u_0, v_0)) = (u'(0))^2 + (v'(0))^2, \\ I_{\phi(p_0)}(d\phi_{p_0}(w)) &= I_{\phi(p_0)}(u'(0)\bar{\mathbf{x}}_u(u_0, v_0) + v'(0)\bar{\mathbf{x}}_v(u_0, v_0)) = (u'(0))^2 + (v'(0))^2. \end{aligned}$$

Ejemplo 4.1.9. La catenoide y el helicoides son localmente isométricos.

Sea $\mathbf{x}(u, v) = (a \cosh v \cos u, a \cosh v \sin u, av)$ la parametrización de la catenoide y sea $\mathbf{y}(u, v) = (w \cos z, w \sin z, az)$ la parametrización del helicoides, donde $0 < u, z < 2\pi$ y $-\infty < v, w < \infty$.

Ahora, sea $z = u$; $w = a \sinh v$, lo cual es válido dado que el mapeo $(u, v) \mapsto (u, a \sinh v)$ es suave y biyectivo en la región donde u y v están definidas. Luego, la matriz de cambio de coordenadas está dada por

$$\begin{pmatrix} 1 & 0 \\ 0 & a \cosh v \end{pmatrix},$$

y el Jacobiano de este cambio de coordenadas proviene del determinante de esta matriz, $a \cosh v$, el cual es distinto de cero. En otras palabras, este cambio de coordenadas induce un difeomorfismo local entre estas dos superficies. Así, una nueva parametrización del helicoides es

$$\mathbf{y}(u, v) = (a \sinh v \cos u, a \sinh v \sin u, au).$$

Notemos que

$$\begin{aligned} \mathbf{y}_u &= (-a \sinh v \sin u, a \sinh v \cos u, a) \\ \mathbf{y}_v &= (a \cosh v \cos u, a \cosh v \sin u, 0) \end{aligned},$$

por lo que los términos de la primera forma fundamental del helicoides reparametrizado están dados por

$$\begin{aligned} E &= \langle \mathbf{y}_u, \mathbf{y}_u \rangle = a^2 \sinh^2 v + a^2 = a^2(\sinh^2 v + 1) = a^2 \cosh^2 v \\ F &= \langle \mathbf{y}_u, \mathbf{y}_v \rangle = -a^2 \sinh v \cosh v \sin u \cos u + a^2 \sinh v \cosh v \sin u \cos u = 0 \\ G &= \langle \mathbf{y}_v, \mathbf{y}_v \rangle = a^2 \cosh^2 v \end{aligned}$$

Por otro lado,

$$\begin{aligned} \mathbf{x}_u &= (-a \cosh v \sin u, a \cosh v \cos u, 0) \\ \mathbf{x}_v &= (a \sinh v \cos u, a \sinh v \sin u, a) \end{aligned} ,$$

así, los términos de la primera forma fundamental de la catenoide están dados por

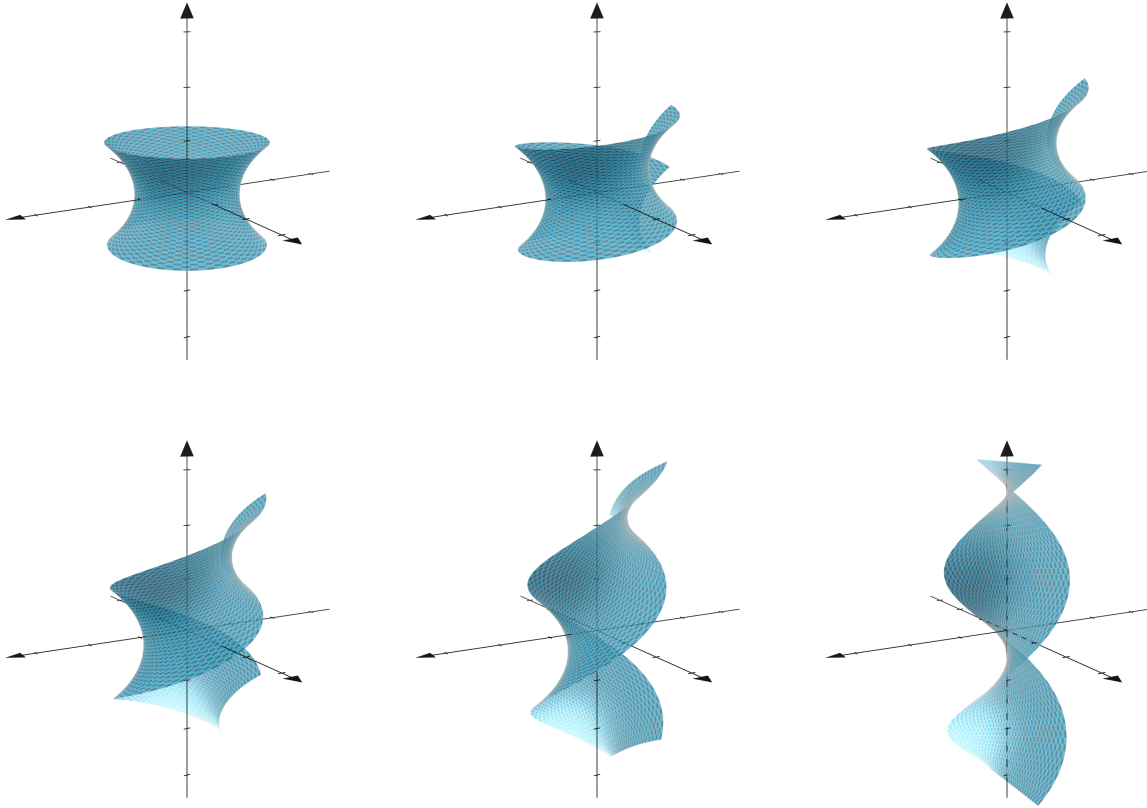
$$\begin{aligned} E &= \langle \mathbf{x}_u, \mathbf{x}_u \rangle = a^2 \cosh^2 v \\ F &= \langle \mathbf{x}_u, \mathbf{x}_v \rangle = -a^2 \sinh v \cosh v \sin u \cos u + a^2 \sinh v \cosh v \sin u \cos u = 0 \\ G &= \langle \mathbf{x}_v, \mathbf{x}_v \rangle = a^2 \sinh^2 v + a^2 = a^2(\sinh^2 v + 1) = a^2 \cosh^2 v \end{aligned}$$

Por lo tanto, el difeomorfismo local inducido por el cambio de coordenadas preserva la primera forma fundamental. Dado que la primera forma fundamental es sólo un reetiquetamiento de la métrica en la superficie. Esto quiere decir que este difeomorfismo local preserva la métrica; por ende, el helicoides y la catenoide son localmente isométricos.

