東　　海　　大　　學

應　　用　　數　　學　　系

碩　　士　　論　　文

數學函數之 3D 模型建立與列印

REAL FORM CREATION OF MATHEMATICAL FUNCTIONS VIA SOFTWARE AND 3D PRINTERS

研 究 生　：潘諾曼

指導教授：黃皇男

中華民國一O五年一月

數學函數之 3D 模型建立與列印

REAL FORM CREATION OF MATHEMATICAL FUNCTIONS VIA SOFTWARE AND 3D PRINTERS

研 究 生 : 潘 諾 曼

Student : Norman Mario Pasaribu

指 導 教 授 : 黃 皇 男

Advisor : Huang-Nan Huang

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Applied Mathematics, College of Science

TUNGHAI UNIVERSITY

January 2016

中華民國 一 **O** 五年一月

東　海　大　學

應用數學系

碩士學位口試委員審定書

本系碩士班　潘諾曼　　君

所提論文　REAL FORM CREATION OF MATHEMATICAL FUNCTIONS VIA SOFTWARE AND 3D PRINTERS
（數學函數之3D模型建立與列印）

合於碩士班資格水準，業經本委員會評審通過，特此證明。

口試委員：陳淑[illegible]

陳碩聰

指導教授：黃皇男

所　　長：楊智烜

中　華　民　國　　一〇五　年　一　月　二十二　日

TUNGHAI UNIVERSITY

DEPARTMENT OF APPLIED MATHEMATICS

Student's Name : Norman Mario Pasaribu

Thesis Title : REAL FORM CREATION OF MATHEMATICAL FUNCTIONS VIA SOFTWARE AND 3D PRINTERS (數學函數之 3D 模型建立與列印)

Advisor : Huang-Nan Huang

Day/ Date : Friday/ January 22nd, 2015

Oral Defense Committee Members:

1. **陳碩聰 — Chen, Shuo-Tsung**
   Department of Electronic Engineering,
   National Formosa University, Taiwan

2. **陳淑珍 — Chen, Shu-Zhen**
   Department of Applied Mathematics,
   Tunghai University, Taiwan

ABSTRACT

This thesis studies on the processing of using 3D printers to generate the real 3D solid object corresponding to a given mathematical function. First of all, the surface object of the mathematical function should be generated by using commercial mathematical software like Mathematica, Maple, Matlab, or free mathematical modeling software such as MathMod, K3DSurf etc. Later on the object file is sent to the free software Blender or Netfabb for adding the thickness to the surface with output as a STL file. Finally, the specified 3D printer's software reads in the STL file and drives the 3D printer to form the solid object. The possible difficulties during this procedure and efficiency comparison between mathematical software in generating the surface object are also clarified such that the interested person can get in very quickly.

Keywords: Geometry, 3D modeling, mathematical functions, STL file format, solid object

中文摘要

本論文討論將數學函數所代表的 3D 物件透過 3D 印表機列印出來的完整過程，讓數學函數變成可以觸碰的真實物件。首先採用如 Mathematica, Maple 或 Matlab 等商業數學軟體或是 MathMod, K3DSurf 等的自由建模軟體，建立兩個變數函數或是 3 度空間隱函數的曲面圖形檔；進一步使用字自由軟體，Blender 或 Netfabb 調整曲面的厚度，存為 3D 物件常見的 STL 檔案；最後一步將此檔案透過 3D 印表機軟體，來驅動 3D 印表機輸出實體物件。文中比較不同軟體製作曲面圖形檔的效益，以及可能遭遇的問題，作為想要利用 3D 印表機列印數學函數的實體物件者參考。

關鍵字: 幾何, 3D 模型, 數學函數, STL 檔案, 實體物件

Contents

List of Tables

List of Figures

Acknowledgements

Praise The Lord who has given me the grace and a life until this time. Thanks to Tunghai University, Officials of International Student Affairs who agreed and accepted me as a student to learn and grow better in this University. I would like to thank my Department (Department of Mathematics). Especially to my Advisor Prof. Huang-Nan Huang who has advised, guided and taught me during this research process and also for all of my teachers.

Special thanks to my big family in Indonesia, my second family in Taiwan (IFGF Taichung), my friends in ISA THU, my boss 何亘翊老師 in the student labor office of Tunghai University who had given me so much support and prayer such that I could pass my live in Taiwan.

And I don't forget to thank my research team member: 楊秉諺, 陳銘郎, and 王世銘 who accompanied me during this research process until we finished it.

Finally I thank to every people that I could not mention one by one. And I hope this research may give us any advantages.

And I would like to dedicate my Thesis to my big family.

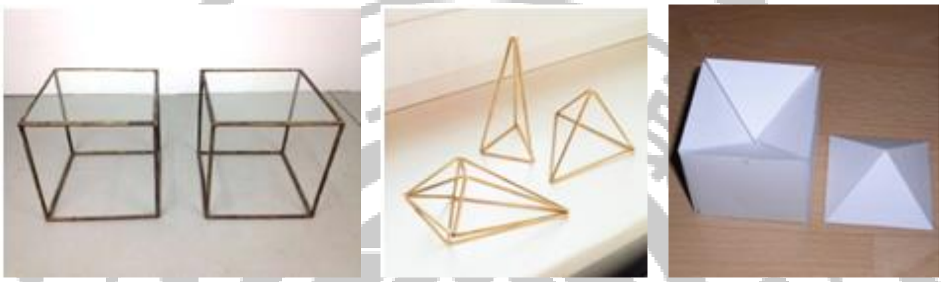
I

Introduction

In the seventeenth century there was a man as a mathematician referred to Mathematics as "*The Queen of the Sciences*". He is Johann Carl Friedrich Gauss born in Germany (30 April 1777–23 February 1855). He is also an astronomer and physicist[1]. The human life is never apart from Mathematics and there is no exception to anyone. We realize that Mathematics is always present in our daily lives. In the modern world with a modern education, Mathematics is one of the fundamental lessons that be certainly taught at schools. At kindergarten level mathematics has been introduced and taught to the students. And the students never stop to learn Mathematics until they finished Ph.D. or the other education degree.

A lot of things are learned in mathematics. Mathematics consists of 7 major divisions such as Arithmetic, Algebra, Analysis, Applied Mathematics, Combinatory, Foundations, Geometry and Topology. Every division also learns a lot of knowledge. Geometry is one of the interesting parts to learn in Mathematics. It is because of geometry learns about points, straight lines, curves, surfaces, and solids. Geometry has some division that very interesting enough to be learned like: Euclidean plane geometry ("2D space"), three-dimensional Euclidean geometry ("3D space") and n-dimensional Euclidean geometry ("n-D space"). Although quite interesting, to understand the contents of Geometry is not so easy. It is caused by the study of this lesson is more likely to be abstract so that an understanding of the contents has not been quite satisfied.

Before the computer has been invented and used in the world of education, to give an overview or explanation of the geometry lesson to the students is difficult. Lessons are just explained by using the abstract object and just told the students the mathematical functions on geometry. It could give an adverse influence to the interest of student learning. To reduce the adverse effects, teachers try to find solutions[2]. Teachers usually used the blackboard to draw the picture or image to enhance the content. Sometimes teachers would use the real object to explain the lesson which it was created by their selves. Teachers used a lot of kinds of material to construct the real object of geometry such as iron wire, bamboo, paper, clay, woods, and so forth.



iron wire woods paper

Figure 1.1: The DIY geometry objects

After the computer has been invented and often used in the world of education, teachers are very helpfully to explain the lesson of Geometry. Teachers could use computer software to draw or figure the geometry object and plot the graph of mathematical functions. And so far, there is a lot of 3D software or 3D programs had been created and used to describe the mathematical object (geometry objects) such as Geometric Sketchpad, Cabri, Geogebra, K3DSurf, Matlab, MathMod, Mathematica, Maple, Rhinoceros, Blender, MeshLab and so forth[3]. By using Cabri, Geogebra, Matlab, K3DSurf, MathMod, Mathematica and Maple much of Mathematics object (3D object) could be shown on the desktop just by using Mathematical functions. We are also able to combine some of 3D mathematical objects and draw them together to obtain a new 3D object. And 3D object or 3D models also could be produced becomes the real form as solid objects by using 3D printers.

The concept to create the 3D object is the most important part when we want to make the real form of mathematical functions. We could follow the workflow given in Fig. 1.2[3]:

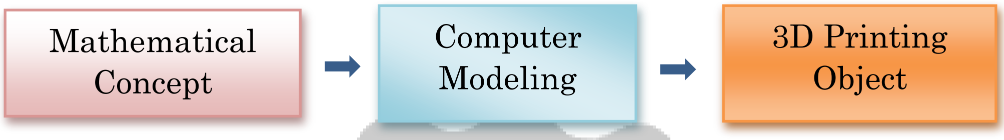


Figure 1.2: Workflow of creating the real form of a mathematical function

Mathematical concept means obtaining the object which want to create such as graph, geometry object and so forth. Computer modeling means by using the computer software to construct the Mathematics object via the GUI (object file's generating method) interface. There are some ways that could be used to create the 3D object, and it depends on the software and the material product. And commonly there are three popular categories to represent a model:

a. *Polygonal modeling*

One of the most popular modeling methods is 3D polygonal modeling. 3D polygonal modeling has a good accuracy during modeling process and a quick meshing process[4]. The vertices are connected by line segments to form a Polygon mesh. Polygonal modeling method is very flexible and the process of computing is so fast. It also makes the computer could render the object so quickly[5].

b. *Curve modeling*

One of the basic subject in computer graphics is curve modeling[6]. The surfaces are defined by curves which could be adjusted by weighted control points. The curve follows the points but does not necessarily interpolate. The curve closer to that point will be pulled by increasing of the weight for a point. And the types of curve include non uniform rational B-spline (NURBS), geometric primitives, splines, and patches.

c. Digital sculpting

3D sculpting is very popular way in the few years for 3D modeling although still a new method of modeling. There are 3 types of digital sculpting such as displacement, dynamic tessellation and volumetric.

- Displacement

  Displacement is the most commonly used to construct a 3D model at this moment. It uses a dense model for constructing and will store the new locations for each vertex position.

- Dynamic tessellation

  This method is similar to Voxel. The different of them is for dividing the surface. Dividing of surface is using the triangulation to maintain a smooth surface and allow finer details. For a very artistic exploration model these method is allowed. This method is usually used to create a game. And for a game engine, the new mesh usually will have the original high resolutions mesh information transferred into displacement data or normal map data.

- Volumetric

  Volumetric has the similar capabilities as displacement. When there are not enough polygons in a region to achieve a deformation, it does not suffer from polygon stretching[7].

3D printing object is the final process to get the solid object. It would be called as the real form of 3D solid object.

The research modeling presented in this thesis belongs to the polygonal and curves modeling. To get the 3D solid objects it would be using a 3D printer engine by the fused deposition modeling technique as discussed in next section.

II

Creation of 3D Objects

In mathematics we know that 3D object is covered by a surface which could be interpreted a surface of 2-variable functions or iso-surface of higher dimension functions. An iso-surface is a 3D analog of an iso-line in a 2D space. It is a surface and represents the points of a constant value within a volume of space. In other words, it is a level set of a continuous function whose domain is a 3D space. 3D space is a geometric three-parameter model. The 3D object could be labeled by a combination of three parameters chosen from the terms height, width, length, depth, and breadth. Provided that the terms width, height, length, depth, and breadth do not all lie in the same plane then any three directions could be chosen. In physics and mathematics, a sequence of n numbers could be understood to identify a specified location of the object in an n-dimensional space. When $n = 3$, the set of all such collection is called as a 3D Euclidean space. It is commonly represented by the symbol $\mathbb{R}^3$ ("3D space")[8].

