

Modelos 3D de la colección de figuras de Zoel García de Galdeano



Julia Solana Sancho

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Directores del trabajo:

Luis Rández García

Raquel Villacampa Gutiérrez

14 de septiembre de 2018

Abstract

This project is focused on the study of several aspects of differential geometry, numerical analysis and divulgation taking into account that the main aim is the visualization and printing of mathematical surfaces. In particular some figures located in the faculty of Sciences of the University of Zaragoza from the collection by Zoel Garcia de Galdeano are considered. This collection was donated by Zoel Garcia de Galdeano, a mathematician who was a professor at the University of Zaragoza for 30 years between the late nineteenth and early twentieth centuries. It is a collection of the catalog of geometric models designed in Germany in the second half of the nineteenth century for facilitate the teaching of mathematics and is the only collection that exists in Spain



Figura 1: Exhibition at the Faculty of Sciences of the University of Zaragoza

This project is divided in four chapters. The first chapter is devoted to implementation in interactive PDF files. The two formats allowed by the software acrobat reader are detailed: U3D and PRC. The first can be generated by using the software Meshlab and the second one can by with Asymptote. Both formats allow the interactive manipulation and a correct visualization of the surfaces. The key point in this chapter is the importance between visible and printable surfaces

In the second chapter, the STL (“Standard Triangle Language”) format is described. This format is widely used for printing 3D objects. This format encodes a surface by a process called “tessellation”, that consists in split the surface into triangles called facets and identifies them by its three vertices and a unit normal vector with some restrictions.

Different software to generate surfaces such as Mathematica, OpenSCAD, Sage, MeshLab or Mesh-Mixer are detailed. Sometimes they are used together with others like octave and python. Some important aspects of the software CURA are detailed when printing in 3D, for instance: printing speed, temperature of the heater, orientation of the figure and scaling.

Chapter 3 is dedicated to the differential geometry of surfaces, in particular the fundamental forms and the differential equations that arise in the study of the lines of curvature and geodesics. To solve the

differential equations that appear, an analysis of numerical methods has been introduced, such as the Euler method or explicit Runge-Kutta method.

Finally, in last chapter some applications in other fields of science such as chemistry and possible extensions of work are given.

Índice general

Abstract	III
1. Introducción	1
1.1. Visualización	1
1.1.1. Ficheros U3D	2
1.1.2. Asymptote	4
2. Generación de superficies e impresión	5
2.1. STL	5
2.2. OpenSCAD	7
2.2.1. Introducción	7
2.2.2. Impresión en 3D	10
2.3. Mathematica	11
2.4. Impresión: CURA	12
3. Geometría de superficies	13
3.1. Métodos numéricos	13
3.1.1. Métodos numéricos para problemas de valor inicial	13
3.1.2. Métodos numéricos para problemas de contorno	14
3.2. Geodésicas y líneas de curvatura	15
3.2.1. Líneas de curvatura	15
3.2.2. Geodésicas	19
4. Conclusiones	23
4.1. Aplicaciones	23
4.2. Líneas futuras de trabajo	23
4.3. Agradecimientos	23
Bibliografía	25
APÉNDICE 1	27
APÉNDICE 2	41

Capítulo 1

Introducción

Con el objetivo de mejorar la enseñanza de las matemáticas y encontrar herramientas para su mejor comprensión, se construyeron en la segunda mitad del siglo XIX en Alemania modelos matemáticos para facilitar la visualización de objetos geométricos. Estos modelos representan objetos de geometría diferencial, superficies algebraicas o instrumentos para la física. No se tiene conocimiento de la existencia en España de más colecciones que la conservada en la Facultad de Ciencias de la Universidad de Zaragoza que fue donada por el matemático Zoel García de Galdeano [3].

Galdeano (Pamplona, 1846 - Zaragoza, 1924) fue profesor en la Facultad de Ciencias de Zaragoza durante 30 años y dado su interés en la enseñanza de las matemáticas, adquirió una colección de dichos modelos. Actualmente esta colección consta de 35 modelos de escayola y está expuesta en el hall de la Facultad de Ciencias.

En este trabajo se han elegido algunos de los modelos geométricos de la colección de Galdeano para su impresión 3D. Además, se va a detallar el procedimiento para la generación de superficies, su análisis geométrico y matemático.

1.1. Visualización

Un aspecto importante a la hora de imprimir superficies es su grosor, que dependerá de la geometría de la superficie, de la calidad de impresión... No todas las superficies visualizables son sólidas, aspecto necesario para su impresión. Más tarde se verá una herramienta para dar grosor a una superficie. A continuación se muestra la información proporcionada por un software de impresión ante una superficie imprimible (fig. (1.2)) y otra que no lo es (fig. (1.1)) (los gramos de material plástico empleados son para construir la base).

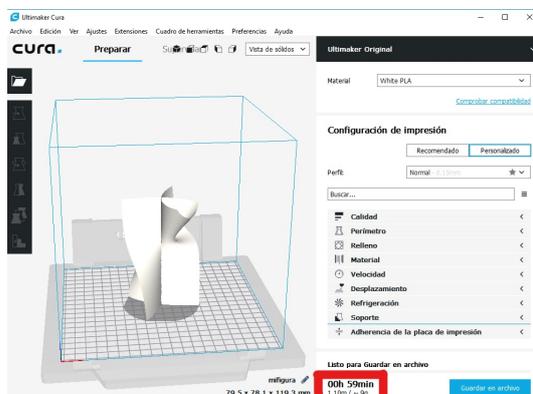


Figura 1.1: Superficie no imprimible

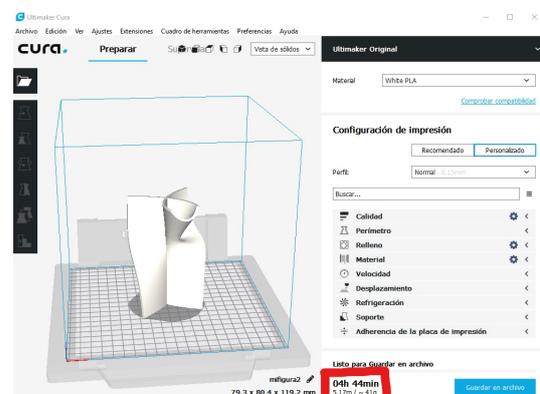


Figura 1.2: Superficie imprimible

Para la implementación de superficies matemáticas interactivas en archivos digitales PDF se detallan los dos formatos soportados (U3D y PRC) por este tipo de archivos y cómo puede realizarse en L^AT_EX.

Una vez generado el archivo PDF se tendrá que leer necesariamente con el software gratuito Acrobat Reader.

1.1.1. Ficheros U3D

Universal 3D (U3D) es un formato de archivo para gráficos 3D por ordenador. Puede ser leído por varios programas de software entre los cuales se encuentra MeshLab, un software libre para editar y procesar mallas triangulares tridimensionales, y que ofrece herramientas para procesar datos generados con dispositivos de digitalización 3D y preparar modelos para impresión 3D (Ver fig. 1.3).

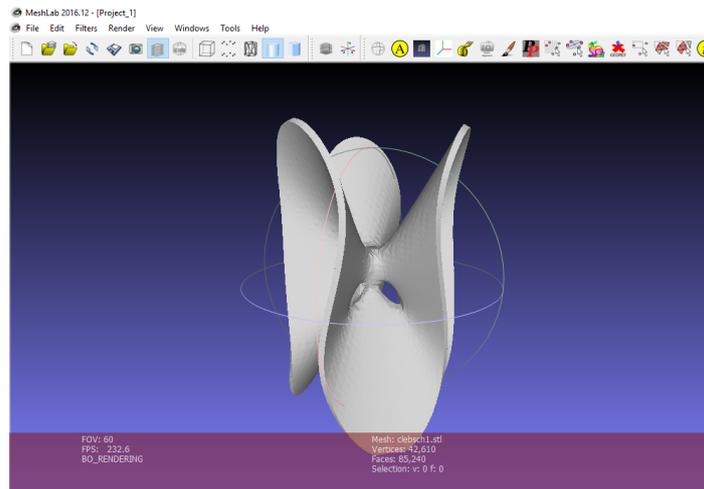


Figura 1.3: Archivo STL visualizado con MeshLab

En la parte inferior de la pantalla se observa la complejidad de la figura dada por el número de vértices y facetas que tiene el archivo STL (ver sección 2.1)

Al exportar en formato U3D con MeshLab se genera un archivo U3D y una plantilla \LaTeX donde incluye ese archivo, veamos un ejemplo:

```
\documentclass[a4paper]{article}
\usepackage[3D]{movie15}
\usepackage{hyperref}
\usepackage[UKenglish]{babel}

\begin{document}
\includemovie[
poster,
toolbar, %same as 'controls'
label=VII20.u3d,
text=(figura.u3d),
3Daac=60.000000, 3Droll=0.000000, 3Dc2c=-0.021850 -17.881401 -0.002079,
3Droo=17.881414, 3Dcoo=-0.021854 -0.968587 0.002079,
3Dlights=CAD,
]{\linewidth}{\linewidth}{figura.u3d}
\end{document}
```

El resultado aparece en la fig. (1.4)

(figura.u3d)

Figura 1.4: Figura de la colección de Galdeano ([7]).