Es posible visualizar la transformación de un helicoides a una catenoide mediante el siguiente mapeo:

$$\begin{aligned} x(u, v) &= \cos \theta \sinh v \sin u + \sin \theta \cosh v \cos u \\ y(u, v) &= -\cos \theta \sinh v \cos u + \sin \theta \cosh v \sin u \\ z(u, v) &= u \cos \theta + v \sin \theta, \end{aligned}$$

para $(u, v) \in (-\pi, \pi] \times (-\infty, \infty)$, con parámetros de deformación $-\pi < \theta \leq \pi$, donde $\theta = \pi$ tendríamos un helicoides diestro; $\theta = \pm\pi/2$ una catenoide y $\theta = 0$ un helicoides zurdo.

FIGURA 4.2: Transformación de un catenoide en un helicoid variando θ .

Para el código de la transformación anterior, se hizo fundamentalmente con esta función de *Manim*.

```

1  s = ValueTracker(0)
2
3  catenoide = always_redraw(
4      lambda: Surface(
5          lambda u, v: axes.c2p(
6              np.cos(s.get_value()) * np.sinh(v) * np.sin(u) + np.sin(s.get_value()) * np.cosh(v) * np.cos(u),
7              - np.cos(s.get_value()) * np.sinh(v) * np.cos(u) + np.sin(s.get_value()) * np.cosh(v) * np.sin(u),
8              u * np.cos(s.get_value()) + v * np.sin(s.get_value())
9          ),
10     u_range=[-PI, PI],
11     v_range=[-1, 1],
12     resolution=16,
13     fill_opacity = 0.5
14 ))

```

Aquí s es el parámetro θ que variamos.

Proposición 4.1.10. Sean S_1, S_2 dos superficies regulares. Si existen cartas coordenadas $\mathbf{x} : U \rightarrow S_1$ y $\bar{\mathbf{x}} : U \rightarrow S_2$ tal que $E = \bar{E}$, $F = \bar{F}$, y $G = \bar{G}$, entonces $\phi = \bar{\mathbf{x}} \circ \mathbf{x}^{-1} : \mathbf{x}(U) \subset S_1 \rightarrow S_2$ es una isometría local.

Todas aquellas propiedades de una superficie regular que se definen en términos de la primera forma fundamental son llamadas propiedades intrínsecas de la superficie.

En consecuencia, dos superficies isométricas tienen las mismas propiedades intrínsecas, y las propiedades intrínsecas se preservan bajo isometrías.

Vamos a probar que la curvatura gaussiana es una propiedad intrínseca. Específicamente, vamos a mostrar que localmente se puede expresar en términos de los coeficientes de la primera forma fundamental y de sus derivadas.

En cualquier punto de una superficie regular $S \subset \mathbb{R}^3$ y una elección de coordenadas existe una base (no necesariamente ortogonal) para $\mathbb{R}^3$ desde el punto dado localmente por $\{\mathbf{x}_u, \mathbf{x}_v, \mathbf{N}\}$.

Consideremos las derivadas con respecto a u y v de este triedro:

$$\mathbf{x}_{uu} = \Gamma_{11}^1 \mathbf{x}_u + \Gamma_{11}^2 \mathbf{x}_v + l\mathbf{N}, \quad \mathbf{N}_u = a_{11}\mathbf{x}_u + a_{12}\mathbf{x}_v, \quad (4.1)$$

$$\mathbf{x}_{uv} = \Gamma_{12}^1 \mathbf{x}_u + \Gamma_{12}^2 \mathbf{x}_v + m\mathbf{N}, \quad \mathbf{N}_v = a_{21}\mathbf{x}_u + a_{22}\mathbf{x}_v, \quad (4.2)$$

$$\mathbf{x}_{vv} = \Gamma_{22}^1 \mathbf{x}_u + \Gamma_{22}^2 \mathbf{x}_v + n\mathbf{N}. \quad (4.3)$$

Las funciones Γ_{ij}^k , definidas por estas ecuaciones, son llamadas *símbolos de Christoffel* de S asociados a la parametrización. Los índices i, j y k varía sobre 1 y 2, donde 1 corresponde a la variable u y 2 a la variable v . Los subíndices ij refieren a la segunda parcial de la carta coordenada $\mathbf{x}$, y el supraíndice k al componente en la base ordenada $\{\mathbf{x}_u, \mathbf{x}_v, \mathbf{N}\}$.