Now, the object on a 3D space could be constructed by using 3D software. To construct a 3D object we also could be generated by using the mathematical functions which depends on the software that we are using. The popular software to construct the 3D object represented by mathematical functions like: MathMod, K3DSurf, Matlab, Mathematica 10 (Wolfram Mathematica), and Maple. Before we create or construct the 3D models by using these software, the first step to do is to make a mathematics concept of the models (object) which we know as mathematical functions (mathematics equations) and then use the software to compute and plot the graph. On this research we would like to use the mathematical functions to construct the 3D object in the 3D modeling process. Although sometimes is not easy to construct 3D objects by just using mathematical functions, because of some prior knowledge is required in understanding a lot of mathematical functions as the reference. After we finished the 3D modeling process, then the graph or the 3D object displays on computer which also could be able to produce a real form or solid object by using 3D printers.

And in a very modern life has an advancement of technology and its increased very significantly. One of them is 3-dimensional printer usage in daily lives. By combining 3D software and 3D printer we could do a very interesting thing. A lot of companies are doing an intense research, creation and development for 3-dimensional printer machine. The 3D printer has advantages very much for every user. By using a 3D printer machine we could create our favorite 3D object or 3D design model to be a solid object (real form). The commonly materials are used in 3D model printing process like: plastic, powder, resins and other materials. The other materials are like: titanium, stainless steel, bronze, brass, silver, gold, ceramics, chocolate and so forth. There are four popular techniques that are used for printing 3D models such as Fused Deposition Modeling or FDM, Stereo-lithography or SLA, Selective Laser Sintering or SLS, Poly Jet or Ink Jet 3D printing [9-13].

a. *Fused Deposition Modeling (FDM)*

   This technology was commercialized in 1990 and developed by S. Scott Crump in the late 1980s. Fused deposition modeling or FDM is an additive manufacturing technology. This method commonly used for modeling, prototyping, and production applications. One of the popular techniques used for 3D Model printing. FDM works on an "additive" principle by laying down material in layers. A plastic filament or metal wire is released from a coil and supplies material to produce a part[14].

b. *Stereolithography (SLA)*

   Stereolithography is one process in three dimensions (3D) to produce the 3D solid object. There are some synonyms be used for stereolithography like: optical fabrication, photosolidification, desktop manufacturing, additive manufacturing, automatic modeling, solid freeform fabrication, solid imaging, optical shaping, steric polymerization, electron beam melting, digital part materialization, free forming and so forth[15]. Stereolithography is used for creating the models, production parts, prototypes, patterns in a layer by layer used photo-polymerization method. The process is by which light causes chains of molecules to link them together and forming polymers[16].

c. *Selective Laser Sintering (SLS)*

Selective Laser Sintering (SLS) is technique to produce the 3D solid object and also as a technique for additive manufacturing. SLS uses a laser to source the power and sinter powdered material (typically metal). The laser is binding the material together to create a solid structure automatically at points in space according to the shape of the object. SLS is similar to direct metal laser sintering (DMLS). It has the same concept but different in technical details. SLS is a relatively new technology for produce the 3D solid object. And so far SLS has been used mainly for rapid prototyping and for low-volume production of component parts[17].

d. *Poly Jet/ Ink Jet 3D Printing*

Poly jet technology is one technique and powerful additive manufacturing method. Poly jet could produce the smoothly object, accurate prototypes, parts and tooling. The accuracy is as high as 0.1 mm with 16-micron layer resolution. Poly jet 3D printer could produce the thin walls and the complex geometries using the widest range of materials. Poly Jet 3D printing is similar to ink jet printing. This printing technique support printing process by using the materials with the different colors. The printing result would be colorful[18].

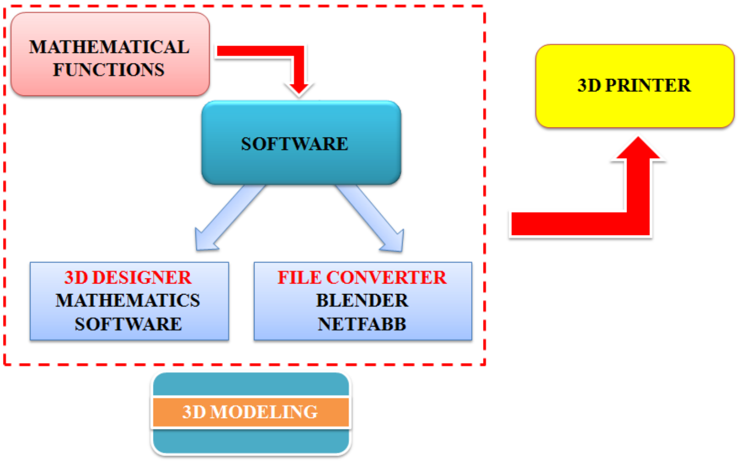


Figure 2.1: 3D modeling process of mathematical function

Commonly 3D printer could read the STL file format to print out the 3D models. STL (Stereolithography) file is one of files format that native to the stereo-lithography CAD (computer-aided design) software. STL file is needed to print the 3D object by 3D printer. And STL file is one of the formats file that 3D printers take as input. STL file is created by 3D Systems[19]. For describing a mesh of triangles the STL file format is a standard format. The file essentially consists of a long list of triangles. Each triangle is given by the coordinates in three-dimensions of its three corners.

Some of 3D modeling software only prepares the OBJ file format. OBJ (or .OBJ) is a geometry definition file format and developed by Wavefront Technologies. OBJ file is advancement for visualizer animation package. The file format is open and adopted by other 3D graphics application vendors for the most part universally accepted format. The OBJ file format is used to represent the simple data format of 3D geometry object like: the position of each vertex, the UV position of each texture coordinate vertex, vertex normal, and the faces. The vertices are stored in a counter clockwise order by default. OBJ files could contain scale information in a human readable comment line although OBJ coordinates have no units[20]. To get the STL file of 3D models we could use the converter file software like Blender or Netfabb.

2.1 MathMod

MathMod is a Mathematics modeling software that visualize and animate implicit and parametric surfaces. MathMod is supported by 3D, 4D plotting and also for animation. OBJ file format is as the output file type of MathMod software. The Scripting language is written in JSON file format. MathMod is also supported by texture and pigmentation, noise and turbulence effects[21].

In order to have a bench mark for comparison between the different software the **Olympic Logo** is chosen as the bench mark object. The Olympic logo consists of five torus and different centers. Each torus has the equation of following this form:

$$\left(\sqrt{(x-a)^2+(y-b)^2}-r_1\right)^2+(z-c)^2-r_2=0 \tag{1}$$

And its center is located at (a, b, c). By constructing 5 of torus equations with different center positions and radius (r_1, r_2) as shown in Table 1, we obtain the Olympic Logo with their equations explicitly described separately by

Table 1. Center (a, b, c) and radius (r_1, r_2) of 5 tori in Olympic logo

No	a	b	c	r_1	r_2
1	7	0	0	3	0.1
2	0	0	0	3	0.1
3	-7	0	0	3	0.1
4	3.5	-3	0	3	0.1
5	-3.5	-3	0	3	0.1

$$\left(\sqrt{(x-7)^2+y^2}-3\right)^2+z^2-0.1=0$$

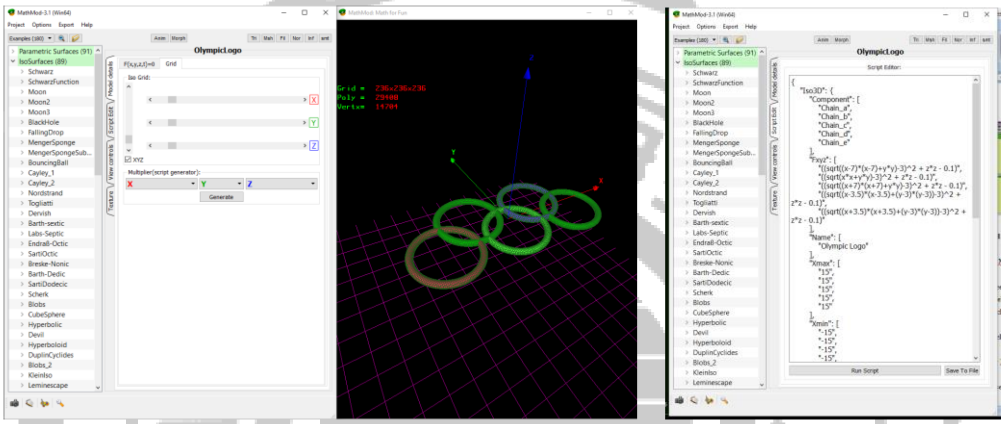
$$\left(\sqrt{x^2+y^2}-3\right)^2+z^2-0.1=0$$

$$\left(\sqrt{(x+7)^2+y^2}-3\right)^2+z^2-0.1=0$$

$$\left(\sqrt{(x-3.5)^2+(y-3)^2}-3\right)^2+z^2-0.1=0$$

$$\left(\sqrt{(x+3.5)^2+(y-3)^2}-3\right)^2+z^2-0.1=0$$

Once the mathematical functions (equations) of the object is obtain, the interested region of the object is determined as $[-15,15] \times [-15,15] \times [-15,15]$. After we did determine for the region of the object where the object is enclosed, it means the preparation for mathematical concept is done, and then we could use MathMod to graph the Olympic Logo. The control panel (for drawing options) of MathMod is shown in Fig. 2 such that we would get the beautiful model (perfect model).



(a) (b)

Figure 2.2: (a) MathMod layout and (b) script editor

The steps to create the 3D object of mathematical function by using MathMod are:

- Run the MathMod program.
- Write the script file in script editor (see Fig. 2.2(b)).
- Run the script file and check the model details property.
- Save the 3D object in the OBJ file format.

Below there are some properties should be check carefully with attention:

a. *Properties Editor*

To get the best performance of model, we have to scroll down the Iso-grid panel be minimum.

"Properties editor >> Grid/ Colors >> Iso-grid"

b. *Control Panel*

On the control panel, make sure that these 4 buttons are on

- Triangles
- Mesh
- Fill
- Smooth

Its purpose is to make sure that the object is surface.

c. *Script File*

We should create the script file correctly and then will get the good result of the 3D object. The script file would be put on the script editor of the MathMod software then run it (as shown in Fig. 2.4 for detail information). The script file of MathMod software should be written correctly then it could be run.

- *Saving Object (OBJ file)*

The software is run to get the 3D model as shown in Fig. 2.3, then save the file into OBJ file by clicking the button on the menu bar:

"Export >> `wavefront.obj` " and type the name of the file.