1.1.2. Asymptote

Asymptote es un lenguaje de gráficos vectoriales propio de \LaTeX que proporciona un sistema natural de coordenadas para su uso en dibujo técnico. En particular se tiene interés en su empleo para el dibujo de superficies. Se puede incluir el código de Asymptote en ficheros \LaTeX empleando el entorno `\begin{asy}`, `\end{asy}` y se compila con la secuencia: `pdfLatex + asymptote + pdfLatex`. De esta manera el formato de archivo gráfico que se genera es PRC (Product Representation Compact) desarrollado por Adobe. Hay que notar que es necesario incluir el paquete `\usepackage{asymptote}` en el preámbulo del fichero \LaTeX .

Para el dibujo de superficies dadas en forma implícita ($g(x,y,z) = 0$) existen dos opciones, `contour3` y `smoothcontour3`. `contour3` dibuja la superficie en una malla de tamaño máximo 20×20 en las coordenadas (x,y) mientras que `smoothcontour3` es más precisa ya que usa técnicas adaptivas en la elección de puntos.

Ejemplo con `smoothcontour3`:

```
\begin{asy}
import graph3;
import solids;
size(8cm);
import smoothcontour3;
real f(real x, real y, real z) {return z^2*(z+4)+y*(y^2-3*x^2) ;};
draw(implicitsurface(f, (-4,-4,-4), (4,4,4)),surfacepen=olive );
\end{asy}
```

Ejemplo con `contour3`:

```
\begin{asy}
import graph3;
import contour3;
size(6cm);
real f(real x, real y, real z) {return z^2*(z+4)+y*(y^2-3*x^2);}
draw(surface(contour3(f,(-4,-4,-4),(4,4,4), 20)),blue));
\end{asy}
```

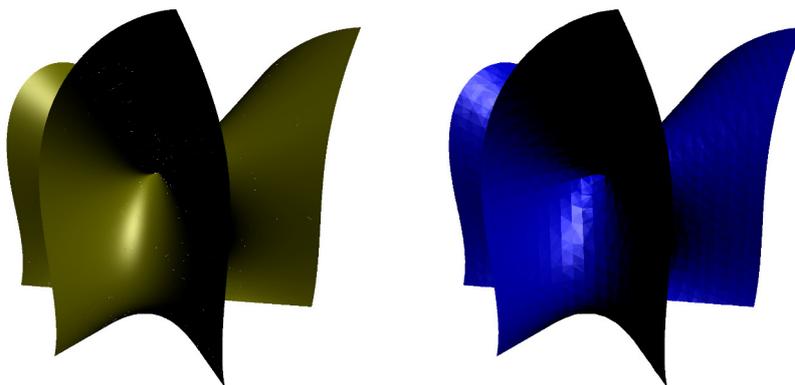


Figura 1.5: `smoothcontour3` (izda.), `contour3` (dcha.)

Capítulo 2

Generación de superficies e impresión

Uno de los objetivos de esta memoria es analizar los aspectos geométricos y matemáticos y de implementación para la impresión de algunas figuras de la colección de Galdeano, para ello se va a detallar el formato de archivo STL que será en el que se exporten las superficies y que el software CURA de impresión lee. Se detallan a continuación algunos programas de software empleados para generar las superficies como OpenSCAD, SAGE, Mathematica, MeshLab o MeshMixer.

2.1. STL

STL es un formato de archivo creado por 3D Systems. El verdadero significado de la extensión de archivo STL parece que se ha perdido pero se han adoptado algunas como “Standard Triangle Language” o “Standard Tessellation Language” [2], [4].

Este formato solo describe la geometría de la superficie de un objeto tridimensional sin detallar su color (aunque hay versiones no estándar de STL que sí transforman información de color), textura u otros atributos del modelo comúnmente representado por un programa de diseño asistido por ordenador (CAD).

Se codifica mediante el concepto de “teselación”. La superficie se divide en triángulos (también llamados facetas) y cada faceta se identifica de manera única mediante tres vértices y un vector normal unitario, especificados por sus coordenadas.

La norma STL tiene algunas reglas especiales para la teselación y almacenamiento de información:

- Los vértices se enumeran en sentido antihorario cuando se mira la superficie desde el exterior.
- La dirección del vector unitario debe ser hacia afuera siguiendo la regla de la mano derecha (Ver fig. 2.1).

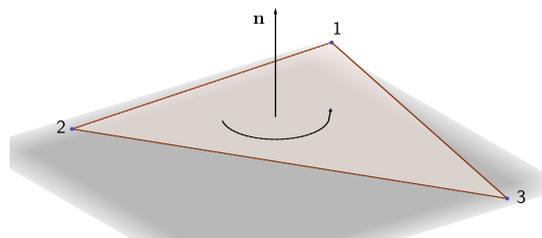


Figura 2.1: Faceta STL

- Todo triángulo debe compartir dos vértices con cada uno de sus triángulos adyacentes. Es decir, un vértice de un triángulo no puede estar en el lado de otro triángulo. (Ver fig. 2.2)

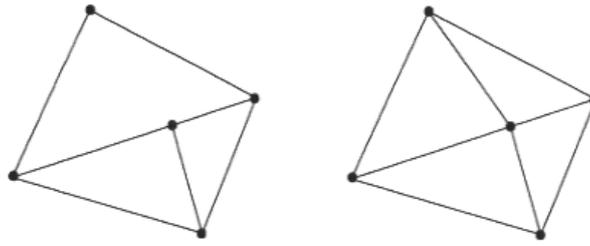


Figura 2.2: Teselación incorrecta / teselación correcta

Hay que notar que los archivos STL no contienen información de escala y las unidades son arbitrarias. La sintaxis del código ASCII STL es la siguiente:

```
solid nombre
facet normal ni nj nk          (coordenadas del vector normal unitario)
  outer loop
    vertex v1x v1y v1z        (coordenadas de cada vértice)
    vertex v2x v2y v2z
    vertex v3x v3y v3z
  endloop
endfacet
facet normal
...
endfacet
.
.
.
endsolid nombre
```

Para cada triángulo se almacena en coma flotante la información de su vector normal y de sus tres vértices.

Se ilustra a modo de ejemplo el código ASCII del archivo STL que representa el cubo $\left[-\frac{1}{2}, \frac{1}{2}\right]^3$.

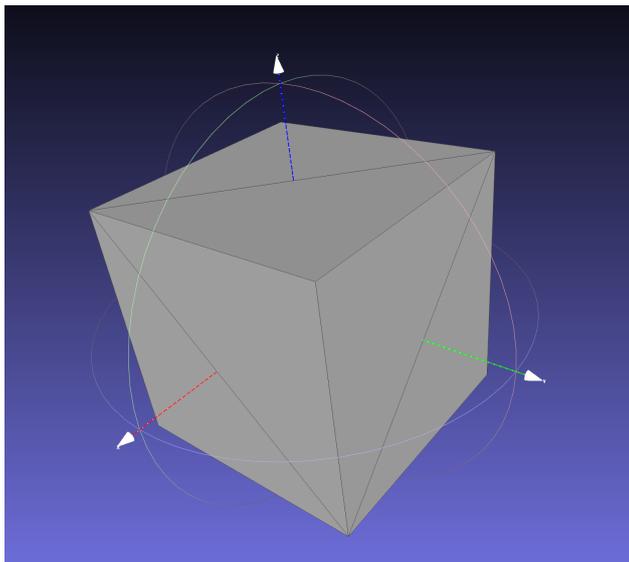
En las líneas 3-16 del cuadro 2.1 están detalladas las dos facetas de la cara superior del cubo, que tienen el mismo vector normal y comparten dos vértices. Observar que los vértices están descritos en sentido antihorario. Notar que el cubo tiene 6 caras que son cuadrados y cada una se divide en dos triángulos siendo así 12 el número total de triángulos o facetas.

Como se ha visto, el formato de archivo STL se aproxima a la superficie del modelo CAD mediante triángulos y si la superficie es curva, la aproximación nunca será perfecta. Evidentemente, al reducir el tamaño de los triángulos se podría lograr una impresión de mayor calidad, pero esto implicaría un archivo STL de mayor tamaño. Por ello es importante encontrar el equilibrio entre calidad de impresión y tamaño del archivo. También es posible emplear la versión binaria de STL para reducir el tamaño del fichero.

```

cube(size=1,center=true);
solid cubo
facet normal 0 0 1
  outer loop
    vertex -0.5 0.5 0.5
    vertex 0.5 -0.5 0.5
    vertex 0.5 0.5 0.5
  endloop
endfacet
facet normal 0 0 1
  outer loop
    vertex 0.5 -0.5 0.5
    vertex -0.5 0.5 0.5
    vertex -0.5 -0.5 0.5
  endloop
endfacet
...10 facetas triangulares más
endsolid cubo

```



Cuadro 2.1: Archivo STL de un cubo

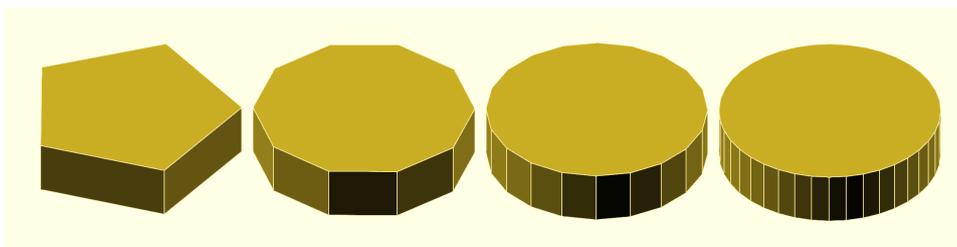
2.2. OpenSCAD

2.2.1. Introducción

OpenSCAD es un software libre para crear modelos de CAD (diseño asistido por ordenador) en 3D que proporciona dos tipos de modelado tridimensional:

- Geometría sólida constructiva (CSG) y extrusión de primitivas de 2D en el espacio ([8]). CSG utiliza primitivas geométricas como el cubo, esfera, cilindro y el poliedro modificándolas y combinándolas con herramientas de unión, intersección, diferencia, rotación...
- En el caso bidimensional las primitivas son rectángulos, circunferencias y polígonos. Pueden realizarse transformaciones lineales y/o de rotación para obtener modelos tridimensionales.

