

Summary

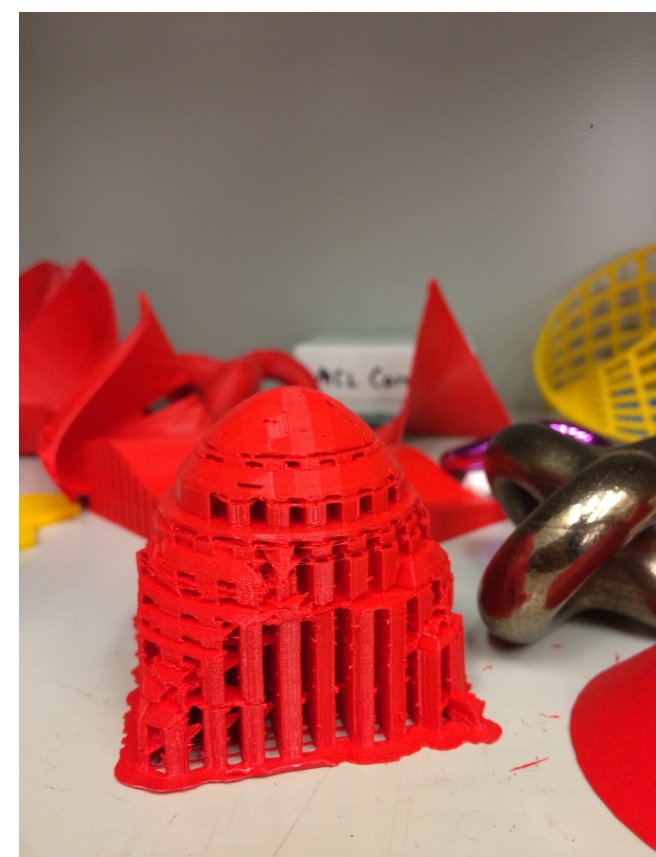
We are three Calculus-III students who have spent the semester using a 3-D printer to print multi-variable functions. We have been working with an Ultimaker2 printer and have learned to use programs such as: Cura, Blender, and SageMath. We started the semester with little to no understanding of our selected topic; and by the end of the semester, we have been able to successfully print objects that were expressed mathematically.

Motivation

This project is to explore surfaces in three-dimensional space by 3D printing models of them. The goal is to start by choosing appropriate equations and end with physical models which would be an accurate representation for use in multi-variable calculus classes.

Difficulties

The first function we attempted to print can be described by the equation $z = 1 - (x^2 + y^2)$. We created the function in Blender under the math-functions extension without adding any thickness. The result is as follows:



A function we printed that can be described as the equation $z = (x + y)^2 / (x^2 + y^2)$. It was created with the same math function extension as the previous function, with a low x-step and y-step that resulted in a print with a jacked bottom. The results is as follows:



Materials/Programs

- ▶ Ultimaker2
- ▶ Cura
- ▶ Blender
- ▶ SageMath
- ▶ .025mm Ultimaker PLA filament
- ▶ Exacto-knife
- ▶ rubbing alcohol

Process

- ▶ Enter your function into blender as either $z(x,y)$ or $x(u,v)$, $y(u,v)$, $z(u,v)$
- ▶ Identify the xy-steps or the uv-steps, to adjust the prints smoothness. The higher the step the smoother the print, but it will take longer to finish.
- ▶ Add the amount of thickness to your function for it to be printable. The greater the thickness the stronger the print will be, but it will take longer to finish.
- ▶ When all adjustments are made, export the file as a .stl. Open Cura and import the stl file
- ▶ We recommend scaling and rotating the print in Cura, because it gives you a visual of what it will look like in the printer when it is complete.
- ▶ Turn on the 3d printer and preheat the nozzle to 250 degrees Celsius and the plate to 50.
- ▶ Insert a micro usb into the computer and import the file into it using Cura. Then insert the usb into the Ultimaker2, find the file in the menu options and hit print.
- ▶ Once finished, peel the print off the base and use an Exacto knife to cut away the support structure.