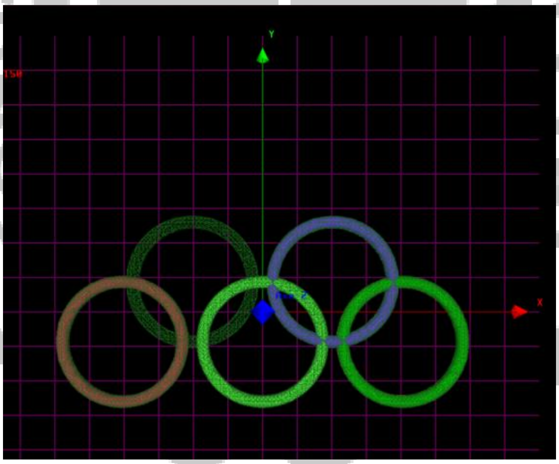


Figure 2.3: 3D modeling view of the Olympic logo

The MathMod software until now only prepares the OBJ file format. To get the STL file we need to use the other software for format conversion. In Sec. 2.6 we would see how to convert the OBJ file into STL file by using the free software Blender and/or Netfabb.

```
{
    "Iso3D": {
        "Component": [
            "Chain_a",
            "Chain_b",
            "Chain_c",
            "Chain_d",
            "Chain_e"
        ],
        "Fxyz": [
            "((sqrt((x-7)*(x-7)+y*y)-3)^2 + z*z - 0.1)",
            "((sqrt(x*x+y*y)-3)^2 + z*z - 0.1)",
            "((sqrt((x+7)*(x+7)+y*y)-3)^2 + z*z - 0.1)",
            "((sqrt((x-3.5)*(x-3.5)+(y-3)*(y-3))-3)^2 + z*z - 0.1)",
            "((sqrt((x+3.5)*(x+3.5)+(y-3)*(y-3))-3)^2 + z*z - 0.1)"
        ],
        "Name": [
            "Olympic Logo"
        ],
        "Xmax": [
            "15",
            "15",
            "15",
            "15",
            "15"
        ],
        "Xmin": [
            "-15",
            "-15",
            "-15",
            "-15",
            "-15"
        ],
        "Ymax": [
            "15",
            "15",
            "15",
            "15",
            "15"
        ],
        "Ymin": [
            "-15",
            "-15",
            "-15",
            "-15",
            "-15"
        ],
        "Zmax": [
            "15",
            "15",
            "15",
            "15",
            "15"
        ],
        "Zmin": [
            "-15",
            "-15",
            "-15",
            "-15",
            "-15"
        ]
    }
}
```

Figure 2.4: MathMod's script file for producing Olympic logo

2.2 Wolfram Mathematica 10

Mathematica 10 is one of commercial mathematical modeling software. Mathematica was released in 1988 for the first time[22]. By using Mathematica 10 we also could create the 3D object and export the file into STL format file immediately. Mathematica 10 is able to construct the 3D object by using mathematical functions. The way to construct the objects or the models is almost same like the previous software in Sec. 2.1. Mathematica 10 would use the mathematical functions and also need to do determining the region of the 3D object.

Steps to create the 3D object by using Mathematica 10:

- Run the program Wolfram Mathematica 10.
- Menu bar >> New file.
- Type the script file into the new file (see Fig. 2.5).
- Run the script file by pressing "Shift + Enter" button.
- The STL file has been created.

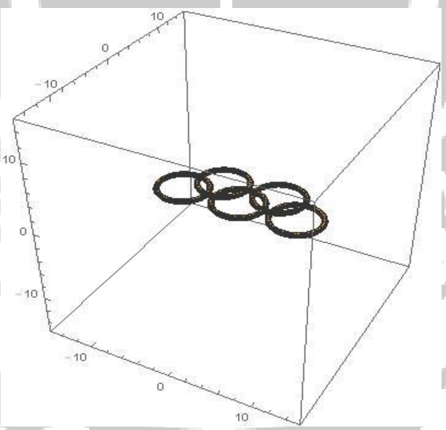


Figure 2.5: 3D model view of Olympic logo via Wolfram Mathematica 10

Although the STL file has been gotten, we also should know the 3D objects or 3D models that constructed by Mathematica 10 does not have a thickness already. The same case like in the Sec. 2.1, we could use the other software to get the easy way to add the thickness of the 3D objects by using Netfabb Basic or Blender software.

```
model1 = ContourPlot3D[(Sqrt[(x - 7) (x - 7) + y y] - 3)^2 + z z -
     0.1 == 0, {x, 0, 16}, {y, -8, 8}, {z, -8, 8}];
model2 = ContourPlot3D[(Sqrt[x*x + y*y] - 3)^2 + z*z - 0.1 ==
    0, {x, -8, 8}, {y, -8, 8}, {z, -8, 8}];
model3 = ContourPlot3D[(Sqrt[(x + 7)*(x + 7) + y*y] - 3)^2 + z*z -
     0.1 == 0, {x, -12, 6}, {y, -8, 8}, {z, -8, 8}];
model4 = ContourPlot3D[(Sqrt[(x - 3.5)*(x - 3.5) + (y - 3)*(y - 3)] -
        3)^2 + z*z - 0.1 == 0, {x, -5, 8}, {y, -5, 8}, {z, -8, 8}];
model5 = ContourPlot3D[(Sqrt[(y - 3)*(y - 3) + (x + 3.5)*(x + 3.5)] -
        3)^2 + z*z - 0.1 == 0, {x, -8, 8}, {y, -8, 8}, {z, -8, 8}];
Show[model1, model2, model3, model4, model5,
 PlotRange -> {{-15, 15}, {-15, 15}, {-15, 15}}]
Model = Show[model1, model2, model3, model4, model5,
   PlotRange -> {{-15, 15}, {-15, 15}, {-15, 15}}];
Export["OL.stl", Model]
```

Figure 2.6: Wolfram Mathematica 10 script file for Olympic logo

2.3 K3DSurf

K3DSurf is mathematical modeling software. The program could visualize and manipulate mathematical models in three, four, five and six dimensions. K3DSurf supports parametric equations and iso-surfaces. K3DSurf uses parametric descriptions of its physical models. The parametric method of representing surfaces/curves uses a function to map some portion of $\mathbb{R}^2$ (the domain) to a patch of the surface in $\mathbb{R}^3$. Because any position in the plane, and thus any position on the surface patch, could be uniquely given by two coordinates, the surface is said to be parameterized by those coordinates. Parametric equations could be either “Implicit” or “Explicit”[23].

```
((sqrt((x-7)*(x-7)+y*y)-3)^2 + z*z - 0.1):
((sqrt(x*x+y*y)-3)^2 + z*z - 0.1):
((sqrt((x+7)*(x+7)+y*y)-3)^2 + z*z - 0.1):
((sqrt((y-3)*(y-3)+(x-3.5)*(x-3.5))-3)^2 + z*z - 0.1):
((sqrt((y-3)*(y-3)+(x+3.5)*(x+3.5))-3)^2 + z*z -0.1):
```

Figure 2.7: K3DSurf software’s script file for Olympic logo

Steps to create the 3D object by using K3DSurf:

- Run the K3DSurf program.
- Fill the mathematical equation into the equation function toolbox.
- Determine the region of the object (X, Y, Z max and min).
- Click "Options: Export/Resolution/Optimisation" to save the file as OBJ file.
- Type the "Example.obj" as the name of the 3D object and then click "edit OBJ". The OBJ file could be found in "C:\Program Files (x86)\K3DSurf-0.6.2\bin".

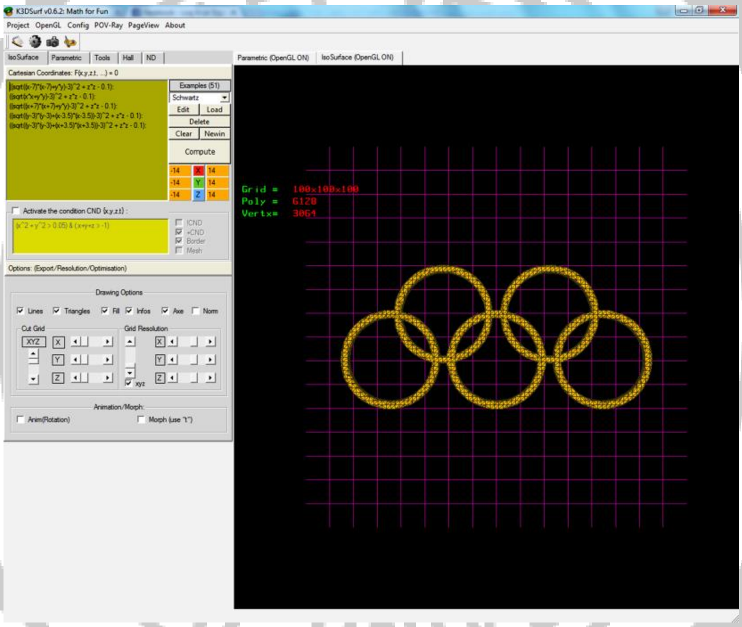


Figure 2.8: Desktop view of the K3DSurf software

By using K3DSurf, we could not get the thickness for all 3D objects immediately. And to have the thickness of the 3D models we should do some modification or combination with another 3D object. We need a trick to add the thickness of the object by using K3DSurf. The same case like MathMod software, by using K3DSurf we are able to export the 3D model into OBJ file. Then we would use the other software to convert the file into STL file after adding some thickness information.

2.4 Maple

Maple is a commercial computer algebra system developed and also able to construct a 3D model. Maple is sold commercially by Maplesoft and the software company is based in Waterloo, Ontario, Canada[24]. Maple software does not only prepare for 3D modeling, but there is a lot of usability of this software. On this paper the 3D object would be constructed by Maple version 18.

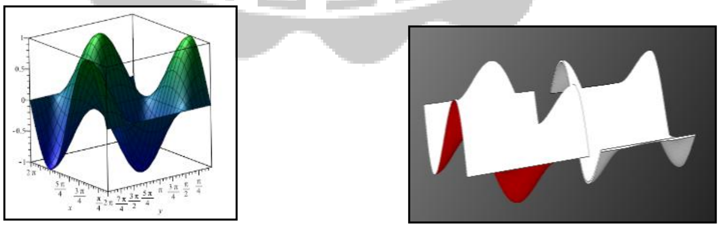
The steps how to create 3D object by using Maple 18 is listed below:

- Run the Maple 18 program and click “New Worksheet”.
- Type the Maple 18 code on the worksheet.
- The STL file would be saved on “C:\Users\School\AppData\Local\Temp”.

```
plot3d(sin(x)*cos(y),
x = 0 .. 2*Pi,
y = 0 .. 2*Pi,
colorscheme = ["MediumBlue", "Green"], thickness = 1);
surface := plot3d(sin(x)*cos(y), x = 0 .. 2*Pi, y = 0 .. 2*Pi,
colorscheme = ["MediumBlue", "Green"], thickness = 1);
TrigonoCurveFile := cat(FileTools:-TemporaryDirectory(),
"/TrigonoCurve.stl");
plottools[exportplot](TrigonoCurveFile, surface);
plottools[importplot](TrigonoCurveFile)
```

Figure 2.9: Maple 18’s script file for Olympic logo

Maple 18 could create the 3D object (Fig. 2.10(a)) with the ability to output in the STL file format. But the STL file is nor really perfect because the thickness of the model is too less (Fig. 2.10(b)). By using Blender software then we would get the perfect models with the good thickness.



(a) (b)

Figure 2.10: (a) 3D model by using maple and (b) 3D model before and after editing by Blender

2.5 Matlab

Matlab is a high-level language and interactive environment software. And millions of engineers and scientists worldwide are using Matlab to do their work. By using Matlab we could explore and visualize ideas and collaboration across disciplines including signal and image processing, control systems, communications, computational finance and so forth[25]. Matlab software is one of the very famous commercial software and very useful for engineers and scientists. Matlab is the software with a proprietary programming language developed by MathWorks. And Matlab allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Excel, Fortran and Python[26].

Matlab is also able to create or construct 3D objects by using the mathematical function. For example, consider a Eistute 3D model:

$$(x^2 + y^2)^3 - 4x^2y^2(z^2 + 1) = 0$$

To create its 3D object we should prepare the script file of the object appropriate to the Matlab language program (see Fig. 2.11), the 3D model view produced by Matlab is then shown in Fig. 2.12. We see from this example that Matlab could export this type of surfaces into STL file format.

```
fun=@(x,y,z)((x.^2+y.^2).^3-4.*x.^2.*y.^2.*(z.^2+1)) ;
[X,Y,Z]=meshgrid(-2:0.1:2,-2:0.1:2,-2:0.1:2);
val=fun(X,Y,Z);
fv=isosurface(X,Y,Z,val,0);
p = patch(fv);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'Green');
set(p,'EdgeColor' , 'none');
daspect([1,1,1]);
view(3);
camlight
lighting phong
axis on
stlwrite(‘Eistute.stl’,fv)
```

Figure 2.11: Matlab script file for the Estute 3D model

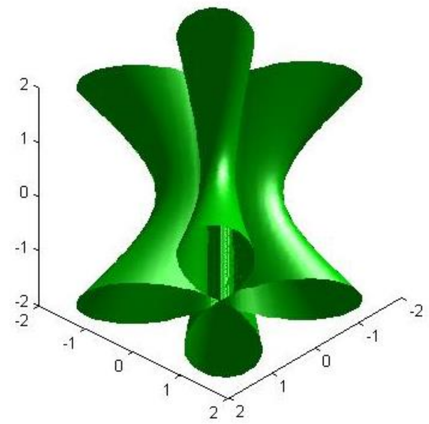


Figure 2.12: Eistute 3D model

Back to model the Olympic logo for comparison between software, the 3D model view of Olympic logo is shown in Fig. 2.13 with script file given in Fig. 2.14.