Es importante destacar que la construcción de una circunferencia o cilindro se realiza mediante aproximaciones poligonales, por lo que la resolución dependerá del número de caras (fn) que le asignemos (ver Fig. 2.3)

Figura 2.3: Dibujo de cilindros con valores de $fn=5, 10, 20$ y 40 .

Las superficies de revolución se generan con la extrusión de polígonos. Por ejemplo, para generar

la superficie de rotación de la función seno hiperbólico (Fig. 2.4), pueden calcularse las coordenadas de la curva en python como:

```
import numpy as np
z=[[-np.sinh(x)/12, x] for x in np.arange(0,5.05,0.05)]
print (z)
```

y luego generar la superficie (Fig. 2.4) con la orden de extrusión:

```
rotate_extrude($fn=200) polygon( points=[[-0.0, 0.0], [-0.004168, 0.05],
..., [-5.88199, 4.95], [-6.183600, 5.0]]);
```

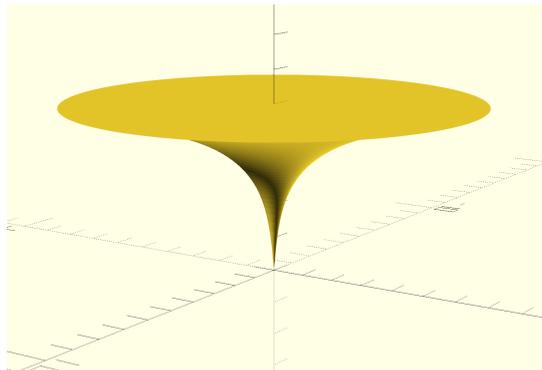


Figura 2.4: Superficie de rotación de la función seno hiperbólico generada por OpenSCAD

OpenSCAD permite el uso de poliedros para la construcción de superficies. Para este fin hay que dar en una lista las coordenadas de los puntos (vértices) del poliedro y en otra la lista de los índices de los vértices que conforman cada faceta, que en este caso son cuadriláteros. Por ejemplo,

```
vertices=[[1.0, 2.0, 3.0], [-1.0, 2.0, 3.0], [-1.0, -2.0, 3.0],
[1.0, -2.0, 3.0], [1.0, 2.0, -3.0], [-1.0, 2.0, -3.0],
[1.0, -2.0, -3.0], [-1.0, -2.0, -3.0]];
facetas=[[0, 1, 2, 3], [0, 4, 5, 1], [0, 3, 6, 4],
[5, 4, 6, 7], [6, 3, 2, 7], [2, 1, 5, 7]];

polyhedron(points=vertices, faces=facetas);
```

los 8 vértices se han numerado del 0 al 7 dando como resultado el sólido poliédrico de la Fig. 2.5.

Hay que notar que las superficies cerradas generadas en las figuras 2.4 y 2.5 son imprimibles.

Con el software SAGE es posible generar una superficie (poliedro) (ver fig. 2.6) y de manera sencilla obtener la lista de los vértices e índices de los puntos que conforman las facetas del poliedro.

```
x, y, z = var('x , y, z')
g(x,y,z) = (x^2+y^2-z^2-2)*(z-(1-y^2/x^2)/(1+y^2/x^2)/2/(1+(x^2+y^2))) -
(1-y^2/x^2)/(1+y^2/x^2)
a=implicit_plot3d(g(x,y,z)==0,(x,-3.5,3.5),(y,-3.5,3.5),(z,-2,2),
plot_points=18, color='orange', mesh=1)
```

Con la orden `puntos=a.vertex_list()` SAGE proporciona una lista de tuplas con las coordenadas de los vértices y para su uso en OpenSCAD se convierte en una lista de listas con la sentencia,

```
[ list(puntos[i]) for i in range(len(puntos)) ]
```

```
[[3.5, -3.5, 0.0], [3.5, -3.088, 0.00902], ..., [-3.5, 3.5, 0.0]]
```

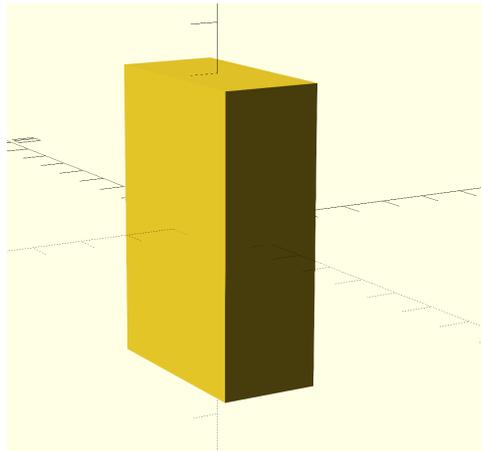


Figura 2.5: Poliedro generado dando la lista de vértices y facetas

Se genera otra lista de listas con los índices de los vértices que conforman cada faceta, con la orden

```
facetas=a.index_faces()
[[0, 1, 2], [3, 4, 5], ..., [6432, 6433, 6434],[6435, 6436, 6437]]
```

siendo 0 el vértice $[3,5,-3,5,0,0]$, 1 el vértice $[3,5,-3,088,0,00902]$... En este caso las facetas son triángulos. Finalmente, copiando lo anterior en un fichero OpenSCAD y con la sentencia

```
polyhedron(points=puntos, faces=facetas);
```

se genera la superficie de la figura (2.7).

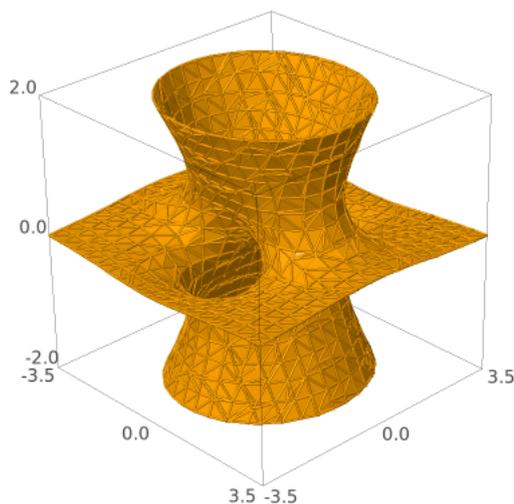


Figura 2.6: Superficie generada por SAGE

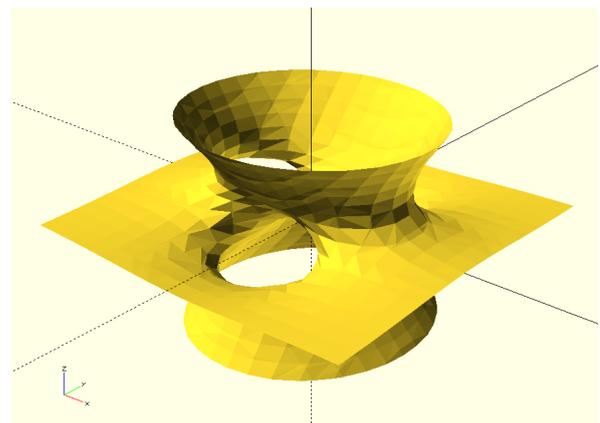


Figura 2.7: Superficie generada por OpenSCAD

Finalmente, hay otra forma de generar superficies explícitas sencillas en OpenSCAD con la ayuda, por ejemplo, de `octave`. Como ejemplo, se define una matriz que corresponde a la función explícita $z = 10 \sin(x) \cos(y)$, $(x,y) \in [1, 10]^2$ y se almacena en un fichero que posteriormente será leído desde OpenSCAD. Las sentencias en `octave` son:

```
octave:1> d = (sin(1:0.2:10)' * cos(1:0.2:10)) * 10;
octave:2> save("super.dat", "d");
```

y la línea en OpenSCAD:

```
surface(file = "super.dat", center = true); \\ OpenSCAD
```

y el resultado se muestra en la Fig. 2.8.