Como $\langle \mathbf{x}_u, \mathbf{N} \rangle = \langle \mathbf{x}_v, \mathbf{N} \rangle = 0$, se tiene que $l = \langle m\mathbf{N}, \mathbf{N} \rangle = \langle \mathbf{x}_{uu}, \mathbf{N} \rangle = e$, una de las funciones componente asociadas a las segunda forma fundamental. De manera análoga, $m = f$ y $n = g$. Utilizando la definición de los coeficientes E, F y G , se obtiene que

$$E_u = 2\langle \mathbf{x}_{uu}, \mathbf{x}_u \rangle, \quad E_v = 2\langle \mathbf{x}_{uv}, \mathbf{x}_u \rangle \quad (4.4)$$

$$F_u = \langle \mathbf{x}_{uu}, \mathbf{x}_v \rangle + \langle \mathbf{x}_u, \mathbf{x}_{uv} \rangle, \quad F_v = \langle \mathbf{x}_{uv}, \mathbf{x}_v \rangle + \langle \mathbf{x}_u, \mathbf{x}_{vv} \rangle, \quad (4.5)$$

$$G_u = 2\langle \mathbf{x}_{uv}, \mathbf{x}_v \rangle, \quad G_v = 2\langle \mathbf{x}_{vv}, \mathbf{x}_v \rangle \quad (4.6)$$

Combinando las relaciones (4.1) y (4.4), se derivan las siguientes ecuaciones

$$\begin{aligned} \begin{cases} \Gamma_{11}^1 E + \Gamma_{11}^2 F &= \frac{1}{2} E_u \\ \Gamma_{11}^1 F + \Gamma_{11}^2 G &= F_u - \frac{1}{2} E_v \end{cases} \implies \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} \Gamma_{11}^1 \\ \Gamma_{11}^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} E_u \\ F_u - \frac{1}{2} E_v \end{pmatrix} \\ \\ \begin{cases} \Gamma_{12}^1 E + \Gamma_{12}^2 F &= \frac{1}{2} E_v \\ \Gamma_{12}^1 F + \Gamma_{12}^2 G &= \frac{1}{2} G_u \end{cases} \implies \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} \Gamma_{12}^1 \\ \Gamma_{12}^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} E_v \\ \frac{1}{2} G_u \end{pmatrix} \\ \\ \begin{cases} \Gamma_{22}^1 E + \Gamma_{22}^2 F &= F_v - \frac{1}{2} G_u \\ \Gamma_{22}^1 F + \Gamma_{22}^2 G &= \frac{1}{2} G_v \end{cases} \implies \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} \Gamma_{22}^1 \\ \Gamma_{22}^2 \end{pmatrix} = \begin{pmatrix} F_v - \frac{1}{2} G_u \\ \frac{1}{2} G_v \end{pmatrix} \end{aligned}$$

Como la matriz de los componentes de la métrica es invertible, los símbolos de Christoffel, Γ_{ij}^k , son expresables en términos de las funciones E , F y G y sus derivadas. En superficies isométricas, los símbolos de Christoffel concuerdan. Además, cualquier expresión dada en términos de las funciones componentes de la métrica, los símbolos de Christoffel, y sus derivadas que sea independiente de la elección de coordenadas es intrínseca.

Teorema 4.1.11. *La curvatura Gaussiana es una cantidad intrínseca.*

Demostración. Notemos que

$$(\mathbf{x}_{uu})_v - (\mathbf{x}_{uv})_u = 0, \quad (1)$$

De las relaciones en (4.1), y la independencia lineal de $\mathbf{x}_u$, $\mathbf{x}_v$ y $\mathbf{N}$, se obtiene:

$$(\Gamma_{11}^1)_v + \Gamma_{11}^1 \Gamma_{12}^1 + \Gamma_{11}^2 \Gamma_{22}^1 + e a_{21} - (\Gamma_{12}^1)_u - \Gamma_{12}^1 \Gamma_{11}^1 - \Gamma_{12}^2 \Gamma_{12}^1 - f a_{11} = 0 \quad (4.7)$$

$$(\Gamma_{11}^2)_v + \Gamma_{11}^1 \Gamma_{12}^2 + \Gamma_{11}^2 \Gamma_{22}^2 + e a_{22} - (\Gamma_{12}^2)_u - \Gamma_{12}^1 \Gamma_{11}^2 - \Gamma_{12}^2 \Gamma_{12}^2 - f a_{12} = 0 \quad (4.8)$$

$$\Gamma_{11}^1 f + \Gamma_{11}^2 g + e_v + \Gamma_{12}^1 e - \Gamma_{12}^2 f + f_u = 0 \quad (4.9)$$

De la ecuación (3.19), se tiene que

$$a_{21} = \frac{eF - fE}{EG - F^2}, \quad a_{22} = \frac{fF - gE}{EG - F^2}.$$

De la ecuación (4.7), se sigue que

$$\begin{aligned}
 (\Gamma_{11}^2)_v - (\Gamma_{12}^2)_u + \Gamma_{11}^1 \Gamma_{12}^2 + \Gamma_{11}^2 \Gamma_{22}^2 - \Gamma_{12}^1 \Gamma_{11}^2 - \Gamma_{12}^2 \Gamma_{12}^2 &= f a_{12} - e a_{22} \\
 &= f \frac{eF - fE}{EG - F^2} - e \frac{fF - gE}{EG - F^2} \\
 &= \frac{f^2 E - egE}{EG - F^2} = EK
 \end{aligned}$$

Lo anterior demuestra que la curvatura Gaussiana K se puede expresar en términos de las componentes del tensor métrico, los símbolos de Christoffel y sus derivadas. Lo que muestra que K es una cantidad intrínseca. ■

4.2. Curvatura geodésica y curvas geodésicas

Definiremos a las curvas geodésicas, que son las trayectorias que menos se curvan en una superficie S ; éstas derivan de la curvatura geodésica, una manera intrínseca de medir la curvatura de una superficie S .

Curvatura geodésica para curvas en superficies regulares

Supongamos que $S \subset \mathbb{R}^3$ es una superficie regular y $\alpha : I \subset \mathbb{R} \rightarrow S$ una curva suave. En general, no es posible que $\alpha'' \equiv 0$, ya que esto implicaría que S contiene un segmento de recta en $\mathbb{R}^3$. Por otro lado, como α está en S , la componente tangencial a S de α'' puede considerarse como la responsable del «movimiento» que describe la curva α a lo largo de la superficie S .

Por simplicidad, vamos a suponer que la curva $\alpha : I \rightarrow S$ es de rapidez unitaria. Para cada $s \in I$, el vector tangente $T(s) = \dot{\alpha}(s)$ es tangente a S en el punto $\alpha(s)$. Por tanto, $\{T(s), \mathbf{N}(s) \times T(s), \mathbf{N}(s)\}$ es una base ortonormal de $\mathbb{R}^3$ mientras que el conjunto $\{T(s), \mathbf{N}(s) \times T(s)\}$ generan al plano tangente $T_p S$ en $p = \alpha(s)$.