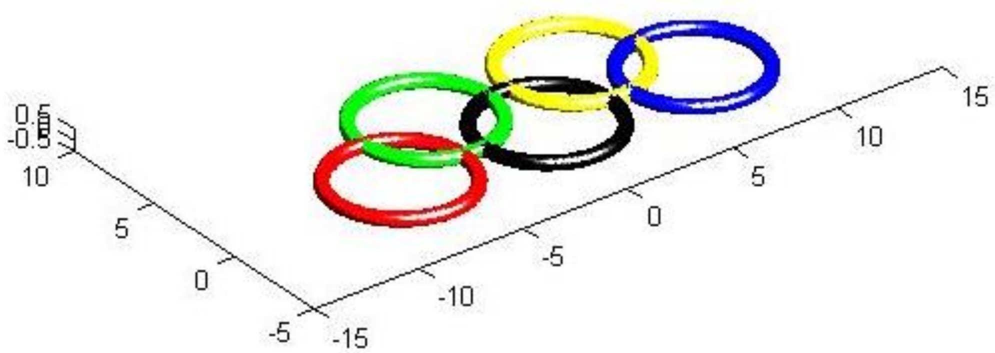


Figure 2.13: 3D model view of Olympic logo via Matlab

```
fun1=@(x,y,z)((sqrt((x - 7).^2 + y.^2) - 3).^2 + z.^2 - 0.1) ;
[X,Y,Z]=meshgrid(-15:0.1:15,-15:0.1:15,-15:0.1:15);
val=fun1(X,Y,Z);
f1=isosurface(X,Y,Z,val,0);
p = patch(f1);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'blue');
set(p,'EdgeColor' , 'none');
daspect([1,1,1]);
view(3);
camlight
lighting phong
axis on
hold on
fun2=@(x,y,z)((sqrt(x.^2 + y.^2) - 3).^2 + z.^2 - 0.1) ;
[X,Y,Z]=meshgrid(-15:0.1:15,-15:0.1:15,-15:0.1:15);
val=fun2(X,Y,Z);
f2=isosurface(X,Y,Z,val,0);
p = patch(f2);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'black');
set(p,'EdgeColor' , 'none');
daspect([1,1,1]);
view(3);
camlight
lighting phong
axis on
hold on
fun3=@(x,y,z)((sqrt((x + 7).^2 + y.^2) - 3).^2 + z.^2 - 0.1) ;
[X,Y,Z]=meshgrid(-15:0.1:15,-15:0.1:15,-15:0.1:15);
val=fun3(X,Y,Z);
f3=isosurface(X,Y,Z,val,0);
p = patch(f3);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'red');
set(p,'EdgeColor' , 'none');
daspect([1,1,1]);
view(3);
camlight
lighting phong
axis on
hold on
```

Figure 2.14(a): Olympic logo Matlab script file-Part 1

```
fun4=@(x,y,z)((sqrt((x - 3.5).^2 + (y - 3).^2) - 3).^2 + z.^2 - 0.1) ;
[X,Y,Z]=meshgrid(-15:0.1:15,-15:0.1:15,-15:0.1:15);
val=fun4(X,Y,Z);
f4=isosurface(X,Y,Z,val,0);
p = patch(f4);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'yellow');
set(p,'EdgeColor' , 'none');
daspect([1,1,1]);
view(3);
camlight
lighting phong
axis on
fun5=@(x,y,z)((sqrt((x + 3.5).^2+(y - 3).^2) - 3).^2 + z.^2 - 0.1) ;
[X,Y,Z]=meshgrid(-15:0.1:15,-15:0.1:15,-15:0.1:15);
val=fun5(X,Y,Z);
f5=isosurface(X,Y,Z,val,0);
p = patch(f5);
isonormals(X,Y,Z,val,p)
set(p,'FaceColor' , 'green');
set(p,'EdgeColor' , 'none');
daspect([1,1,1]);
view(3);
camlight
lighting phong
axis on
```

Figure 2.14(b): Olympic logo Matlab script file-Part 2

2.6 Converting OBJ File into STL File

To print the object by using 3D printers, we should have the STL file format of the object. OBJ file should be converted into the STL file format. We could convert OBJ files into STL files by using free software or the online converter on internet. In this thesis we would use the freeware Netfabb Basic and Blender to convert the file.

a. Netfabb Basic

Netfabb Basic[27] is 3D printing software for handling of files in STL format. Netfabb Basic could be used as a mesh viewer includes an automatic repair and utilizes. By using the Netfabb Basic, we are able to convert the OBJ file into the STL file format. We could follow the procedure to convert the OBJ file into STL file by using Netfabb Basic software as listed below.

- Run the Netfabb Basic.
- Open file (OBJ file).
- Extras >> Repair Part >> Automatic Repair >> Default Repair >> Execute >> Apply Repair >> Remove Old Part >> Right click on the object >> Export Part >> as STL >> Save.
  (File will be saved on the computer).

STL file was ready saved on the computer and ready to use for printing out the object by 3D Printer.

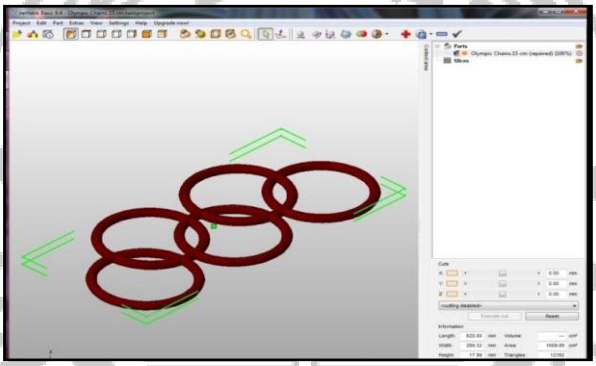


Figure 2.15: Netfabb basic view of Olympic logo in the STL file format

b. Blender

Blender[28] is the free and open source 3D modeling software. Blender software is also able to be a 3D editor. By using Blender, the OBJ file could be converted into the STL file with its layout as shown in Fig. 2.16. To covert the file we are allowed to follow these steps:

- Run the Blender software.
- Import the OBJ file from the computer.
- Right click on the object.
- Click “Setting >> Add modifier >> Solidify”.
- Set the thickness of the object into the “Thickness box” then hit “Enter” and “Apply”.

- Export file into the STL file. And the STL file will be saved on the computer.

Figure 2.16: Blender software layout

By using blender software we are able to add the thickness of our 3D object as much as we want and also use easily.

2.7 3D Object Printing Process

To print out the real solid object by using 3D printers we have to pay attention before we print the object. The steps for printing depend on what kind of 3D printers we use. In this thesis, we select UP 3D printer version 2.18[29] as our machine for solid object output.

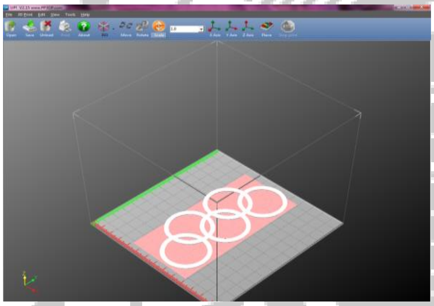
Figure 2.17: UP3D printer

By using the UP 3D printer we could follow the simple steps below:

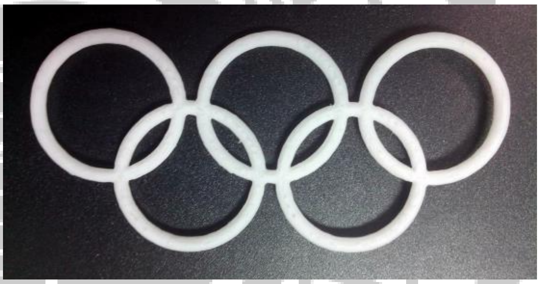
- Run the UP printer program.
- Open the STL file from menu bar (file).
- Edit >> Fix ("sometimes we must not do fix") >> Place.
- Scale the object as big as we want.

  (Sometimes the object will be so large or bigger then we have to scale the object such that the object is on the box layout).
- Print the object.

Figure 2.18 shows the 3D printer software layout and the real solid object printed by UP 3D printer.



(a)



(b)

Figure2.18: (a) Olympic logo is ready to print, and (b) printed 3D solid object

III

3D Objects of Multivariable Functions

3.1 Two-variable Functions

As well we know $z = f(x, y)$ is a surface where $f(x, y)$ is called a function of two variables. The function f of the two variables x and y is a rule that assigns a real number $f(x, y)$ to each pair (x, y) in a portion or all of the xy-plane which is called as domain. The range of the function is the set of collection of its values $f(x, y)$ for all (x, y) in its domain. The graph of a function f with the two variables x and y is a surface $z = f(x, y)$ formed by the points (x, y, z) in xyz-space with (x, y) in its domain[30]. For example $z = f(x, y) = 2xy - \left(\frac{1}{2}\right)(x^4 + y^4) + 1$ is a two-variable function defined on the whole $\mathbb{R}^2$-plane and its range inside $(-\infty, 5)$. The 3D model view is shown in Fig. 3.1.

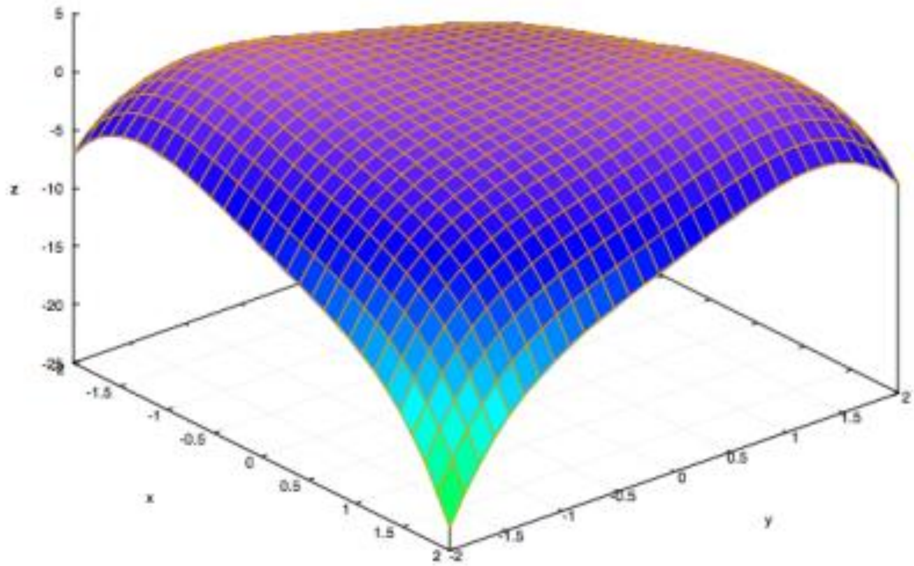
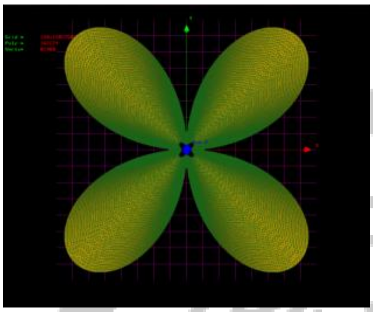


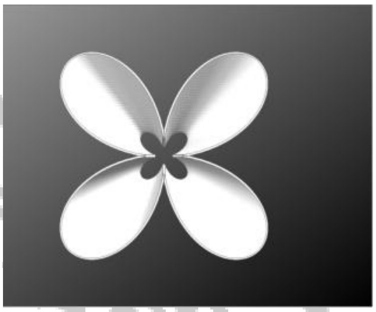
Figure 3.1: The surface corresponding to $z = f(x, y) = 2xy - (x^4 + y^4)/2 + 1$

3.2 Three-variable Implicit Functions

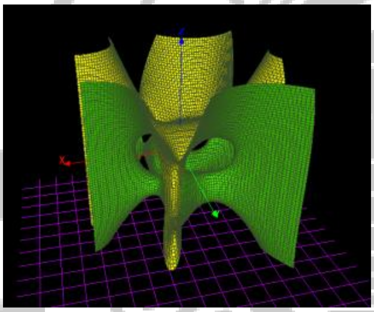
Eleven 3D objects are investigated in this thesis. The corresponding functions, MathMod software display, and the pictures of the real solid objects are listed below.