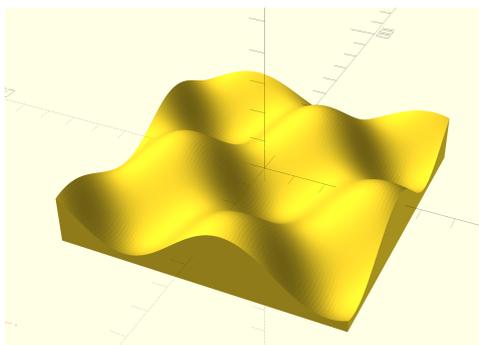


Figura 2.8: Superficie generada por OpenSCAD

2.2.2. Impresión en 3D

OpenSCAD puede exportar de manera directa a formato STL por lo que una de sus mayores utilidades es la impresión en 3D. Es posible modelar algunas figuras de la colección de Galdeano para su posterior impresión. En particular se ha generado la superficie de revolución de la función seno hiperbólico (II_1a). También es posible modelar de manera similar la superficie de rotación de la tractriz. (ver fig. 2.9)

Por otra parte, la superficie generada por SAGE (Fig. 2.6) no tiene grosor luego no es imprimible. Con objeto de añadir anchura se puede guardar en formato STL con la sentencia `a.save('surface.stl')` y con la ayuda del software Meshmixer se la añade grosor en la dirección del vector normal en cada faceta para generar una figura imprimible (Fig. 2.10).

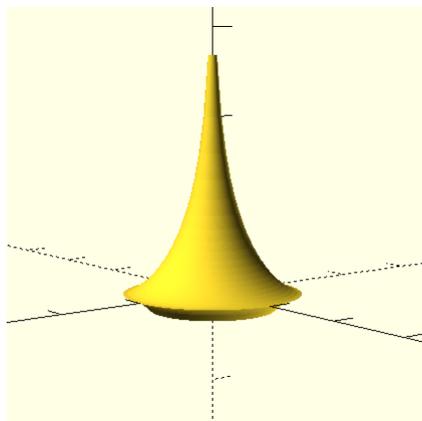


Figura 2.9: Superficie de rotación de la tractriz

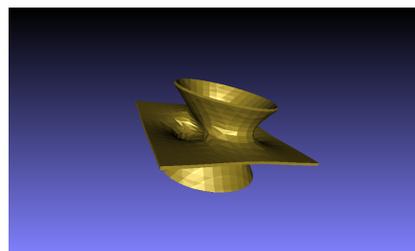


Figura 2.10: Superficie con anchura generada por Meshmixer

2.3. Mathematica

Mathematica es un programa comercial para realizar todo tipo de cálculos simbólicos y numéricos, representaciones gráficas de curvas, superficies... y dotado de un lenguaje de programación de propósito general. En este trabajo se ha utilizado para la generación de superficies, exportarlas a formato STL y resolución numérica de los sistemas diferenciales que aparecen en el cálculo de líneas de curvatura y geodésicas.

El siguiente ejemplo muestra la facilidad de dibujo de una superficie imprimible con la opción `PlotTheme -> ThickSurface` que le da grosor, necesario para su posterior impresión. En las figuras (2.11) y (2.12) se aprecia dicho efecto. El código utilizado es el siguiente:

```
output1=ContourPlot3D[-5 (x^2 y + x^2 z + y^2 x + y^2 z + z^2 y + z^2 x) +
2 (x y + x z + y z) == 0 , {x, -1, 1}, {y, -1, 1}, {z, -1, 1},
RegionFunction -> Function[{x, y, z}, x^2 + y^2 + z^2 < 1],
PlotTheme -> "ThickSurface"]
```

```
Export["cayley.stl",output1]      (exporta a formato STL)
```

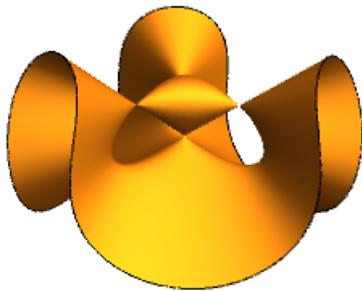


Figura 2.11: Superficie sin ThickSurface

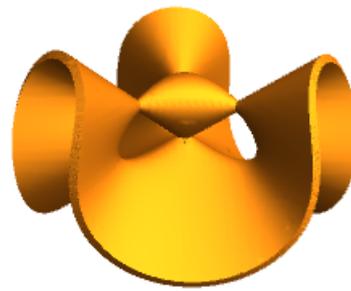


Figura 2.12: Superficie con ThickSurface

Mathematica es capaz de exportar a formato STL con buenos resultados de impresión. En este trabajo se han impreso algunas superficies de la colección de Galdeano generadas con Mathematica como Cayley, VII_18, X_5, VII_16, VII_20 (Ver fig.(2.13)).



Figura 2.13: Superficie colección Galdeano generada con Mathematica

2.4. Impresión: CURA

CURA es el software gratuito que se ha empleado para la preparación de los modelos para su impresión en 3D y para ello hay que tener en cuenta los siguientes aspectos (Ver fig. 2.14):

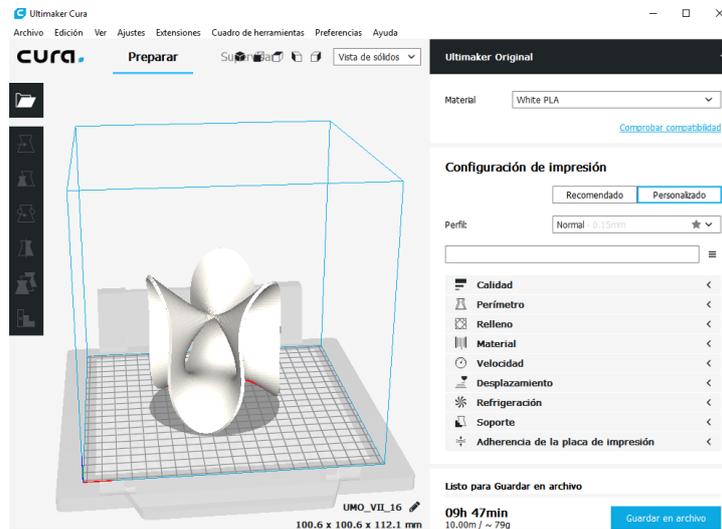


Figura 2.14: Ventana de manipulación en CURA

- Abrir el archivo STL y escalarlo adecuadamente (puesto que como se ha visto, los archivos STL no contienen información de escala).
- Seleccionar el porcentaje de relleno de la superficie (20-25 % es suficiente).
- Elegir la distancia deseada entre una capa y otra (cada capa es una curva de nivel de la superficie). Se suele tomar entre 0,1-0,3mm. Cuanto menor sea más calidad tendrá la superficie impresa.
- Para determinadas superficies puede ser necesario el uso de soportes, que se generan automáticamente por CURA, y en ocasiones es conveniente rotar la figura para facilitar la generación de soportes y su correcta impresión.
- El color del material utilizado es importante puesto que de él depende la temperatura que hay que programar al extrusor. En este trabajo se ha empleado material plástico termofusible PLA (ácido poliláctico) primero de color azul y más tarde blanco, dando mejores resultados de impresión.
- La velocidad de impresión es importante puesto que una impresión muy rápida no daría tiempo a una correcta solidificación del material y una impresión muy lenta podría obturar el extrusor.

Capítulo 3

Geometría de superficies

En este capítulo, se motiva el estudio de la geometría de superficies, sus líneas geodésicas y de curvatura. Puesto que estas líneas son solución de PVI y/o de contorno, será necesario el empleo de métodos numéricos.

3.1. Métodos numéricos

Se van a describir algunos métodos numéricos para resolver los problemas de valor inicial y de contorno que aparecen en el cálculo de líneas de curvatura y geodésicas en superficies paramétricas.

3.1.1. Métodos numéricos para problemas de valor inicial

Sea el problema,

$$\begin{aligned} \mathbf{y}' &= \mathbf{f}(x, \mathbf{y}), & a < x < b, \\ \mathbf{y}(a) &= \alpha \end{aligned} \quad (3.1)$$

con $\mathbf{f}: [a, b] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ y se asume que el PVI anterior tiene solución única. El hecho de tratar ecuaciones diferenciales de primer orden no supone ninguna restricción ya que toda ecuación diferencial de orden superior se puede transformar en la forma (3.1). En efecto, dada una ecuación diferencial escalar de orden m ,

$$u^{(m)} = g(x, u, u', \dots, u^{(m-1)}), \quad a < x < b, \quad (3.2)$$

si $\mathbf{y}(x) = (\mathbf{y}_1(x), \mathbf{y}_2(x), \dots, \mathbf{y}_m(x))^T$ con

$$\begin{aligned} \mathbf{y}_1(x) &= u(x) \\ \mathbf{y}_2(x) &= u'(x) \\ &\vdots \\ \mathbf{y}_m(x) &= u^{(m-1)}(x) \end{aligned}$$

la EDO de orden m (3.2) se transforma en el sistema de primer orden,

$$\begin{aligned} \mathbf{y}'_1(x) &= \mathbf{y}_2(x) \\ \mathbf{y}'_2(x) &= \mathbf{y}_3(x) \\ &\vdots \\ \mathbf{y}'_{m-1}(x) &= \mathbf{y}_m(x) \\ \mathbf{y}'_m(x) &= \mathbf{g}(x, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m) \end{aligned}$$

La resolución numérica de un PVI trata de calcular una aproximación $\{y_i : i = 1, 2, \dots, N + 1\}$ en una red π

$$\pi : a = x_0 < x_1 < \dots < x_{N+1} = b,$$

a la solución exacta en cada punto de la red, es decir, $y_i \simeq \mathbf{y}(x_i)$. Para simplificar se empleará un paso de integración constante proporcional a la longitud del intervalo $[a, b]$: $h = h_i := x_{i+1} - x_i = (b - a)/n$, es decir, $x_i = a + ih, i = 1, \dots, N$

El método más sencillo para resolver (3.1) es el método de Euler explícito,

$$y_{i+1} = y_i + hf_i \quad i = 0, \dots, N \tag{3.3}$$

donde $f_i := f(x_i, y_i)$. Empezando con $y_0 = \alpha$, la fórmula (3.3) proporciona un algoritmo para obtener $\{y_i\}_{i=1}^N$.