Como $\alpha(s)$ es de rapidez unitaria, $\ddot{\alpha}(s) \cdot T(s) = 0$. En consecuencia,

$$\ddot{\alpha} = (\ddot{\alpha} \cdot \mathbf{N}(s) \times T(s)) \mathbf{N}(s) \times T(s) + (\ddot{\alpha} \cdot \mathbf{N}(s)) \mathbf{N}(s) \quad (4.10)$$

La magnitud de la componente normal de $\ddot{\alpha}$ es la curvatura normal $\kappa_n = \ddot{\alpha} \cdot \mathbf{N}(s)$. En consecuencia, la magnitud de la componente tangente de $\ddot{\alpha}$ es llamada la curvatura

geodésica y es denotada por κ_g ,

$$\kappa_g = \dot{\alpha} \cdot \ddot{\alpha} \times \mathbf{N}. \quad (4.11)$$

Por otro lado, podemos descomponer la aceleración en su componente tangencial y normal

$$\alpha''_{\text{tan}} = \kappa_g \mathbf{N} \times T \quad y \quad \alpha''_{\text{normal}} = (\alpha'' \cdot \mathbf{N}) \mathbf{N}.$$

Recordemos que $\kappa(s) = \|\ddot{\alpha}(s)\|$ es la curvatura de α . De la ecuación (4.11), tenemos

$$\kappa^2 = \kappa_g^2 + \kappa_n^2.$$

Para una curva $\alpha(t)$ en una superficie S , con una parametrización arbitraria, es posible obtener una expresión que permita calcular la curvatura geodésica de su parametrización por longitud de arco en términos de la parametrización dada.

Lema 4.2.1. *Sea $\alpha : I \rightarrow S$ una curva suave contenida en una superficie regular, no necesariamente parametrizada por longitud de arco. La curvatura geodésica, $\kappa_{g(t)}$, en el punto $\alpha(t)$ se obtiene por*

$$\kappa_g(t) = \frac{\alpha'(t) \cdot \alpha''(t) \times \mathbf{N}(t)}{\|\alpha'(t)\|^3}. \quad (4.12)$$

Demostración. Sea $h : J \rightarrow I$ la inversa de la función longitud de arco. Recordemos que la reparametrización $\alpha(s) := \alpha(h(s))$ es una curva de rapidez unitaria y $\frac{dh}{ds} = \frac{1}{\|\alpha'(h(s))\|}$. Por la regla de la cadena, tenemos

$$\dot{\alpha}(s) = \frac{\alpha'(h(s))}{\|\alpha'(h(s))\|}, \quad (4.13)$$

$$\ddot{\alpha}(s) = \frac{\alpha''(h(s))}{\|\alpha'(h(s))\|^2} - \frac{\langle \alpha'(h(s)), \alpha''(h(s)) \rangle}{\|\alpha'(h(s))\|^3} \cdot \alpha'(h(s)) \quad (4.14)$$

Tomando en cuenta que $t = h(s)$, sustituyendo (4.13) y (4.14) en la fórmula (4.11) obtenemos el resultado deseado. ■

Ejemplo 4.2.2. *Curvatura geodésica para curvas en el toro. Tomemos un toro en $\mathbb{R}^3$ con parametrización:*

$$\mathbf{x}(u, v) = ((R + r \cos u) \cos v, (R + r \cos u) \sin v, r \sin u). \quad (4.15)$$

Asociado a esta parametrización, un vector normal al toro está dado por:

$$\mathbf{N}(u, v) = (-\cos u \cos v, -\cos u \sin v, -\sin u).$$

Calculemos la curvatura geodésica para el paralelo superior del toro, es decir, la curva en el toro definida por $u = \pi/2$. Una parametrización de esta curva es

$$\alpha(t) = (R \cos t, R \sin t, r).$$

Entonces,

$$\begin{aligned}\alpha'(t) &= (-R \sin t, R \cos t, 0) \quad \|\alpha'(t)\| = R, \\ \alpha''(t) &= (-R \cos t, -R \sin t, 0), \\ \mathbf{N}(t) &= (0, 0, -1).\end{aligned}$$

Utilizando la fórmula (4.12), obtenemos

$$\kappa_g(t) = -\frac{1}{R}.$$

Utilizando la relación de Euler, podemos concluir que la curvatura geodésica de los meridianos centrales en el toro ($u = 0$ y $u = \pi$) es idénticamente cero.

Ahora, consideremos la siguiente familia de curvas en el toro

$$\alpha(t) = ((R + r \cos(\omega_1 t)) \cos \omega_2 t, (R + r \cos(\omega_1 t)) \sin \omega_2 t, r \sin \omega_2 t),$$

con $\omega_1, \omega_2 \in \mathbb{R}$. Esta familia de curvas exhibe los siguientes comportamientos:

- i) Si $\frac{\omega_1}{\omega_2} \in \mathbb{Q}$, la curva $\alpha(t)$ es cerrada.
- ii) Si $\frac{\omega_1}{\omega_2} \notin \mathbb{Q}$, la curva $\alpha(t)$ es densa en el toro.

Para esta familia de curvas, su curvatura geodésica es

$$\begin{aligned}\kappa_g(t) &= \frac{-\omega_2 (R + r \cos \psi_1) \sin \psi_1 (\omega_2^2 (R + r \cos \psi_1) + 2r\omega_1^2 \cos \psi_1)}{(r^2\omega_1^2 + \omega_2^2 (R + r \cos (\omega_1 t + \theta_2))^2)^{3/2}} \\ &\quad - \frac{2r\omega_1 (\omega_1\omega_2 \sin \psi_1 \cos 2\psi_2 + (\omega_1^2 + \omega_2^2) r \cos \psi_1 \sin \psi_1 \cos \psi_2 + R\omega_2 \sin \psi_2 \cos \psi_2)}{(r^2\omega_1^2 + \omega_2^2 (R + r \cos (\omega_1 t + \theta_2))^2)^{3/2}},\end{aligned}$$

con $\psi_1 = \omega_1 t + \theta_1$ y $\psi_2 = \omega_2 t + \theta_2$.