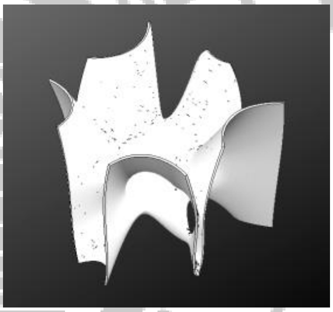
- **Eistute :** $(x^2 + y^2)^3 - 4x^2y^2(z^2 + 1)$



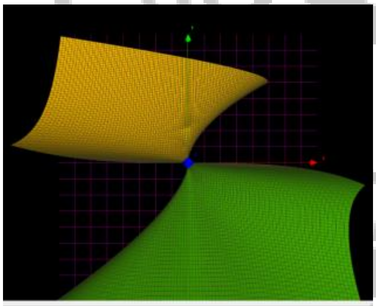


- **Quintik :** $x^5 - 10x^2y^2 + 5xy^4 - 3z^2 - 5y^4 + 10z^3 + 20z^2 + 20y^2 - 15z - 24$

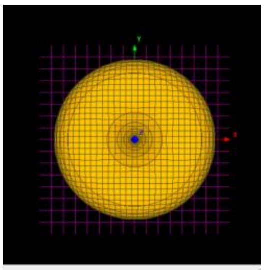




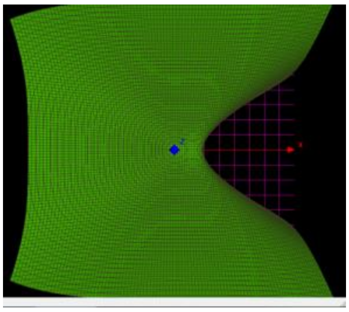
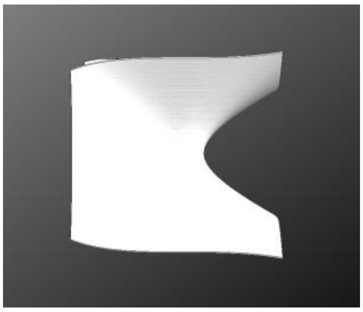
- **DSchneeflocke :** $x^3 + y^2z^3 + yz^4$



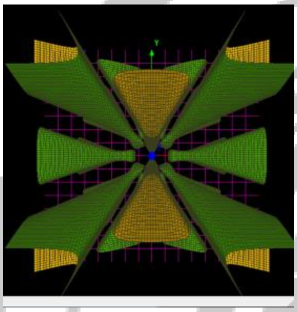
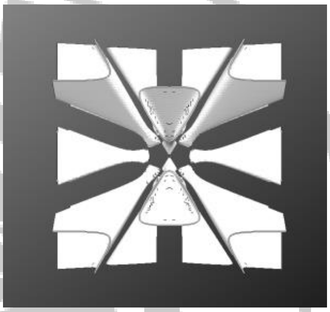
- **Dullo :** $(x^2 + y^2 + z^2)^2 - (x^2 + y^2)$



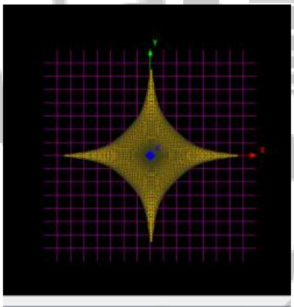
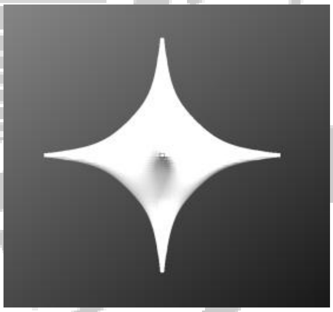
- **Vis a vis :** $x^2 - x^3 + y^2 + y^4 + z^3 - z^4$

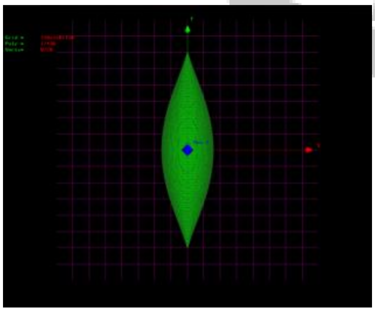
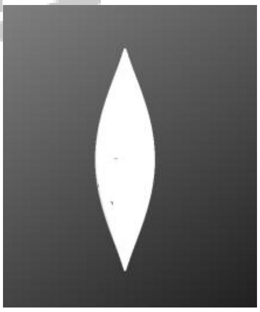
- **Barth Sextik :** $\left((1+\sqrt{5})^2 x^2 - 4y^2\right)\left((1+\sqrt{5})^2 y^2 - 4z^2\right)\left((1+\sqrt{5})^2 z^2 - 4x^2\right) - 10(2+\sqrt{5})(x^2+y^2+z^2-1)^2$

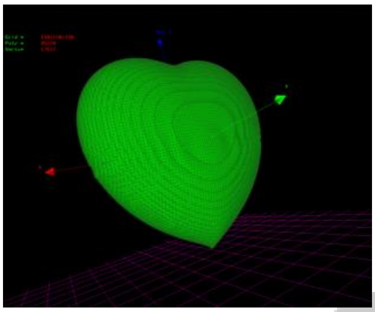
- **Distel :** $x^2 + y^2 + z^2 + 1500(x^2+y^2)(x^2+z^2)(y^2+z^2) - 1$

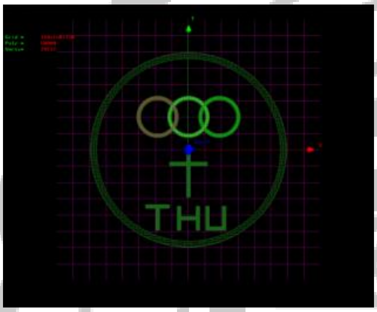
- **Zitrus :** $x^2 + z^2 - y^3(1-y)^3$

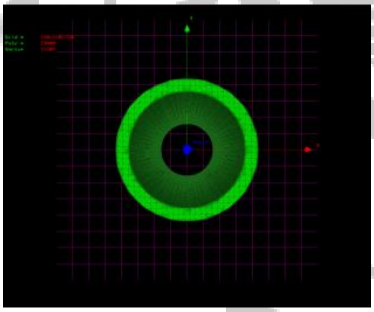
- **Suss :** $\left(x^2+2\left(\frac{9}{4}\right)y^2+z^2-6\right)^3-x^2z^3-2\left(\frac{9}{80}\right)y^2z^3$



- **THU Medal -** (see Appendix I-Part K)



- **Tube -** (See Appendix I-Part J)



3.3 3D Projection of Symmetrized Bidisc

A geometric space with four dimensions is called four dimensional Euclidean space and more specifically four dimensional space ("4D space"). The rules of three dimensional Euclidean space are generalized by it. Over two centuries mathematicians and philosophers had been studied for it, both for its own interest and for the insights it offered into Mathematics and related fields.

Algebraically, it is generated by applying the rules of vectors and coordinate geometry to a space with four dimensions. In particular, a vector with four elements (a 4-tuple) could be used to represent a position in a four dimensional space. The space is a Euclidean space has a metric and norm, and so all directions are treated as the same: the additional dimension is indistinguishable from the other three[31].

Consider a 2×2 matrix W

$$W \sim \begin{bmatrix} 0 & 1 \\ -p & s \end{bmatrix}$$

with its characteristic equation $\lambda^2 - s\lambda + p = 0$, where $\sim$ denotes "similar to". Thus for any given 2×2 matrix, we can calculate its eigenvalue by solving its characteristic equation. Alternatively, suppose we want to find the class of matrices whose eigenvalue is located inside open unit disk $\mathbb{D} = \{\lambda \in \mathbb{C}: |\lambda| < 1\}$, the collection of coefficients of the characteristic equation (s, p) is called the symmetrized bidisc Γ_2:

$$\Gamma_2 = \{(s, p) \in \mathbb{C}^2: \lambda^2 - s\lambda + p = 0, |\lambda| \leq 1\}$$

This 4-D space appears when solving so-called Spectral Nevanlinna-Pick Interpolation problem in the μ-algorithm for robust controller design [32-35]. In this section we select how to find the real form of this space as our typical example to demonstrate the projection of the 4D space into the 3D space.

First of all, we consider a special case with the projection of Γ_2 into $\mathbb{R}^2$, i.e., $\Gamma_2 \cap \mathbb{R}^2$, whose graph is shown in Fig. 2. The characterization of the Γ_2 is described by the following theorem:

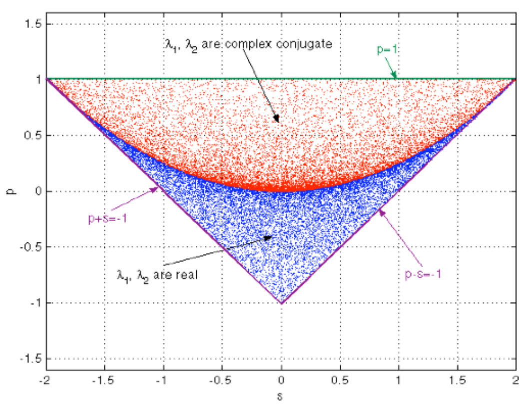


Figure 3.2: The projection of projection of Γ_2 into $\mathbb{R}^2$

Theorem 3.1:

For $s, p \in \mathbb{C}$, the following conditions are equivalent:

(1) $(s, p) \in \Gamma_2$,

(2) $|s| \leq 2$ and $\forall \lambda \in \mathbb{D}, \left|\frac{2\lambda p - s}{2 - \lambda s}\right| \leq 1$,

(3) $|4p - s^2| + |2(s - \bar{s}p)| \leq 4 - |s|^2$.