Se van a considerar solamente métodos de un paso, es decir, aquellos que usan solo la información de la solución numérica en x_i para obtener la solución aproximada en $x_{i+1} = x_i + h$. Entre los métodos de un paso más empleados destacan los *Runge-Kutta* explícitos ([6]).

La forma general de un método Runge-Kutta explícito de s etapas para avanzar la solución de (x_i, y_i) a $(x_i + h, y_{i+1})$ viene dado por:

$$y_{i+1} = y_i + h \sum_{j=1}^s b_j g_j, \quad i = 0, 1, \dots, N$$

donde

$$g_j = f(x_i + hc_j, y_j + h \sum_{k=1}^{j-1} a_{jk} g_k),$$

siendo $\{c_j, b_j, a_{ij}\}$ constantes que hacen que el método RK tenga el orden más alto posible. A menudo usaremos la denominada tabla de Butcher para representar el esquema Runge-Kutta:

c_0	0	...	0	0
c_2	a_{21}	...	0	0
\vdots	\vdots			\vdots
c_s	a_{s1}	...	a_{ss-1}	0
	b_1	...	b_{s-1}	b_s

3.1.2. Métodos numéricos para problemas de contorno

Un sistema de EDOs de primer orden es a menudo complementado con m condiciones de contorno

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = \mathbf{0}$$

donde $\mathbf{g} = (g_1, \dots, g_m)^T : \mathbb{R}^{2m \times 2m} \rightarrow \mathbb{R}^m$. El caso más simple de \mathbf{g} es el de un PVI en donde la solución está dada en el punto inicial,

$$\mathbf{y}(a) = \alpha.$$

La forma general para el problema de contorno lineal de dos puntos para un sistema de primer orden (o para una EDO de mayor grado) es

$$B_a \mathbf{y}(a) + B_b \mathbf{y}(b) = \beta$$

donde B_a y $B_b \in \mathbb{R}^{n \times n}$.

El problema de condiciones de contorno se llama *separable* si es de la forma:

$$\begin{aligned} B_a \mathbf{y}(a) &= \beta_1 \\ B_b \mathbf{y}(b) &= \beta_2 \end{aligned}$$

Este tipo de condiciones separables aparecen en el estudio de las geodésicas. Entre los métodos numéricos para la resolución de problemas de contorno podemos encontrar métodos en diferencias, de tiro o de colocación pero la resolución numérica de este tipo de problemas cae fuera del alcance de esta memoria.

3.2. Geodésicas y líneas de curvatura

La colección de figuras de Galdeano tiene además como particularidad el dibujo de líneas de curvatura y geodésicas (ver figuras 3.1 y 3.2). Por ello se consideran en esta memoria.



Figura 3.1: Superficie de rotación de la tractriz con geodésicas.



Figura 3.2: Elipsoide con líneas de máxima y mínima curvatura.

3.2.1. Líneas de curvatura

Sea S una superficie, $P \in S$ y $\mathbf{x} : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ una parametrización con $P = \mathbf{x}(u_0, v_0)$ un punto regular, es decir, $x_u(u_0, v_0) \times x_v(u_0, v_0) \neq 0$.

Definimos el plano tangente de S en P al subespacio $T_P S$ engendrado por \mathbf{x}_u y \mathbf{x}_v . Para cada $P \in S$ la orientación que se elige en $T_P S$ es aquella que si tomamos una base positiva $\{\mathbf{x}_{u,P}, \mathbf{x}_{v,P}\}$, entonces $\{\mathbf{x}_{u,P}, \mathbf{x}_{v,P}, \mathbf{n}(P)\}$ es una base positiva de \mathbb{R}^3 en la que se define \mathbf{n} como el vector normal unitario a la superficie parametrizada,

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

Primeramente se introduce la primera forma fundamental [1], [9], $\mathbf{I}_P(U, V) = \langle U, V \rangle = U \cdot V$ para $U, V \in T_P S$. En la parametrización \mathbf{x} se tiene la base $\{\mathbf{x}_u, \mathbf{x}_v\}$ y se definen

$$E = \mathbf{I}_P(\mathbf{x}_u, \mathbf{x}_u) = \langle \mathbf{x}_u, \mathbf{x}_u \rangle,$$

$$F = \mathbf{I}_P(\mathbf{x}_u, \mathbf{x}_v) = \langle \mathbf{x}_u, \mathbf{x}_v \rangle,$$

$$G = \mathbf{I}_P(\mathbf{x}_v, \mathbf{x}_v) = \langle \mathbf{x}_v, \mathbf{x}_v \rangle,$$

que usualmente se representa en forma de matriz como:

$$\mathbf{I}_P = \begin{pmatrix} E & F \\ F & G \end{pmatrix}.$$

La segunda forma fundamental se define como,

$$\mathbf{II}_P = \begin{pmatrix} e & f \\ f & g \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{uu} \cdot \mathbf{n} & \mathbf{x}_{uv} \cdot \mathbf{n} \\ \mathbf{x}_{uv} \cdot \mathbf{n} & \mathbf{x}_{vv} \cdot \mathbf{n} \end{pmatrix},$$

y finalmente, la matriz más importante viene dada por:

$$\mathbf{I}_P^{-1} \mathbf{II}_P = \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \begin{pmatrix} e & f \\ f & g \end{pmatrix},$$

cuyos valores propios son las curvaturas principales de S en P y los vectores propios son las direcciones principales.

Definición. Una curva de S es una *línea de curvatura* si el vector tangente en cada punto es una dirección principal.

Ejemplo. Sea el elipsoide con parametrización,

$$\mathbf{x}(u, v) = \left(\cos(u) \cos(v), \frac{\cos(u) \sin(v)}{2}, \frac{\sin(u)}{2} \right), \quad u \in [0, 2\pi], \quad v \in [0, \pi]$$

La sentencia para su dibujo (ver fig. 3.3) en Mathematica es,

```
X={Cos[u] Cos[v], Cos[u] Sin[v]/2, Sin[u]/2};
dibu0=ParametricPlot3D[X, {u,0,2Pi},{v,0,Pi},Mesh->None,PlotRange->All,
Axes->False, Boxed->False]
```

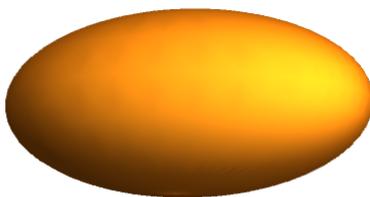


Figura 3.3: Elipsoide.

Entonces, $\mathbf{x}_u = \left(-\sin(u) \cos(v), -\frac{\sin(u) \sin(v)}{2}, \frac{\cos(u)}{2} \right)$, $\mathbf{x}_v = \left(-\cos(u) \sin(v), \frac{\cos(u) \cos(v)}{2}, 0 \right)$.

La matriz de la primera forma fundamental es:

$$\mathbf{I}_P = \begin{pmatrix} \frac{1}{4} \sin^2(u) \sin^2(v) + \sin^2(u) \cos^2(v) + \frac{\cos^2(u)}{4} & \frac{3}{4} \sin(u) \cos(u) \sin(v) \cos(v) \\ \frac{3}{4} \sin(u) \cos(u) \sin(v) \cos(v) & \frac{1}{4} \cos^2(u) \cos^2(v) + \cos^2(u) \sin^2(v) \end{pmatrix},$$

El vector normal es:

$$\mathbf{n} = \frac{-\cos^2(u)}{\sqrt{\cos^4(u) (1 + 3 \sin^2(v)) + \sin^2(2u)}} \begin{pmatrix} \cos(v), 2 \sin(v), 2 \tan(u) \end{pmatrix}$$

y junto con \mathbf{x}_{uu} , \mathbf{x}_{uv} y \mathbf{x}_{vv} se obtiene la segunda forma fundamental:

$$\mathbf{II}_P = \frac{\cos(u)}{\sqrt{\cos^4(u) (1 + 3 \sin^2(v)) + \sin^2(2u)}} \begin{pmatrix} 1 & 0 \\ 0 & \cos^2(u) \end{pmatrix}$$

Por último, la matriz $\mathbf{I}_P^{-1} \mathbf{II}_P$ es:

$$\frac{1}{d} \begin{pmatrix} 16 \cos(u) (3 \cos(2v) - 5) & 96 \sin(u) \cos^2(u) \sin(v) \cos(v) \\ 96 \sin(u) \sin(v) \cos(v) & 16 \cos(u) (\sin^2(u) (3 \cos(2v) + 5) + 2 \cos^2(u)) \end{pmatrix}$$

con $d = \left(3 \cos(2(u-v)) + 3 \cos(2(u+v)) + 6 \cos(2u) + 6 \cos(2v) - 26 \right) \sqrt{\cos^4(u) (1 + 3 \sin^2(v)) + \sin^2(2u)}$
cuyos valores propios son:

$$\lambda_1 = -\frac{16 \cos(u)}{(6 \cos^2(u) \cos(2v) + 3 \cos(2u) - 13)s}, \quad \lambda_2 = \frac{4 \cos(u)}{s}$$

con $s = \sqrt{\cos^4(u) (1 + 3 \sin^2(v)) + \sin^2(2u)}$, siendo sus vectores propios,

$$w_1 = (\sin(u) \cos(u) \cos(v), \sin(v)), \quad w_2 = (-\cos(u) \sin(v), \sin(u) \cos(v)).$$