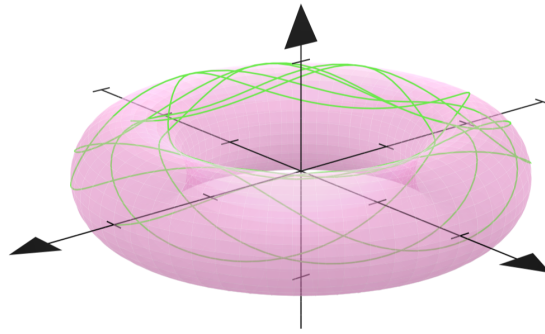


FIGURA 4.3: Una curva de la anterior familia sobre el toro.

La imagen anterior se realizó con este código.

```

1  R_toro = 1.5
2  r_toro = 0.5
3
4  def Christoffel_toro(R=R_toro, r=r_toro):
5      Gamma1_uv = -r * sy.sin(v) / (R + r * sy.cos(v))
6      Gamma2_uu = -(R + r * sy.cos(v)) * sy.cos(v) / r
7
8      Gamma1 = sy.Matrix([[0, Gamma1_uv],
9                          [Gamma1_uv, 0]])
10
11     Gamma2 = sy.Matrix([[Gamma2_uu, 0],
12                         [0, 0]])
13
14     return [Gamma1, Gamma2]
15
16 def f(y, s, C, u_sym, v_sym):
17     y0 = y[0]
18     y1 = y[1]
19     y2 = y[2]
20     y3 = y[3]
21     dy = np.zeros_like(y)
22     dy[0] = y1
23     dy[2] = y3
24
25     Gamma1 = C[0].subs({u_sym: y0, v_sym: y2})
26     Gamma2 = C[1].subs({u_sym: y0, v_sym: y2})
27
28     dy[1] = -(Gamma1[0,0] * y1**2 + 2 * Gamma1[0,1] * y1 * y3 + Gamma1[1,1] * y3**2)
29     dy[3] = -(Gamma2[0,0] * y1**2 + 2 * Gamma2[0,1] * y1 * y3 + Gamma2[1,1] * y3**2)
30
31     return dy.astype(float)
32
33 def solve_geodesic(C, u0, s0, s1, ds):
34     s = np.arange(s0, s1 + ds, ds)
35     return sc.odeint(f, u0, s, args=(C, u, v))

```

Definición 4.2.3. Sea $S \subset \mathbb{R}^3$ una superficie regular. Una curva suave $\alpha : I \rightarrow S$ se dice ser una pre-geodésica si $\kappa_g \equiv 0$. Una curva geodésica es una pre-geodésica parametrizada con rapidez constante.

En el ejemplo 4.2.2, se mostró que los paralelos medios del toro, es decir, las curvas en la parametrización del toro (4.15), que se obtienen al tomar $u = 0$ ó $u = \pi$, son curvas pre-geodésicas en el toro.

Observación 4.2.4. Una curva geodésica en una superficie regular S puede interpretarse como la descripción del movimiento de una partícula en la superficie S con aceleración tangencial cero.

Lema 4.2.5. Sea α una curva de rapidez unitaria en una superficie $S \in \mathbb{R}^3$. Supongamos que α tiene velocidad unitaria. La curvatura geodésica de α está dada por

$$\kappa_g = \kappa \cos \theta,$$

donde κ es la curvatura de la curva α y θ el ángulo entre los vectores $T \times \mathbf{N}$ y $\ddot{\alpha}$.

Demostración. De la descomposición (4.10) para $\ddot{\alpha}$, tenemos

$$\ddot{\alpha} = \kappa_g \mathbf{N} \times T + \kappa_n \mathbf{N}.$$

Tomando el producto punto de $\ddot{\alpha}$ y $\mathbf{N} \times T$, se tiene

$$\kappa_g = \ddot{\alpha} \cdot \mathbf{N} \times T = \|\ddot{\alpha}\| \cos \theta = \kappa \cos \theta,$$

donde θ es el ángulo entre los vectores $\ddot{\alpha}$ y $\mathbf{N} \times T$. ■

Proposición 4.2.6. La curvatura geodésica depende sólo de la métrica. De hecho, tomando $F = 0$ para conveniencia, la curvatura geodésica está dada por la fórmula

$$\kappa_g = \sqrt{EG} \left(-\frac{E_v}{2G} u'^3 + \left(\frac{G_u}{G} - \frac{E_u}{2E} \right) u'^2 v' + \left(\frac{G_v}{2G} - \frac{E_v}{E} \right) u' v'^2 + \frac{G_u}{2E} v'^3 + u' v'' - u'' v' \right).$$

Demostración. Por definición, tenemos $\kappa_g = \ddot{\alpha} \cdot \ddot{\alpha} \times \mathbf{N}$. Sea $\mathbf{x}(u, v)$ una parametrización ortogonal ($F = 0$) tal que $\alpha(s) = \mathbf{x}(u(s), v(s))$. Entonces:

$$\dot{\alpha} = \dot{u} \mathbf{x}_u + \dot{v} \mathbf{x}_v,$$

$$\ddot{\alpha} = \mathbf{x}_{uu}(\dot{u})^2 + \mathbf{x}_{uv}\dot{u}\dot{v} + \mathbf{x}_u\ddot{u} + \mathbf{x}_{vu}\dot{u}\dot{v} + \mathbf{x}_{vv}(\dot{v})^2 + \mathbf{x}_v\ddot{v}.$$