Suppose we are given two complex numbers $\lambda_1, \lambda_2 \in \mathbb{D}$, then define $s = \lambda_1 + \lambda_2$ and $p = \lambda_1 \cdot \lambda_2$ and it is clearly that $p \in \mathbb{D}$, $s \in 2\mathbb{D}$, and $(s, p) \in \operatorname{int} \Gamma_2$ where int denotes the interior. Before construct the corresponding geometry, we need to find the boundary first. From condition (3) of **Theorem 3.1**, the boundary of Γ_2 is described by

$$|4p - s^2| + |2(s - p\bar{s})| = 4 - |s|^2$$

i.e.,

$$|4p - s^2| + |2(s - p\bar{s})| + |s|^2 = 4.$$

To realize, let $s = x + iy$ and $p = w + iz$ with $w, x, y, z \in \mathbb{R}$, and substitute into above equation with the following operation

$$|4p - s^2| = \sqrt{(4w - x^2 + y^2)^2 + (4z - 2xy)^2}$$

$$|2(s - p\bar{s})| = 2\sqrt{(x - wx - yz)^2 + (y + wy - xz)^2}$$

$$|s|^2 = x^2 + y^2$$

Then the equation $|4p - s^2| + |2(s - p\bar{s})| = 4 - |s|^2$ becomes

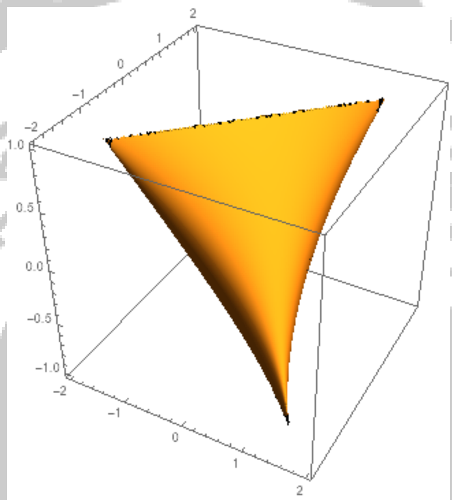
$$F(w, x, y, z) =$$
$$\sqrt{(4w - x^2 + y^2)^2 + (4y - xy)^2} +$$
$$2\sqrt{(x - wx - yz)^2 + (y + wy - xz)^2} + x^2 + y^2 - 4 = 0$$

that is, the boundary of Γ_2 is a level set $F(w, x, y, z) = 0$ of the function $F(w, x, y, z)$ which is a subset of a 4D space. In order to present in 3D space, we need to fix one of the variable as a parameter, for example, when $w = 0$, then this level set can be constructed by using the command `ContourPlot3D` of Mathematica 10.0 as shown in Fig. 3.3. Since as mentioned earlier that $p \in \mathbb{D}$ and

$s \in 2\mathbb{D}$, the ranges of the variables are then defined as $-2 \leq x \leq 2, -2 \leq y \leq 2$ and $-1 \leq z \leq 1$.

```
w = 0.0;
Object1 =
 ContourPlot3D[Sqrt[(x^2 - y^2 - 4 w)^2 + (2 x y - 4 z)^2]+
 2 Sqrt[(x - x w - y z)^2 + (y + y w - x z)^2] + x^2 + y^2 ==
  4, {x, -2, 2}, {y, -2, 2}, {z, -1, 1}, Mesh -> None,
 PlotPoints -> 20]
```

(a)



(b)

Figure 3.3: The boundary of Γ_2 with $w = 0$ via Mathematica. (a) Mathematica code, (b) Mathematica output graph.

In order to be sent to 3D printer, we need to export the graph to STL file by using **`Export`** command inside Mathematica. Thus the code for very parameter value w is listed in Fig. 3.4.

```
w = 0.0;
Object1 =
 ContourPlot3D[Sqrt[(x^2 - y^2 - 4 w)^2 + (2 x y - 4 z)^2]+
 2 Sqrt[(x - x w - y z)^2 + (y + y w - x z)^2] + x^2 + y^2 ==
  4, {x, -2, 2}, {y, -2, 2}, {z, -1, 1}, Mesh -> None,
 PlotPoints -> 20]
Export["Gamma Rew0.stl", Object1]
```

Figure 3.4: Mathematica code to generate the boundary of Γ_2 and output STL file

Since the projection of the boundary of Γ_2 depends on the value of w, we consider it as a parameter with values in $[-1,1]$ and using the command **`Table`** to adjust this parameter as shown in Fig. 3.5 with the output graph.

```
Table[ContourPlot3D[
 Sqrt[(x^2 - y^2 - 4 w)^2 + (2 x y - 4 z)^2]+
 2 Sqrt[(x - x w - y z)^2 + (y + y w - x z)^2] + x^2 + y^2 ==
 4, {x, -2, 2}, {y, -2, 2}, {z, -1, 1}, Mesh -> None,
 AxesLabel -> {Re s, Im s, Im p}, PlotPoints -> 5,
 PlotLabel -> Re p == w], {w, -1, 1, 0.2}]
```

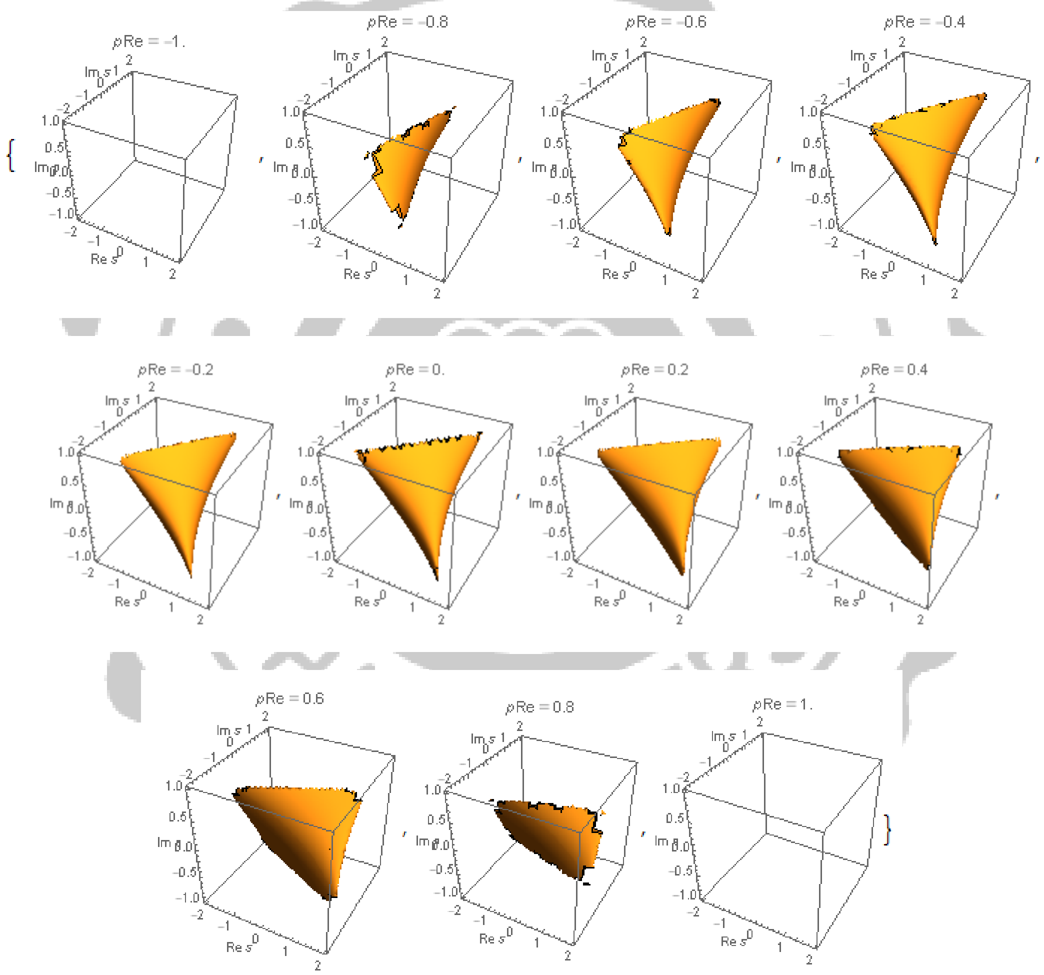


Figure 3.5: The graph of the boundary of Γ_2 with variation in Re $p = w$

From Fig. 3.5, the graph is symmetrized about w. To see the effect of the variation of other parameters, Figs. 3.6–3.8 depict the variation of parameters $z \in [-1,1]$, $x \in [-2,2]$, and $y \in [-2,2]$.

```
Table[ContourPlot3D[
  2 Sqrt[(x - x w - y z)^2 + (y + y w - x z)^2] +
    Sqrt[(x^2 - y^2 - 4 w)^2 + (2 x y - 4 z)^2] + x^2 + y^2 ==
   4, {x, -2, 2}, {y, -2, 2}, {w, -1, 1}, Mesh -> None,
  PlotPoints -> 5, AxesLabel -> {Re s, Im s, Re p},
  PlotLabel -> Im p == z], {z, -1, 1, 0.2}]
```

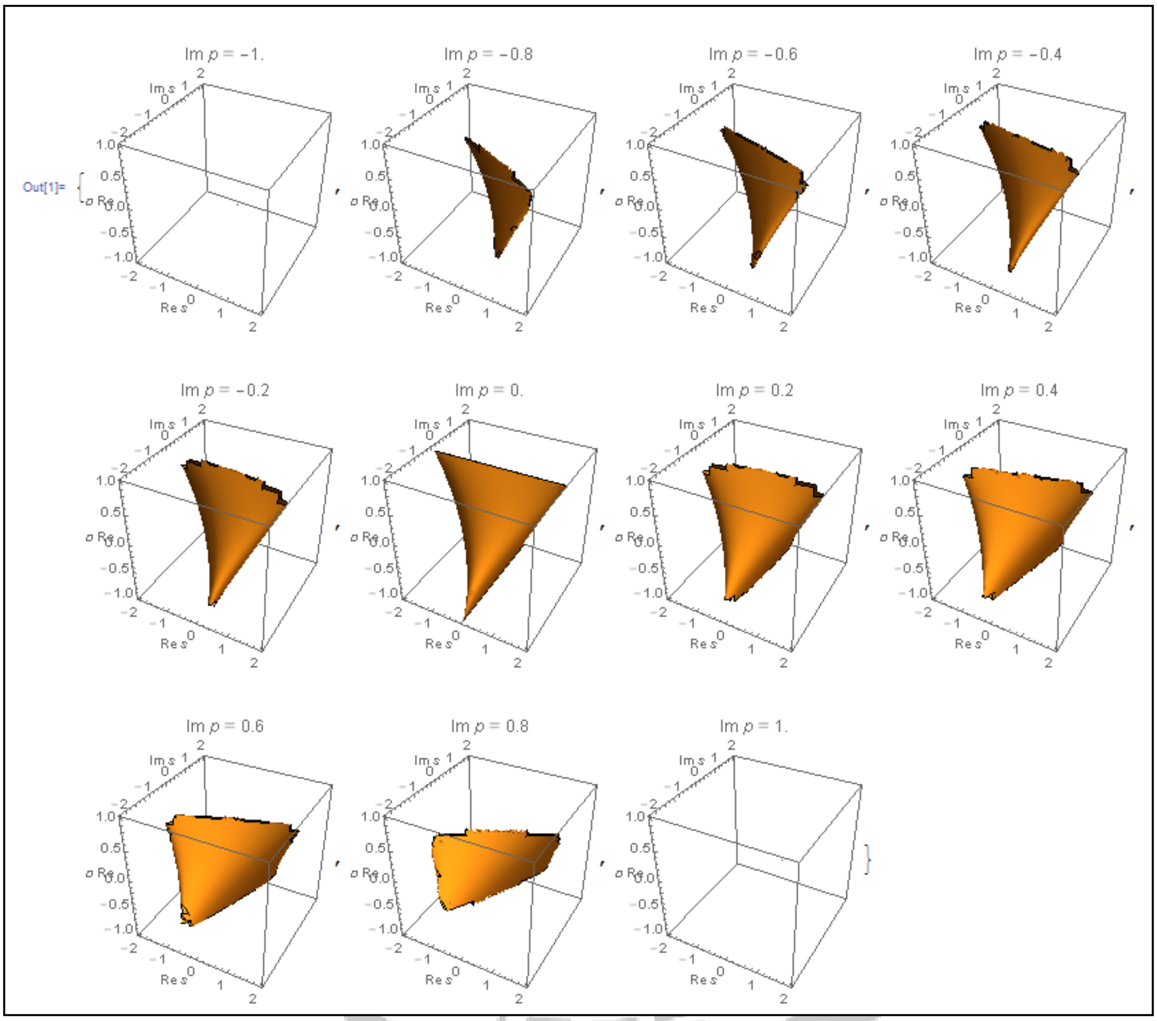


Figure 3.6: The graph of the boundary of Γ_2 with variation in Im $p = z$

```
Table[ContourPlot3D[
  2 Sqrt[(x - x w - y z)^2 + (y + y w - x z)^2] +
    Sqrt[(x^2 - y^2 - 4 w)^2 + (2 x y - 4 z)^2] + x^2 + y^2 ==
   4, {w, -1, 1}, {z, -1, 1}, {y, -2, 2}, Mesh -> None,
  PlotPoints -> 5, AxesLabel -> {Re p, Im p, Im s},
  PlotLabel -> Re s == x], {x, -2, 2, 0.2}]
```