Para el cálculo de las líneas de curvatura hay que resolver, en general numéricamente, los siguientes problemas de valor inicial:

$$\begin{cases} u'(t) = \sin(u(t)) \cos(u(t)) \cos(v(t)), & u(0) = u_0, \\ v'(t) = \sin(v(t)), & v(0) = v_0, \end{cases} \quad (3.4)$$

$$\begin{cases} u'(t) = -\cos(u(t)) \sin(v(t)), & u(0) = u_0, \\ v'(t) = \sin(u(t)) \cos(v(t)), & v(0) = v_0. \end{cases} \quad (3.5)$$

Para la integración numérica del PVI (3.4) se toma $y_1 = u$, $y_2 = v$, con condiciones iniciales, $y_1(0) = \frac{\pi}{4}$, $y_2(0) = \frac{\pi}{3}$, se utiliza el método de Runge-Kutta de orden 2 cuyos coeficientes se representan en la siguiente tabla de Butcher:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

el código programado en Mathematica es:

```
XX[u_,v_]={Cos[u] Cos[v],Cos[u] Sin[v]/2,Sin[u]/2};

y0={Pi/4,Pi/3}/N;
npasos=80;
b=8.;
a=0;
h=(b-a)/(npasos);      (* paso fijo 0.1 *)
poligonal=Table[0,{i,1,npasos+1}];
f[y_List]:={Sin[y[[1]]]Cos[y[[1]]]Cos[y[[2]]] ,Sin[y[[2]]] }

poligonal[[1]]= XX[y0[[1]],y0[[2]]] ;
For[i=1,i<=npasos, i++,
g1=f[y0];
g2=f[y0+h*g1];
y0 = y0 + h/2*(g1+g2);
poligonal[[i+1]]= XX[y0[[1]],y0[[2]]]
]
```

Finalmente se dibuja la línea de curvatura junto con el elipsoide (ver fig (3.4)) con las sentencias,

```
grafico1=Graphics3D[Tube[poligonal,0.02] ];
dibujo>Show[dibu0,grafico1]
```

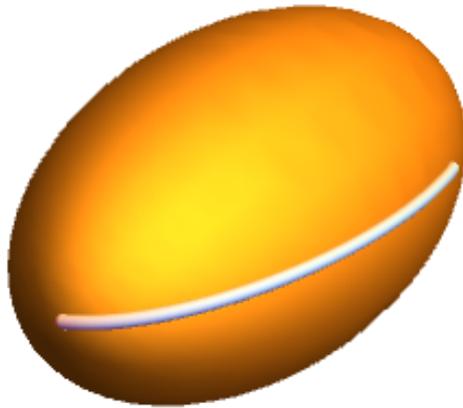


Figura 3.4: Elipsoide con línea de curvatura por método RK de orden 2.

Para la resolución numérica del otro sistema de líneas de curvatura dado por el PVI (3.5) se emplea en este caso el método RK de orden 4 cuya tabla de Butcher es:

0	0			
1/2	1/2	0		
1/2	0	1/2	0	
1	0	0	1	0
	1/6	2/6	2/6	1/6

Haciendo el cambio $y_1 = u$, $y_2 = v$ y tomando como condiciones iniciales, $y_1(0) = \pi$, $y_2(0) = \frac{\pi}{4}$ la implementación en Mathematica es:

```

XX[u_,v_]={Cos[u] Cos[v],Cos[u] Sin[v]/2,Sin[u]/2};

y0={Pi,Pi/4}/N;
npasos=40;
b=8.;
a=0;
h=(b-a)/(npasos);      (* paso fijo 0.2*)
poligonal=Table[0,{i,1,npasos+1}];
f[y_List]:={-Cos[y[[1]] Sin[y[[2]]],Cos[y[[2]] Sin[y[[1]]] }

poligonal[[1]]= XX[y0[[1]],y0[[2]]] ;
For[i=1,i<=npasos, i++,
g1=f[y0];
g2=f[y0+h/2*g1];
g3=f[y0+h/2*g2];
g4=f[y0+h*g3];
y0 = y0 + h/6*(g1+2g2+2g3+g4);
poligonal[[i+1]]= XX[y0[[1]],y0[[2]]]
]

```

Finalmente, se dibuja la línea de curvatura en el elipsoide, figura (3.5)

```
grafico1=Graphics3D[Tube[poligonal,0.02] ];
dibujo=Show[dibu0,grafico1]
```

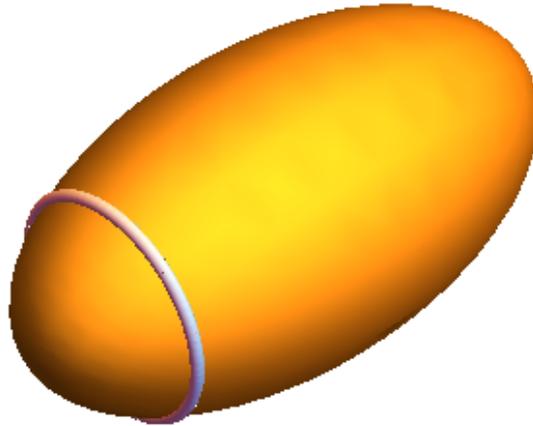


Figura 3.5: Elipsoide con línea de curvatura por método RK de orden 4.

3.2.2. Geodésicas

Intuitivamente, una geodésica es una curva que minimiza la distancia entre dos puntos de una superficie al igual que hacen las rectas en el espacio euclídeo. En esta sección se definen rigurosamente y se aproximarán resolviendo numéricamente los sistemas diferenciales asociados.

Definición. Sea S una superficie, $\gamma: I \rightarrow S$ una curva regular C^∞ y $X: I \rightarrow \mathbb{R}^3$ un campo de S a lo largo de γ . La *derivada covariante* de X a lo largo de γ es el campo de S a lo largo de γ definido como

$$\frac{DX}{dt}(t) := \dot{X}(t) - \langle \mathbf{n}, \dot{X}(t) \rangle \mathbf{n}$$

Definición. Sea S una superficie y $\gamma: I \rightarrow S$ una curva parametrizada C^∞ . Diremos que γ es una geodésica ([1], [5]) si $\frac{D\dot{\gamma}}{dt}$ es idénticamente nula.

Puede verse detallado en [1] que $\frac{D\dot{\gamma}}{dt}$ se expresa como:

$$\begin{aligned} & (u''(t) + \Gamma_{u,u}^u u'(t)^2 + 2\Gamma_{u,v}^u u'(t)v'(t) + \Gamma_{v,v}^u v'(t)^2) \mathbf{x}_u^\gamma + \\ & (v''(t) + \Gamma_{u,u}^v u'(t)^2 + 2\Gamma_{u,v}^v u'(t)v'(t) + \Gamma_{v,v}^v v'(t)^2) \mathbf{x}_v^\gamma \end{aligned}$$

Notación: Las funciones $\Gamma_{i,j}^k$ se llaman *símbolos de Christoffel*:

$$\frac{D\mathbf{x}_u}{\partial u} = \Gamma_{u,u}^u \mathbf{x}_u + \Gamma_{u,v}^u \mathbf{x}_v, \quad \frac{D\mathbf{x}_u}{\partial v} = \frac{D\mathbf{x}_v}{\partial u} = \Gamma_{u,v}^u \mathbf{x}_u + \Gamma_{u,v}^v \mathbf{x}_v, \quad \frac{D\mathbf{x}_v}{\partial v} = \Gamma_{v,v}^u \mathbf{x}_u + \Gamma_{v,v}^v \mathbf{x}_v$$

Entonces para que una curva $\gamma(t) = \mathbf{x}(u(t), v(t))$ sea una geodésica se deben cumplir las ecuaciones diferenciales:

$$u''(t) + \Gamma_{u,u}^u u'(t)^2 + 2\Gamma_{u,v}^u u'(t)v'(t) + \Gamma_{v,v}^u v'(t)^2 = 0$$

$$v''(t) + \Gamma_{u,u}^v u'(t)^2 + 2\Gamma_{u,v}^v u'(t)v'(t) + \Gamma_{v,v}^v v'(t)^2 = 0$$

La unicidad de solución de este sistema de ecuaciones diferenciales dadas las condiciones iniciales o de contorno dará la geodésica buscada.