Como $F = 0$, $\{\frac{\mathbf{x}_u}{\sqrt{E}}, \frac{\mathbf{x}_v}{\sqrt{G}}, N\}$ forman una base ortonormal. Por tanto,

$$\mathbf{N} \times \dot{\alpha} = \left(\frac{\mathbf{x}_u}{\sqrt{E}} \times \frac{\mathbf{x}_v}{\sqrt{G}} \right) \times (\dot{u}\mathbf{x}_u + \dot{v}\mathbf{x}_v) = \left(\frac{\sqrt{E}}{\sqrt{G}}\mathbf{x}_v\dot{u} - \frac{\sqrt{G}}{\sqrt{E}}\mathbf{x}_u\dot{v} \right).$$

De aquí, se tiene

$$\mathbf{x}_{uu} \cdot \mathbf{N} \times \dot{\alpha} = \frac{\sqrt{E}}{\sqrt{G}} \left(\frac{-E_v}{2} \right) \dot{u} - \frac{\sqrt{G}}{\sqrt{E}} \left(\frac{E_u}{2} \right) \dot{v},$$

$$\mathbf{x}_{uv} \cdot \mathbf{N} \times \dot{\alpha} = \frac{\sqrt{E}}{\sqrt{G}} \left(\frac{G_u}{2} \right) \dot{u} - \frac{\sqrt{G}}{\sqrt{E}} \left(\frac{E_v}{2} \right) \dot{v},$$

$$\mathbf{x}_{vv} \cdot \mathbf{N} \times \dot{\alpha} = \frac{\sqrt{E}}{\sqrt{G}} \left(\frac{G_v}{2} \right) \dot{u} - \frac{\sqrt{G}}{\sqrt{E}} \left(\frac{-G_u}{2} \right) \dot{v},$$

$$\mathbf{x}_u \cdot \mathbf{N} \times \dot{\alpha} = -\sqrt{EG} \dot{u},$$

$$\mathbf{x}_v \cdot \mathbf{N} \times \dot{\alpha} = \sqrt{EG} \dot{v}.$$

Sustituyendo en la fórmula $\kappa_g = \ddot{\alpha} \cdot \ddot{\alpha} \times \mathbf{N}$, y simplificando, se obtiene el resultado. ■

Teorema 4.2.7. Sea $\alpha(t)$ una geodésica en una superficie M y sea $I : M \rightarrow \mathbf{N}$ una isometría. Entonces, la curva $\beta(t) = I(\alpha(t))$ es una geodésica de $\mathbf{N}$.

Ejemplo 4.2.8 (Geodésicas en la esfera). Sea S_R^2 sea una R -esfera con parametrización $\mathbf{x}(u, v) = (R \cos u \cos v, R \sin u \cos v, R \sin v)$. Aquí vemos que la parametrización da una superficie de revolución con $g(v) = R \sin v$ y $h(v) = R \cos v$, así que las v -curvas son geodésicas. Esto es, las longitudes desde el polo norte hasta el polo sur son geodésicas. Dado que la esfera es simétrica alrededor de su centro, por una rotación apropiada podemos considerarla como una superficie de revolución alrededor de cualquier línea que pase por el centro. Las v -curvas son entonces círculos en la esfera que conectan los puntos de intersección de la línea con la esfera. Como estos puntos corresponden a los polos norte y sur bajo la rotación, los círculos conectándolos son círculos máximos en S_R^2 .

Recordemos que un círculo máximo es un círculo de radio R en una R -esfera y, por lo tanto, tiene como centro el centro de la esfera. Esta discusión muestra que todos los círculos máximos en S_R^2 son geodésicas. Para ver que todos los círculos máximos son geodésicas en S_R^2 , supongamos que α es una geodésica con velocidad unitaria. Luego, $\alpha''_{\tan} = \alpha'' - (\alpha'' \cdot \mathbf{N})\mathbf{N} = 0$, por definición, así que $\alpha'' = (\alpha'' \cdot \mathbf{N})\mathbf{N}$. Pero, en S_R^2 , sabemos que $\mathbf{N}(\alpha(t)) = \alpha(t)/R$, por lo que $\alpha'' = \frac{1}{R^2}(\alpha'' \cdot \alpha)\alpha$. Luego entonces,

$$(\alpha' \times \mathbf{N})' = \frac{1}{R}(\alpha' \times \alpha)' = \frac{1}{R}(\alpha'' \times \alpha + \alpha' \times \alpha') = 0,$$

dado que α'' es paralelo a α y α' es paralelo a sí mismo. Luego, $\alpha' \times \mathbf{N} \stackrel{\text{def}}{=} n$ es un vector constante con $\alpha \cdot n = 0$, puesto que n es perpendicular a $\mathbf{N} = \alpha/R$. Pero esto quiere decir que α reside en el plano teniendo a n como vector normal. Más aún, ya que $\mathbf{N} = \alpha/R$ está en el plano, desde cualquier $\alpha(t)$ podemos llegar al origen y mantenernos en el plano. Entonces, el plano también pasa por $(0,0,0)$. Como α también está en S_R^2 , está en la intersección de la esfera con un plano pasando a través del origen. Por consiguiente, es parte de un círculo máximo.

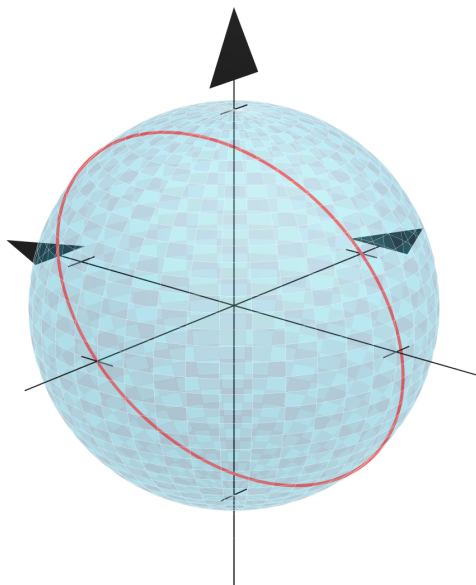


FIGURA 4.4: Una geodésica en la esfera.