Ruled Surface Sage Code

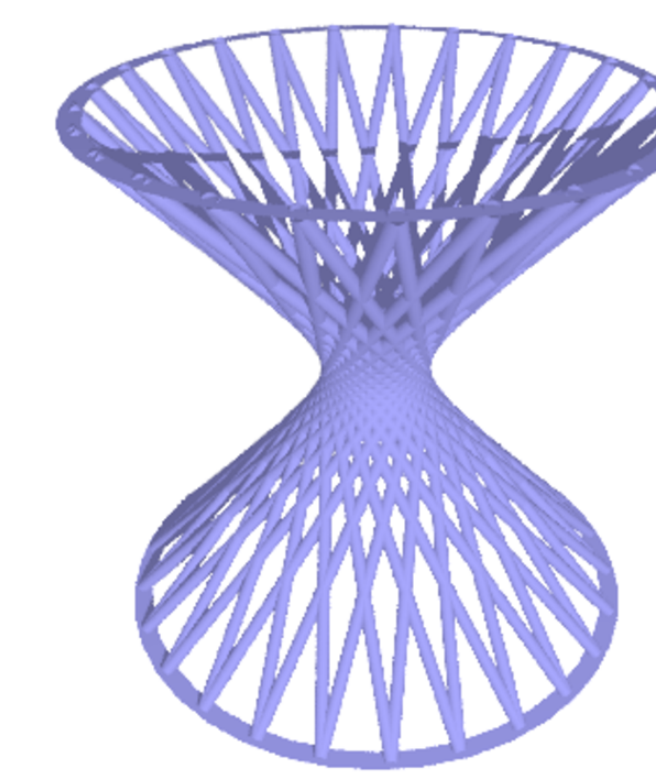
```
var('t, d, x, y, z')
G = 0
C1 = circle((0,0,5), 5)
C2 = circle((0,0,-5),5)
for d in range(16):
    g = vector([cos(d*pi/8), sin(d*pi/8), 0])
    h = vector([-sin(d*pi/8), cos(d*pi/8), 1])
    P = g+(t*h)
    Q = -P
    n = parametric_plot3d(P, (t,-5,5))
    m = parametric_plot3d(Q, (t,-5,5))
    G += n + m
G = G + C1 + C2
K = show(G, axes=false)
K
```

```
from sage.plot.plot3d.shapes2 import Line
var('x y z k')
N = 25
k = 11
a = sum(Line([(cos(2*pi*(n+k)/N), sin(2*pi*(n+k)/N), 0), (cos(2*pi*n/N), sin(2*pi*n/N), 1))] for n in range(N))
a += sum(Line([(cos(2*pi*(n+k)/N), sin(2*pi*(n+k)/N), 1), (cos(2*pi*n/N), sin(2*pi*n/N), 0))] for n in range(N))
a += parametric_plot3d((cos(x)*(1+y), sin(x)*(1+y), 0), (x, 0, 2*pi), (y, -1/25, 1/25))
a += parametric_plot3d((cos(x)*(1+y), sin(x)*(1+y), 1), (x, 0, 2*pi), (y, -1/25, 1/25))
```

Ruled Surface

A surface is considered ruled if through every point on the surface there is a straight line that lies along the surface. The hyperboloid below is considered doubly ruled. For every point on the surface lies two distinct lines.

Ruled Hyperboloid



Hyperboloid Equation

The Cartesian equation of hyperboloid:

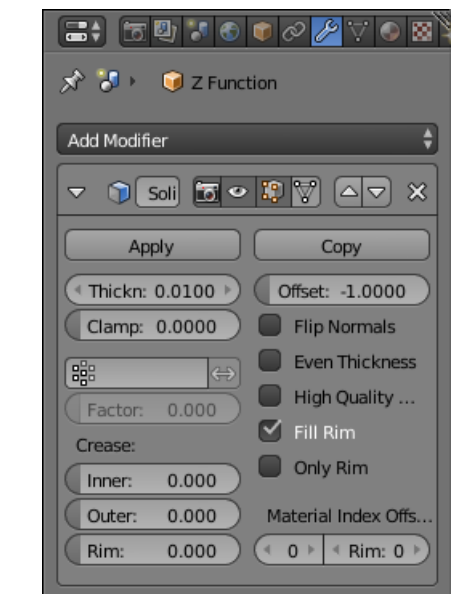
$$\frac{x^2}{a^2} + \frac{y^2}{a^2} - \frac{z^2}{c^2} = 1$$

The parametrized equation:

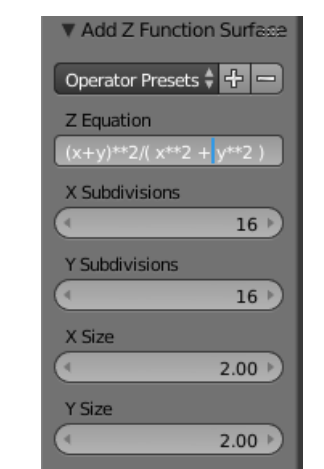
$$\begin{aligned} x &= a\sqrt{1+u^2}\cos v \\ y &= a\sqrt{1+u^2}\sin v \\ z &= cu \end{aligned}$$

Improvements

To fix our problem with the printer not printing a function that was too thin, we increased to thickness to a large enough size that the printer could read to make the function. Here's how we increased the thickness in blender:



Increasing the step for any function causes the print to be a lot smoother and gets rid of the rough edges. We did this by asking blender to plot more x and y points. Here's how we increased the smoothness in Blender:



Conclusions

- ▶ Throughout the semester, we have learned about calculus in multivariables and about the software needed to interact with a 3-D printer. Along with this, we have drastically improved the quality of our prints and our understanding of why these functions do the things they do.
- ▶ We have taken a few of these functions into our calculus lectures to add as a visual aid for the professor.
- ▶ We were able to parameterize a hyperboloid and express it as a collection of tangent lines.
- ▶ As the project has progressed, the easier it has become to work with the printer to create a good print. It has also given us tools that we can use further on in our academic careers to succeed, such as SageMath, which is an incredibly powerful open-source mathematics software.
- ▶ Unfortunately, we were not able to do everything that we would have like to have done.
- ▶ If we had more time, we were going to make a function with a tangent plane, evaluated at a point, attached to it.
- ▶ We were talking about making a Clebsch Surface and defining the function so the 27 straight lines that comprise the surface would be raised and visible. This would be an extremely useful print that could be used as a visual in math lectures.