Ejemplo. Sea el hiperboloide con parametrización:

$$\mathbf{x}(\theta, \phi) = (\cosh(\theta) \cos(\phi), \cosh(\theta) \sin(\phi), \sinh(\theta)), \quad \theta \in (0, 2\pi), \quad \phi \in (-5, 5)$$

Se dibuja en Mathematica,

```
hiper={Cosh[\[Theta]] Cos[v],Cosh[\[Theta]] Sin[v],Sinh[\[Theta]]};
```

```
dibu0=ParametricPlot3D[hiper,{v,0,2Pi},{\[Theta],-5,5},Mesh->None ,
Axes->False,Boxed->False,PlotPoints->50,PlotTheme->"ThickSurface",
RegionFunction->Function[{x,y,z},x^2+y^2<5^2]];
```



Las ecuaciones diferenciales de las geodésicas son:

$$\phi''(t) = -2\theta'(t) \tanh(\theta(t)) \phi'(t) \quad ; \quad \theta''(t) = -\theta'(t)^2 \tanh(2\theta(t)) + \frac{1}{2} \tanh(2\theta(t)) \phi'(t)^2$$

Dado el sistema de ecuaciones diferenciales de segundo orden podemos hallar la única geodésica que pasa por un punto y tiene un vector tangente a la superficie dado o aquella que une dos puntos de la superficie. En el primer caso resolveremos el problema de valor inicial (PVI: (3.6)) y en el segundo el problema de contorno (PC: (3.7)).

$$\begin{cases} \phi''(t) = -2\theta'(t) \tanh(\theta(t)) \phi'(t), & \phi(t_0) = \phi_0, \quad \phi'(t_0) = \phi'_0 \\ \theta''(t) = -\theta'(t)^2 \tanh(2\theta(t)) + \frac{1}{2} \tanh(2\theta(t)) \phi'(t)^2, & \theta(t_0) = \theta_0, \quad \theta'(t_0) = \theta'_0 \end{cases} \quad (3.6)$$

$$\begin{cases} \phi''(t) = -2\theta'(t) \tanh(\theta(t)) \phi'(t), & \phi(t_0) = \phi_0, \quad \phi(t_1) = \phi_1 \\ \theta''(t) = -\theta'(t)^2 \tanh(2\theta(t)) + \frac{1}{2} \tanh(2\theta(t)) \phi'(t)^2, & \theta(t_0) = \theta_0, \quad \theta(t_1) = \theta_1 \end{cases} \quad (3.7)$$

El PVI se puede resolver mediante el método de RK de orden 2 visto en la sección anterior 3.1. Primero se transforma el sistema de segundo orden en uno de primero,

$$\begin{array}{ll} y_1 = \theta & y'_1 = y_2 \\ y_2 = \theta' & y'_2 = -\tanh(2y_1)y_2^2 - \frac{\tanh(2y_1)}{2}y_4^2 \\ y_3 = \phi & y'_3 = y_4 \\ y_4 = \phi' & y'_4 = -2\tanh(y_1)y_2y_4 \end{array}$$

se toman las condiciones iniciales $y_1(0) = -2, y_2(0) = 1, y_3(0) = \frac{\pi}{2}, y_4(0) = -1$.

```

hiper2[u_,v_]={Cosh[u] Cos[v],Cosh[u] Sin[v],
Sinh[u]};

f[y_List]:={y[[2]], -(Tanh[2y[[1]]] ] *(y[[2]])^2- Tanh[2y[[1]]]/2 *(y[[4]])^2),
y[[4]],-(2 Tanh[y[[1]]] *y[[2]]*y[[4]])}

y0={-2,1,Pi/2,-1};//N;
npasos=20;
b=2.;
a=0;
h=(b-a)/(npasos);    (* paso fijo 0.1 *)

poligonal=Table[0,{i,1,npasos+1}];
poligonal[[1]]= hiper2[y0[[1]],y0[[3]]] ;

For[i=1,i<=npasos, i++,
g1=f[y0];
g2=f[y0+h*g1];
y0 = y0 + h/2*(g1+g2);
poligonal[[i+1]]= hiper2[y0[[1]],y0[[3]]] ;
]

```

La representación gráfica de la geodésica en el hiperboloide viene dada por (ver fig. (3.6)).

```

grafico1=Graphics3D[Tube[poligonal,0.22]];
dibujo=Show[dibu0,grafico1]

```

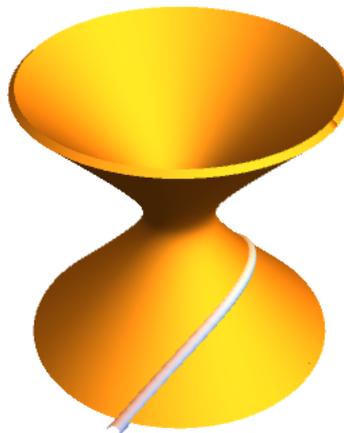


Figura 3.6: Geodésica del PVI con método RK de orden 2.

Como no se han visto métodos numéricos para problemas de contorno, (3.7) se resuelve en Mathematica siendo $\phi(0) = 0$, $\theta(0) = -3$, $\phi(1) = \pi$, $\theta(1) = 4$,

```
soluC=NDSolve[{2 Tanh[u[t]] *u'[t]*v'[t]+v''[t]==0,
Tanh[2\Theta[t]] *(u'[t])^2- Tanh[2\Theta[t]]/2 *(v'[t])^2+
u''[t]==0,
u[0]==-3,v[0]==0,u[1]==4,v[1]==Pi},{u,v},t];
```

Y se dibuja junto con el hiperboloide (Ver fig.3.7)

```
hiper1={Cosh[u[t]] Cos[v[t]],Cosh[u[t]] Sin[v[t]],
Sinh[u[t]]}

dibu1=ParametricPlot3D[Evaluate[hiper1/.soluC],
{t,0,1},PlotStyle ->Directive[Red,Tube[0.22]],PlotRange ->All]

Show[dibu0, dibu1]
```

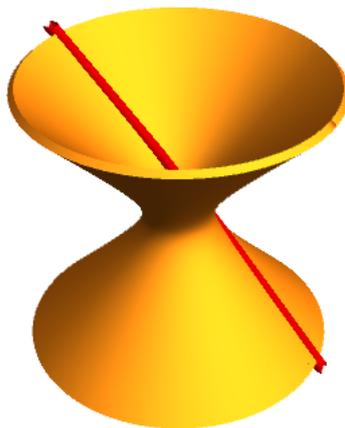


Figura 3.7: Geodésica del PC

Capítulo 4

Conclusiones

La impresión 3D puede ser una buena herramienta para el aprendizaje y motivación de la ciencia pues puede acercarla de manera tangible e interesante. Sin embargo, hay que resaltar algunos aspectos negativos que pueden dificultar el uso de esta herramienta: requiere de mucho tiempo en la impresión por lo que no resulta útil en muchas ocasiones; es impredecible pues procesos similares pueden dar lugar a resultados distintos y es necesario ajustar adecuadamente sus características como la velocidad de impresión, la temperatura del extrusor...

4.1. Aplicaciones

Una posible aplicación de este trabajo es la generación e impresión de orbitales atómicos, interesante en el campo de la química para su divulgación y enseñanza.



Figura 4.1: Orbitales atómicos

Otra aplicación interesante sería la impresión de superficies de “Imaginary, una mirada matemática”, exposición itinerante de la Real Sociedad Matemática Española.

4.2. Líneas futuras de trabajo

Han quedado aspectos sin tratar debido al alcance de este trabajo y que podrían ser interesantes para su ampliación. Se podría hacer un estudio de las geodésicas en superficies dadas en coordenadas implícitas o estudiar métodos numéricos para la resolución de problemas de contorno.

4.3. Agradecimientos

Agradecer al IUMA (Instituto Universitario de investigación en Matemáticas y Aplicaciones) el uso de su impresora 3D y del ácido poliláctico PLA.

Bibliografía

- [1] E. ARTAL, *Geometría de curvas y superficies*, apuntes curso 2016/2017. Universidad de Zaragoza.
- [2] M. BURNS, *The STL Format*, Automated Fabrication, Section 6.5. http://www.fabbers.com/tech/STL_Format.
- [3] J. BERNUÉS PARDO, M. T. LOZANO IMÍZCOZ, I. POLO BLANCO, *La colección de modelos matemáticos de la Universidad de Zaragoza*. La Gaceta de la RSME, Vol. 15 (2012), Núm. 1, Págs. 187-204.
- [4] D. CHAKRAVORTY, *STL File Format*, <https://all3dp.com/what-is-stl-file-format-extension-3d-printing/>.
- [5] L. J. HERNÁNDEZ PARICIO, *Introducción a la Geometría Diferencial*, Universidad de La Rioja. <http://www.unirioja.es/cu/luhernan/lnes.html>.
- [6] E. HAIRER, S. P. NØRSETT, G. WANNER, *Solving Ordinary Differential Equations I*, Springer Science & Business Media, 2008.
- [7] M. T. LOZANO IMÍZCOZ, *Fichas de los modelos matemáticos donados por Don Zoel García de Galdeano a la Facultad de Ciencias de la Universidad de Zaragoza*, 2011.
- [8] *OpenSCAD user manual*, https://en.wikibooks.org/wiki/OpenSCAD_User_Manual.
- [9] T. SHIFRIN, *A first course in curves and surfaces.*, University of Georgia, 2016.