De manera similar al caso del toro, este código fue con el que se graficó esta geodésica.

```

1  def Christoffel_esfera():
2      Gamma1 = sy.Matrix([[0, -sy.tan(v)],
3                          [-sy.tan(v), 0]])
4
5      Gamma2 = sy.Matrix([[sy.sin(v)*sy.cos(v), 0],
6                          [0, 0]])
7
8      return [Gamma1, Gamma2]
9
10 def f(y, s, C, u_sym, v_sym):
11     y0 = y[0]
12     y1 = y[1]
13     y2 = y[2]
14     y3 = y[3]
15     dy = np.zeros_like(y)
16     dy[0] = y1
17     dy[2] = y3
18

```

```

19 Gamma1 = C[0].subs({u_sym: y0, v_sym: y2})
20 Gamma2 = C[1].subs({u_sym: y0, v_sym: y2})
21
22 dy[1] = -(Gamma1[0,0] * y1**2 + 2 * Gamma1[0,1] * y1 * y3 + Gamma1[1,1] * y3**2)
23 dy[3] = -(Gamma2[0,0] * y1**2 + 2 * Gamma2[0,1] * y1 * y3 + Gamma2[1,1] * y3**2)
24
25 return dy.astype(float)
26
27 def solve_geodesic(C, u0, s0, s1, ds):
28     s = np.arange(s0, s1 + ds, ds)
29     return sc.odeint(f, u0, s, args=(C, u, v))
30

```

4.3. Ecuaciones geodésicas y la relación de Clairaut

Con el propósito de obtener curvas geodésicas en superficies regulares, vamos a considerar el caso de curvas regulares contenidas en una vecindad coordinada de una parametrización de una superficie regular. Partiendo de esto vamos a derivar el conjunto de ecuaciones geodésicas que se pueden utilizar para obtener curvas geodésicas.

Teorema 4.3.1. *Sea S una superficie regular y $\mathbf{x} : U \subset \mathbb{R}^2 \rightarrow S$ una parametrización de S . Sea $\alpha : I \rightarrow S$ una curva regular suave tal que $\alpha(I) \subset \mathbf{x}(U)$, es decir, para una curva suave $\alpha(t) = \mathbf{x}(u(t), v(t))$ para ciertas funciones suaves $u, v : I \rightarrow U$.*

$\alpha(t)$ es una curva geodésica si y sólo si las funciones $u(t), v(t)$ satisfacen el sistema de ecuaciones diferenciales de segundo orden

$$\begin{aligned}
 u'' + (u')^2 \Gamma_{11}^1 + 2u'v' \Gamma_{12}^1 + (v')^2 \Gamma_{22}^1 &= 0, \\
 v'' + (u')^2 \Gamma_{11}^2 + 2u'v' \Gamma_{12}^2 + (v')^2 \Gamma_{22}^2 &= 0.
 \end{aligned}$$

Más aún, para cada $(u_0, v_0) \in U$, y $a, b \in \mathbb{R}$, existe un $\delta > 0$ y una única curva geodésica $\alpha : (-\delta, \delta) \rightarrow S$ tal que $\alpha(0) = \mathbf{x}(u_0, v_0)$ y $\alpha'(0) = a\mathbf{x}_u(u_0, v_0) + b\mathbf{x}_v(u_0, v_0)$.

Demostración. Por la regla de la cadena, tenemos

$$\alpha'(t) = \mathbf{x}_u u' + \mathbf{x}_v v', \quad y \quad \alpha''(t) = \mathbf{x}_{uu}(u')^2 + \mathbf{x}_{uv} u' v' + \mathbf{x}_u u'' + \mathbf{x}_{uu} u' v' + \mathbf{x}_{vv} (v')^2 + \mathbf{x}_v v''.$$

Tomando en cuenta las relaciones (4.1) para $\mathbf{x}_{uu}$, $\mathbf{x}_{uv}$, $\mathbf{x}_{vv}$, α'' se puede expresar como

$$\begin{aligned}\alpha''(t) &= (u'' + (u')^2\Gamma_{11}^1 + 2u'v'\Gamma_{12}^1 + (v')^2\Gamma_{22}^1)\mathbf{x}_u \\ &\quad + (v'' + (u')^2\Gamma_{11}^2 + 2u'v'\Gamma_{12}^2 + (v')^2\Gamma_{22}^2)\mathbf{x}_v \\ &\quad + ((u')^2e + 2u'v'f + (v')^2g)\mathbf{N}.\end{aligned}$$

Como las componentes $\mathbf{x}_u$ y $\mathbf{x}_v$ de α'' representan la componente transversal de α'' , una condición necesaria y suficiente para que α sea una curva geodésica es que los coeficientes de $\mathbf{x}_u$ y $\mathbf{x}_v$ para el vector de α'' se anulen simultáneamente. La segunda parte del teorema es consecuencia del teorema de existencia y unicidad de soluciones de ecuaciones diferenciales ordinarias. ■

Corolario 4.3.2. *Si $\mathbf{x}(u, v)$ es una parametrización ortogonal ($F = 0$), las ecuaciones geodésicas se vuelven*

$$\begin{aligned}u'' + \frac{E_u}{2E}u'^2 + \frac{E_v}{E}u'v' - \frac{G_u}{2E}v'^2 &= 0, \\ v'' + \frac{E_v}{2G}u'^2 + \frac{G_u}{G}u'v' - \frac{G_v}{2G}v'^2 &= 0,\end{aligned}$$

donde $E = \langle \mathbf{x}_u, \mathbf{x}_u \rangle$ y $G = \langle \mathbf{x}_v, \mathbf{x}_v \rangle$.