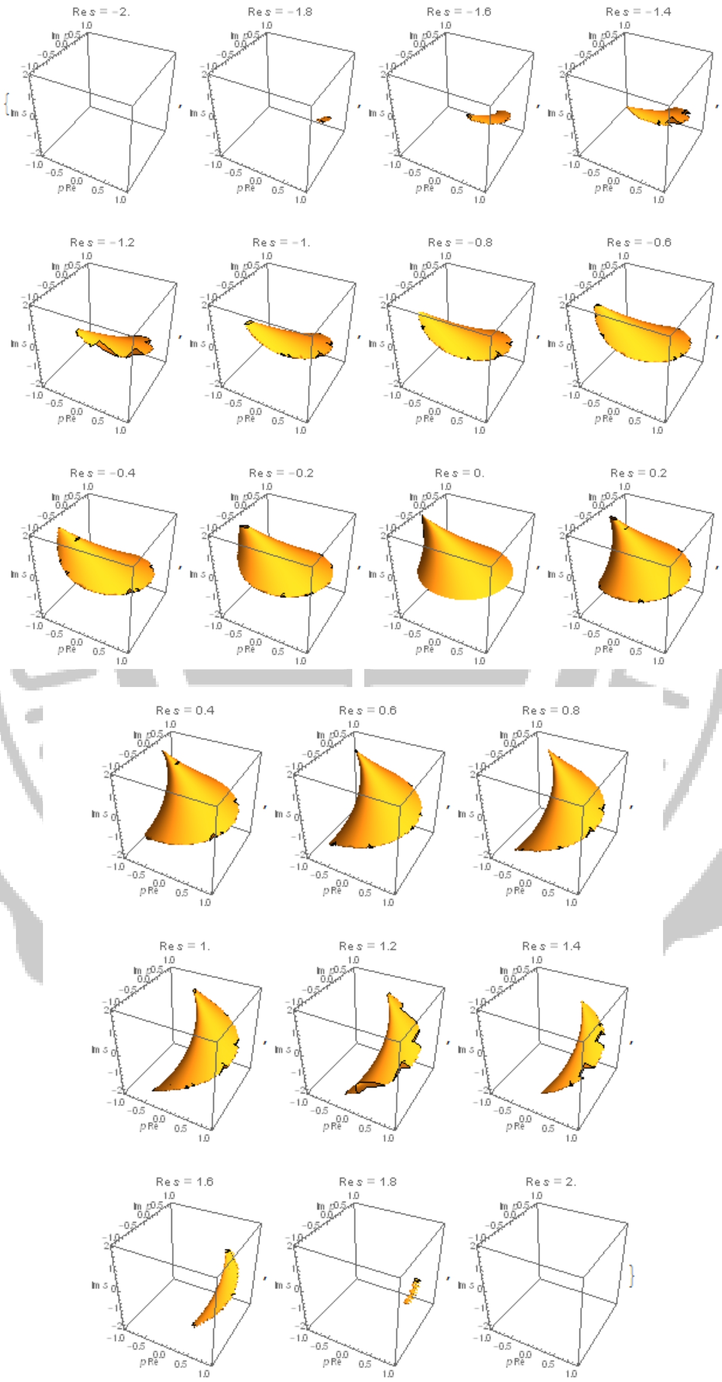


Figure 3.7: The graph of the boundary of Γ_2 with variation in Re $s = x$

```
Table[ContourPlot3D[
  2 Sqrt[(x - x w - y z)^2 + (y + y w - x z)^2] +
    Sqrt[(x^2 - y^2 - 4 w)^2 + (2 x y - 4 z)^2] + x^2 + y^2 ==
   4, {w, -1, 1}, {z, -1, 1}, {x, -2, 2}, Mesh -> None,
  PlotPoints -> 5, AxesLabel -> {Re p, Im p, Re s},
  PlotLabel -> Im s == y], {y, -2, 2, 0.2}]
```

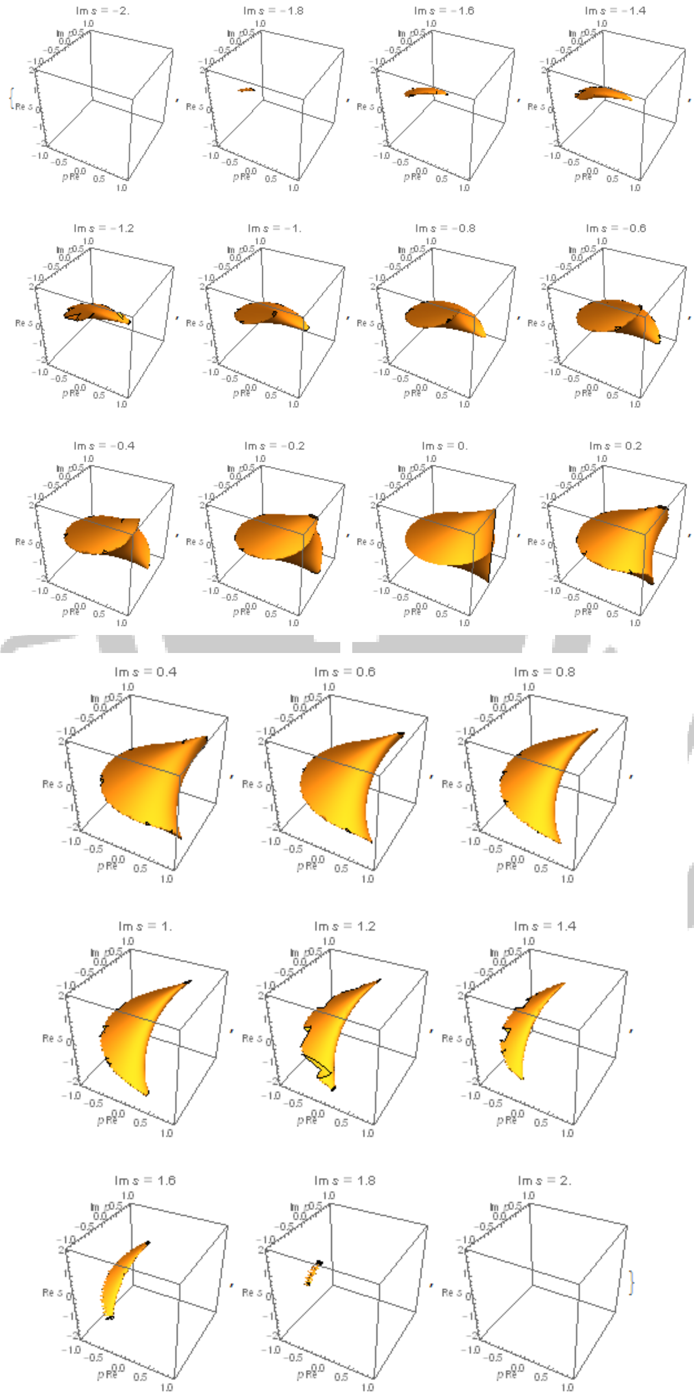


Figure 3.8: The graph of the boundary of Γ_2 with variation in Im $s = y$

IV

Discussion and Conclusion

4.1 Discussion

The real form creation of mathematical functions by using software and 3D printer is an exited thing. Even though sometime the encounter difficulties certain. There are some important things that we need to know on creation of the real form or real solid object of mathematical functions. One of them is the selection of the software that we want to use and the specifications of our computers to run the software. The other is the result of the computer software after we did 3D modeling. To print the 3D model we should have the STL file format. Not for all 3D software prepares the STL file of 3D models. STL file is very important thing to have while producing the real form of 3D objects or 3D models. To get the STL file we could use another 3D software converter. The table below will show the difference of some Mathematics 3D software.

Table 1: Comparison between mathematical software

Software	MathMod	Mathematica	K3DSurf	Maple Soft	MATLAB
OBJ File	√	√	√	√	√
STL File	–	√	–	√	√
Thickness	√	√	√	–	–
File Converter	Need	No	Need	No	No
Script File Code	Need	Need	No	Need	Need
Thickness Addition	Blender	Blender (Sometimes)	Blender	Blender	Blender

After STL file format is satisfied, the other thing that we should need to have is the thickness of the 3D object. Most of 3D printers are only able to print the object with the thickness more than 0.15 mm (> 0.15 mm). Converter software as like Blender is able to add the thickness of 3D models. This is the easy way to add the thickness of the models.

4.2 Conclusion

Mathematics is the knowledge that very interesting to learn, although sometimes many difficulties to be faced. It is a challenge and pleasure that we will feel while studying. Before the computer and software were invented and used as a media in the teaching of Mathematics, Mathematics is very-very abstract. Time is going well and the development of technology is growing so fast then the usage of computers is very helpful to understand Mathematics. By combination of computers and 3D printers, Mathematics becomes more real and more interesting.

In math we are familiar to mathematical functions. Programmers had been found supporting software that could illustrate or describe the Mathematical function mainly on the material geometry. Software and 3D Printer are very perfect combination to bring Mathematics becomes real and useful. Software and 3D Printer also could help to release the assumption that Mathematics is something boring, but rather something real and interesting.

By using software and 3D printers, it is expected that the learning of Mathematics to be more attractive and real. Mathematical functions are not only as variables and numbers but become the real object that could be seen by eyes. Software and 3D printers also could be use not only at school but also in daily life or for daily needs.