APÉNDICE 1

Fichas de algunos modelos de la colección
de figuras de Galdeano ([7])

Serie I nº1: Superficie de rotación de la tractriz

(pinche aquí)

Nombre original: Rotationsfläche der Tractrix

Información matemática: Es la superficie de revolución en torno al eje OZ de una tractriz situada en el plano XZ. Esta tractriz es la curva cuya tangente en cada punto corta al eje OZ a una distancia constante. Es la trayectoria que describe un objeto P arrastrado por otro Q que se desplaza a lo largo del eje OZ manteniendo con él una distancia constante. La ecuación de la superficie en paramétricas es:

$$\begin{aligned}x &= \cos u \sin v \\y &= \sin u \sin v \\z &= \cos v + \ln\left(\tan \frac{v}{2}\right)\end{aligned}$$

Serie II nº4: Superficie de rotación del senh

(pinche aquí)

Información matemática: Es una superficie con curvatura constante positiva. Es la superficie de revolución en torno al eje OZ de la función seno hiperbólico de z en el plano XZ ($x = \sinh(z)$). La ecuación de la superficie en paramétricas es:

$$\begin{aligned}x &= \cos u \sinh v \\y &= -\sin u \sinh v \\z &= -v\end{aligned}$$

Serie II nº1a: Superficie de Kummer

(pinche aquí)

Nombre original: Kummer'sche Fläche

Información matemática: Una superficie de Kummer es una cuártica (superficie algebraica de grado 4) con 16 puntos dobles ordinarios. En este caso tiene los 16 puntos dobles. Existen 16 planos tal que cada uno de ellos pasa por 6 puntos dobles y cada uno de los 16 puntos dobles pertenece a 6 de estos planos. Su ecuación es:

$$x^4 + y^4 + z^4 - 4x^2 - 4y^2z^2 - 4y^2 - 4z^2x^2 - 4z^2 - 4x^2y^2 + 12\sqrt{3}xyz + 1 = 0$$

Serie III nº4: Elipsoide

(pinche aquí)

Información matemática: La ecuación de esta superficie es:

$$\frac{x^2}{A} + \frac{y^2}{B} + \frac{z^2}{C} = 1$$

Las líneas impresas son líneas de curvatura (máxima y mínima) de la superficie.

Serie VI nº6: Curva espacial cúbica sobre un cilindro elíptico:elipse cúbica.

(pinche aquí)

Nombre original: Raumcurve 3. Ordnung.

Información matemática: Sobre un cilindro elíptico de ecuación:

$$\frac{x^2}{a} + \frac{y^2}{b} = 1,$$

se dibuja la curva intersección con un cono de ecuación:

$$x^2 + y^2 - z^2 = 0$$

El resultado es una curva espacial, la elipse cúbica.

Serie VII nº2: Superficie de Cayley

(pinche aquí)

Nombre original: Fläche dritter Ordnung.

Información matemática: Es una cúbica con 4 puntos singulares (nodos) que están dispuestos como los vértices de un tetraedro. Los lados de este tetraedro son 6 de las 9 rectas reales contenidas en la superficie. Las otras 3 están en un plano horizontal. Una ecuación es:

$$2(xy + xz + yz) - 5(x^2y + x^2z + xy^2 + xz^2 + y^2z + yz^2) = 0$$

Serie VII n°16: Superficie de orden 3 con un punto singular U_6

(pinche aquí)

Nombre original: Fl. 3. Ord. mitunipl. Knotenpunkt U_6

Información matemática: Superficie de orden 3 con un punto singular U_6 cuyo plano tangente interseca a la superficie en 3 rectas. Una ecuación de esta superficie es :

$$z^2(z+4) + y(y - \sqrt{3}x)(y + \sqrt{3}x) = 0$$

Serie VII nº17: Superficie de orden 3 con un punto singular U_6

(pinche aquí)

Nombre original: Fl. 3. Ord. Mit unipl. Knotenpunkt U_6

Información matemática: Superficie de orden 3. Tiene un punto singular de tipo U_6 y dos rectas reales, dibujadas en el modelo. Su ecuación es

$$z^3 + 4z^2 + yx^2 + y^3 = 0$$

Serie VII nº18: Superficie de orden 3 con un punto singular U_7

(pinche aquí)

Nombre original: Fl. 3. Ord. Mit unipl. Knotenpunkt U_7

Información matemática: Superficie de orden 3. Tiene un punto singular de tipo U_7 y tres rectas reales. Su ecuación es

$$x^3 + 4x^2 + zy^2 + xz^2 = 0$$

Serie VII nº 19: Superficie de orden 3 con un punto singular U_8

(pinche aquí)

Nombre original: Fl. 3. Ord. Mit unipl. Knotenpunkt U_8

Información matemática: Tiene un único punto singular y una sola recta real. Una ecuación es:

$$z^3 + 4z^2 + y^2z + x^3 = 0$$

Serie VII nº 20: Superficie reglada de orden 3

(pinche aquí)

Nombre original: Regelfläche 3. Ordnung

Información matemática: Superficie de orden 3. Es una superficie reglada con una línea de puntos dobles. Su ecuación es

$$y^3 + y^2z + yx^2 - 2xy - zx^2 = 0$$

Serie VII nº 25: Superficie de orden 3. Hessiana de la superficie del modelo VII 7

(pinche aquí)

Nombre original: Hesse'sche Fl. Einer Fl. 3. Ord.

Información matemática: La superficie Hessiana de una superficie S en el espacio proyectivo dada por un polinomio $f(x_1, x_2, x_3, x_4) = 0$ es la superficie definida por el determinante de la matriz Hessiana de f . En este caso S es la superficie de Cayley del modelo VII 7 y la ecuación es:

$$25056 - 60534x^2 + 6237x^3 - 60534y^2 - 18711xy^2 - 14544z + 50805x^2z + 27x^3z + 50805y^2z - 81xy^2z - 3336z^2 - 8103y^2z^2 + 564z^3 + 100z^4 = 0$$

Serie X nº5: Cíclide de Dupin

(pinche aquí)

Nombre original: Dupin'sche Cyclide

Información matemática: La cíclide de Dupin se obtiene de un toro por inversión en una esfera. Es un ejemplo de superficie de orden 4. En este caso el centro de inversión pertenece al toro de partida, de manera que el resultado contiene al punto del infinito. Se dice que es una cíclide toroidal parabólica. La inversión del toro de ecuación:

$$(R^2 - r^2 + (x - e)^2 + y^2 + z^2)^2 - 4R^2((x - e)^2 + y^2) = 0$$

en la esfera unidad

$$x^2 + y^2 + z^2 = 1$$

produce la cíclide

$$((x^2 + y^2 + z^2)(e^2 - r^2 + R^2) + 1 - 2ex)^2 - 4R^2((-e(x^2 + y^2 + z^2) + x)^2 + y^2) = 0$$

Si $R = 2$, $r = 1$, $e = 1$ se tiene la cíclide de Dupin

$$1 - 4x - 12x^2 - 16y^2 + 8(x^2 + y^2 + z^2) + 16x(x^2 + y^2 + z^2) = 0$$

APÉNDICE 2

Imágenes de los modelos impresos.

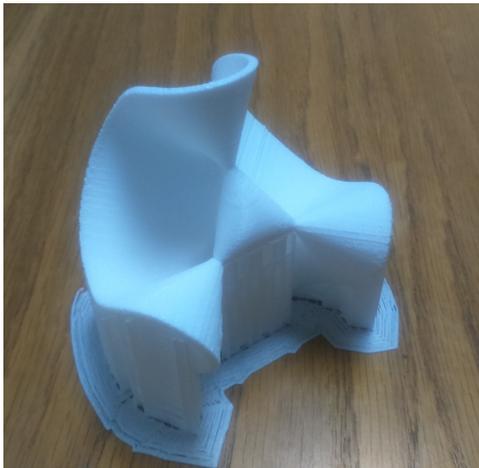


Figura 4.2: Modelo Cayley, serie VII n°2

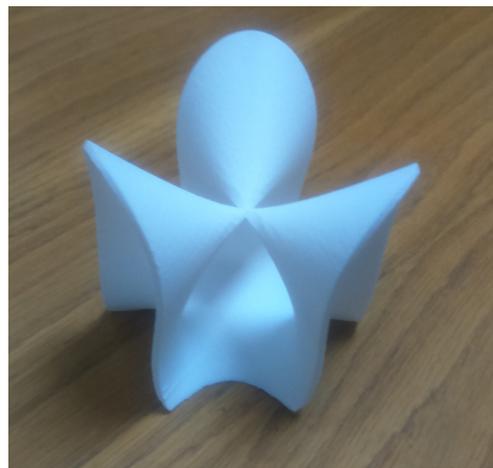


Figura 4.3: Serie VII n°16



Figura 4.4: Serie VII n°18



Figura 4.5: Serie VII n°20



Figura 4.6: Serie VII n°25

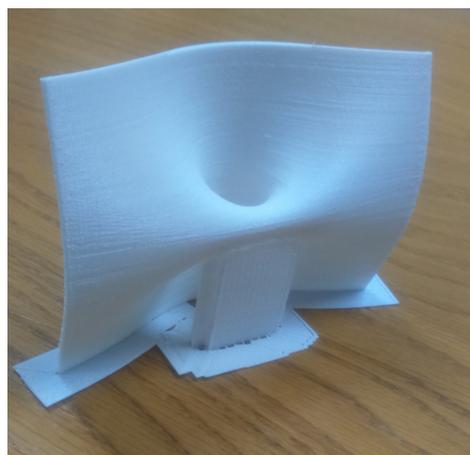


Figura 4.7: Serie X n°5



Figura 4.8: Serie VII n°1, superficie de Clebsch. Contiene 27 rectas reales



Figura 4.9: Hiperboloide con geodésica



Figura 4.10: Serie VI n°6, Curva espacial cúbica sobre un cilindro elíptico: elipse cúbica