Proposición 4.3.3 (Relación de Clairaut). *En una superficie de revolución, una geodésica $\alpha(t)$ hace un ángulo $\psi(t)$ con los meridianos para los cuales $\sin \psi(t)$ veces el radio de la superficie en $\alpha(t)$ es constante.*

Proposición 4.3.4. *Decimos que una parametrización ortogonal $\mathbf{x}(u, v)$ es una parametrización de Clairaut en u si $E_v = 0$ y $G_v = 0$. La parametrización en v si $E_u = 0$ y $G_u = 0$. Las ecuaciones geodésicas se simplifican en estos casos a*

$$\begin{aligned}u'' + \frac{E_u}{2E}u'^2 - \frac{G_u}{2E}v'^2 &= 0 \\ v'' + \frac{G_u}{G}u'v' &= 0 \\ u'' + \frac{E_v}{E}u'v' &= 0 \\ v'' - \frac{E_v}{2G}u'^2 + \frac{G_v}{2G}v'^2 &= 0\end{aligned}$$

A continuación un ejemplo acerca de geodésicas en diferentes superficies. Se hizo un código que resuelve las ecuaciones geodésicas y genera una animación.

Ejemplo 4.3.5 (Bruja de Agnesi). *La siguiente figura muestra una geodésica sobre la superficie de revolución obtenida de la bruja de Agnesi.*

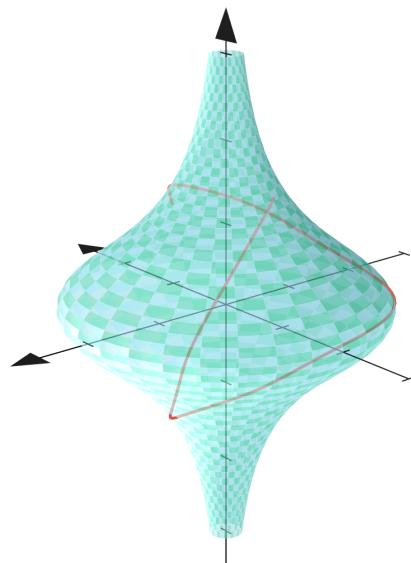


FIGURA 4.5: Una geodésica en la bruja de Agnesi.

Este es el código que nos resuelve una geodésica sobre la bruja de Agnesi.

```

1  u, v = sy.symbols('u v')
2
3  a_agnesi = 1.0
4
5  def Christoffel_agnesi(a=a_agnesi):
6      f = 2 * a / (1 + u**2)
7      f_prime = sy.diff(f, u) # -4a*u / (1 + u**2)**2
8      g_prime = 1 # 1
9
10     E = sy.simplify(f_prime**2 + g_prime**2)
11     G = f**2
12
13     E_u = sy.diff(E, u)
14     G_u = sy.diff(G, u)
15
16     Gamma1_uu = sy.Rational(1, 2) * (1 / E) * E_u
17     Gamma1_vv = sy.Rational(-1, 2) * (1 / E) * G_u
18
19     Gamma2_uv = sy.Rational(1, 2) * (1 / G) * G_u
20
21     Gamma1 = sy.Matrix([
22         [Gamma1_uu, 0],
23         [0, Gamma1_vv]
24     ])
25
26     Gamma2 = sy.Matrix([
27         [0, Gamma2_uv],

```

```

28         [Gamma2_uv, 0]
29     ])
30
31     return [Gamma1, Gamma2]
32
33 def f(y, s, C, u_sym, v_sym):
34
35     # Sistema de 4 EDOs de primer orden.
36     y0, y1, y2, y3 = y
37     dy = np.zeros_like(y)
38     dy[0] = y1 # u'
39     dy[2] = y3 # v'
40
41     # Sustitución de variables simbólicas
42     Gamma1 = C[0].subs({u_sym: y0, v_sym: y2})
43     Gamma2 = C[1].subs({u_sym: y0, v_sym: y2})
44
45     # Ecuaciones geodésicas (u'' = dy[1], v'' = dy[3])
46     dy[1] = -(Gamma1[0,0] * y1**2 + 2 * Gamma1[0,1] * y1 * y3 + Gamma1[1,1] * y3**2)
47     dy[3] = -(Gamma2[0,0] * y1**2 + 2 * Gamma2[0,1] * y1 * y3 + Gamma2[1,1] * y3**2)
48
49     return dy.astype(float)
50
51 def solve_geodesic(C, u0, s0, s1, ds):
52     s = np.arange(s0, s1 + ds, ds)
53     # u0 = [u(0), u'(0), v(0), v'(0)]
54     return sc.odeint(f, u0, s, args=(C, u, v))

```

Bibliografía

- [1] Misael Avendaño Camacho. *Teorema Fundamental de Superficies y el Criterio de Frobenius*. Tesis de licenciatura, Universidad de Sonora, Hermosillo, Sonora, 2006. Disponible en <https://lic.mat.uson.mx/web/index.php/tesis/>.
- [2] S. Chongchitnan. *Exploring University Mathematics with Python*. Springer, 2023. ISBN 978-3-031-46269-6.
- [3] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976. ISBN 0-13-212589-7.
- [4] A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. Chapman & Hall/CRC, 2006. ISBN 978-1-58488-448-4.
- [5] C. Imaz and Z. Vorel. *Ecuaciones diferenciales ordinarias*. Editorial Limusa-Wiley, 1968.
- [6] J. McCleary. *Geometry from a Differentiable Viewpoint*. Cambridge University Press, 1994. ISBN 9780521424806.
- [7] J. Oprea. *Differential Geometry and Its Applications*. The Mathematical Association of America, 2007. ISBN 978-0-88385-748-9.
- [8] L. Perko. *Differential Equations and Dynamical Systems*. Springer New York, 1996. ISBN 978-0-38794-778-5.
- [9] Manuel Alexis Amparano Ruiz. *Superficies minimales: definición, propiedades y aplicaciones*. Tesis de licenciatura, Universidad de Sonora, Hermosillo, Sonora, 2006. Disponible en <https://lic.mat.uson.mx/web/index.php/tesis/>.
- [10] M. Spivak. *Cálculo en variedades*. Reverté, 1970. ISBN 978-8-42915-142-8.
- [11] The Manim Community Developers. Manim – Mathematical Animation Framework, November 2025. URL <https://www.manim.community/>.