References

[1] Mathematics as “The Queen of the Sciences”, https://en.wikipedia.org/wiki/Mathematics

[2] Oliver Knill and Elizabeth Slavkovsky, Illustrating Mathematics Using 3D Printers

[3] Henry Segerman, 3D printing for Mathematical visualization

[4] Zheng Li, Jeremy.(2015). CAD, 3D Modeling, Engineering Analysis, and Prototype Experimentation. Springer Cham Heidelberg. New York

[5] Russo, Mario.(2006). Polygonal Modeling: Basic and Advanced Techniques. Word Publishing Inc. USA

[6] Ahmet Cinar, Fuzzy Object Blending n 2-D. Computational Science - ICCS 2001: International Conference, San Francisco

[7] 3D Modeling, https://en.wikipedia.org/wiki/3D_modeling

[8] William E. Lorensen, Harvey E. Cline: Marching Cubes: A high resolution 3D surface construction algorithm

[9] 3D Printer Materials, http://www.think3d.in/raw-materials-for-3d-printing/

[10] Fused Deposition Modeling (FDM), https://www.youtube.com/watch?v=WHO6G67GJbM

[11] Stereo-lithography (SLA), https://www.youtube.com/watch?v=NM55ct5KwiI

[12] Selective Laser Sintering (SLS), https://www.youtube.com/watch?v=9E5MfBAV_tA

[13] Polyjet/ink jet 3D printing, https://www.youtube.com/watch?v=Som3CddHfZE

[14] FDM – Fused Deposition Modeling, https://en.wikipedia.org/wiki/Fused_deposition_modeling#cite_note-1

[15] SLA, http://www.photopolymer.com/stereolithography.htm

[16] SLA – Stereolithography,

https://en.wikipedia.org/wiki/Stereolithography#cite_note-Patent-1

[17] SLS – Selective Laser Sintering, https://en.wikipedia.org/wiki/Selective_laser_sintering#cite_note-1

[18] Poly jet/ Ink jet 3D Printing,http://www.stratasys.com/3d-printers/technologies/polyjet-technology

[19] STL file, https://en.wikipedia.org/wiki/STL_(file_format)

[20] OBJ File, https://en.wikipedia.org/wiki/Wavefront_.obj_file

[21] MathMod, Abderrahman Taha, http://sourceforge.net/projects/mathmod/

[22] Wolfram Mathematic , http://www.wolframalpha.com/about.html?_ga=1.255511560.1409171756.1442894385

[23] K3DSurf, Abderrahman Taha, http://k3dsurf.sourceforge.net/

[24] Maple, https://en.wikipedia.org/wiki/Maple_(software)

[25] MATLAB, http://www.mathworks.com/products/matlab/

[26] MATLAB, https://en.wikipedia.org/wiki/MATLAB

[27] Netfabb Basic, http://www.netfabb.com/downloadcenter.php?basic=1

[28] Blender Foundation, Blender, http://www.blender.org/

[29] 3D Printer Tutorial, UP 3D Printer, http://www.pp3dp.com/

[30] Dinnen Sean.(2000). Function of Two Variables (2nd Ed). Chapman and Hall/ CRC Mathematics

[31] Four Dimensional Space (4D), https://en.wikipedia.org/wiki/Fourdimensional_space

[32] J. Agler and N.J. Young, The two-point spectral Nevalinna-Pick problem, *Integral Equations Operator Theory* **37** (2000) 375-385.

[33] J. Agler and Nicholas J. Young. (2004). The hyperbolic geometry of the symmetrized bidisc. Appeared in the Journal of Geometric Analysis, Vol 14, pp. 375-403.

[34] J. Agler and Nicholas J. Young. (2004). The two-by-two spectral Nevanlinna-Pick problem. Transactions of the American Mathematical Society, Vol.356, pp. 573-585.

[35] C.M. Chen, The Graphs of Symmetrized Bidiscs and Spectral Interpolating Functions, Master Thesis, Department of Mathematics, Tunghai University, Taiwan, July 2010.

[36] Type of STL file, https://en.wikipedia.org/wiki/STL_(file_format)

[37] STL Binary Format, http://www.fabbers.com/tech/STL_Format

[38] ABS Printing Material, https://en.wikipedia.org/wiki/Acrylonitrile_butadiene_styrene

[39] PLA Printing Material, https://en.wikipedia.org/wiki/Polylactic_acid

Appendix I

This appendix collects all MathMod Script files for 3D objects in Sec. 3.2.

A. Eistute

```
{
    "Iso3D": {
        "Description": [
            "Eistute"
        ],
        "Fxyz": [
            "(x^2+y^2)^3-4*(x^2)*(y^2)*(z^2+1)"
        ],
        "Name": [
            "Eistute"
        ],
        "Xmax": [
            "5"
        ],
        "Xmin": [
            "-5"
        ],
        "Ymax": [
            "5"
        ],
        "Ymin": [
            "-5"
        ],
        "Zmax": [
            "5"
        ],
        "Zmin": [
            "-5"
        ]
    }
}
```

B. Quintik

```
{
    "Iso3D": {
        "Description": [
            "Quintik"
        ],
        "Fxyz": [
            "x^5-(10*x^3)*y^2+5*x*y^4-3*z^2-
(10*x^2)*y^2-5*y^4+10*z^3+20*z^2+20*y^2-15*z-24"
        ],
        "Name": [
            "Quintik"
        ],
        "Xmax": [
            "4"
        ],
        "Xmin": [
            "-4"
        ],
        "Ymax": [
            "4"
        ],
        "Ymin": [
            "-4"
        ],
        "Zmax": [
            "4"
        ],
        "Zmin": [
            "-4"
        ]
    }
}
```

C. DSchneeflocke


```
{
    "Iso3D": {
        "Description": [
            "DSchneeflocke"
        ],
        "Fxyz": [
            "x^3+(y^2)*z^3+y*z^4"
        ],
        "Name": [
            "DSchneeflocke"
        ],
        "Xmax": [
            "4"
        ],
        "Xmin": [
            "-4"
        ],
        "Ymax": [
            "4"
        ],
        "Ymin": [
            "-4"
        ],
        "Zmax": [
            "4"
        ],
        "Zmin": [
            "-4"
        ]
    }
}
```


D. Dullo


```
{
    "Iso3D": {
        "Description": [
            "Dullo"
        ],
        "Fxyz": [
            "(x^2+y^2+z^2)^2-(x^2+y^2)"
        ],
        "Name": [
            "Dullo"
        ],
        "Xmax": [
            "4"
        ],
        "Xmin": [
            "-4"
        ],
        "Ymax": [
            "4"
        ],
        "Ymin": [
            "-4"
        ],
        "Zmax": [
            "4"
        ],
        "Zmin": [
            "-4"
        ]
    }
}
```

E. Vis a vis

```
{
    "Iso3D": {
        "Description": [
            "Vis a vis"
        ],
        "Fxyz": [
            "x^2-x^3+y^2+y^4+z^3-z^4"
        ],
        "Name": [
            "vis a vis"
        ],
        "Xmax": [
            "4"
        ],
        "Xmin": [
            "-4"
        ],
        "Ymax": [
            "4"
        ],
        "Ymin": [
            "-4"
        ],
        "Zmax": [
            "4"
        ],
        "Zmin": [
            "-4"
        ]
    }
}
```

F. Barth Sextik

```
{
    "Iso3D": {
        "Description": [
            "Barth Sextik"
        ],
        "Fxyz": [
            "(((1+sqrt(5))^2)*x^2-
4*y^2)*(((1+sqrt(5))^2)*y^2-
4*z^2)*(((1+sqrt(5))^2)*z^2-4*x^2)-
10*(2+sqrt(5))*(x^2+y^2+z^2-1)^2"
        ],
        "Name": [
            "Barth Sextik"
        ],
        "Xmax": [
            "14"
        ],
        "Xmin": [
            "-14"
        ],
        "Ymax": [
            "14"
        ],
        "Ymin": [
            "-14"
        ],
        "Zmax": [
            "14"
        ],
        "Zmin": [
            "-14"
        ]
    }
}
```

G. Distel

```
{
    "Iso3D": {
        "Description": [
            "Distel"
        ],
        "Fxyz": [

"x^2+y^2+z^2+1500*(x^2+y^2)*(x^2+z^2)*(y^2+z^2)
-1"
        ],
        "Name": [
            "Distel"
        ],
        "Xmax": [
            "1"
        ],
        "Xmin": [
            "-1"
        ],
        "Ymax": [
            "1"
        ],
        "Ymin": [
            "-1"
        ],
        "Zmax": [
            "1"
        ],
        "Zmin": [
            "-1"
        ]
    }
}
```

H. Zitrus

```
{
    "Iso3D": {
        "Description": [
            "Zitrus"
        ],
        "Fxyz": [
            "x^2+z^2-y^3*(1-y)^3"
        ],
        "Name": [
            "Zitrus"
        ],
        "Xmax": [
            "0.5"
        ],
        "Xmin": [
            "-0.5"
        ],
        "Ymax": [
            "1"
        ],
        "Ymin": [
            "-1"
        ],
        "Zmax": [
            "0.5"
        ],
        "Zmin": [
            "-0.5"
        ]
    }
}
```

I. Suss

```
{
    "Iso3D": {
        "Description": [
            "Suss"
        ],
        "Fxyz": [
            "(x^2+2*(9/4)*y^2+z^2-6)^3-x^2*z^3-
2*(9/80) *y^2*z^3"
        ],
        "Name": [
            "Suss"
        ],
        "Xmax": [
            "5"
        ],
        "Xmin": [
            "-5"
        ],
        "Ymax": [
            "5"
        ],
        "Ymin": [
            "-5"
        ],
        "Zmax": [
            "5"
        ],
        "Zmin": [
            "-5"
        ]
    }
}
```

J. Tube

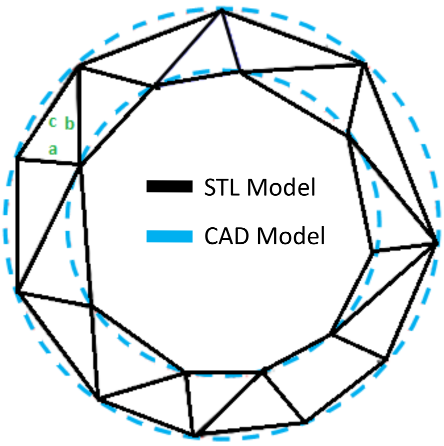
```
{
    "Iso3D": {
        "Component": [
            "Tube"
        ],
        "Description": [
            "Tube is combined by circle and cube "
        ],
        "Fxyz": [
            "if((x^10 + y^10 +z^10 <
3*(3.5^10)),(x^2+y^2-1.5)*(x^2+y^2-3),(x^10 + y^10
+z^10 - 3*(3.5^10)))"
        ],
        "Name": [
            "Tube"
        ],
        "Xmax": [
            " 10"
        ],
        "Xmin": [
            "-10"
        ],
        "Ymax": [
            "10"
        ],
        "Ymin": [
            "-10"
        ],
        "Zmax": [
            "10"
        ],
        "Zmin": [
            "-10"
        ]
    }
}
```

K. THU Medal


```
{
    "Iso3D": {
        "Component": [
            ""
        ],
        "Fxyz": [
            "((sqrt((x-6.5)*(x-6.5)+y*y)-4)^2 + z*z - 0.4)",
            "((sqrt(x*x+y*y)-4)^2 + z*z - 0.4)",
            "((sqrt((x+6.5)*(x+6.5)+y*y)-4)^2 + z*z - 0.4)",
            "(x^2+z^2-0.4)",
            "(z^2+(y+10)*(y+10)-0.4)",
            "((sqrt(x*x+(y+7)*(y+7))-20)^2 + z*z - 0.8)",
            "((x+6.5)*(x+6.5)+z^2-0.4)",
            "(z^2+(y+19)*(y+19)-0.4)",
            "((x+2)*(x+2)+z^2-0.4)",
            "((x-2)*(x-2)+z^2-0.4)",
            "((y+21.5)*(y+21.5)+z^2-0.4)",
            "((x-4)*(x-4)+z^2-0.4)",
            "((x-7.5)*(x-7.5)+z^2-0.4)",
            "((y+23.7)*(y+23.7)+z^2-0.4)"
        ],
        "Name": [
            "THU"
        ],
        "Xmax": [
            "15",
            "15",
            "15",
            "15",
            "4",
            "26",
            "9",
            "-4",
            "30",
            "30",
            "2.3",
            "30",
            "30",
            "7.5"
        ],
        "Xmin": [
            "-15",
            "-15",
            "-15",
            "-15",
            "-4",
            "-26",
            "-30",
            "-9",
            "-26",
            "-26",
            "-2.3",
            "-30",
            "-30",
            "4"
        ],
"Ymax": [
            "15",
            "15",
            "15",
            "-6",
            "15",
            "26",
            "-19",
            "30",
            "-19",
            "-19",
            "30",
            "-19",
            "-19",
            "30"
        ],
        "Ymin": [
            "-15",
            "-15",
            "-15",
            "-17",
            "-15",
            "-30",
            "-24",
            "-30",
            "-24",
            "-24",
            "-30",
            "-24",
            "-24",
            "-30"
        ],
        "Zmax": [
            "15",
            "15",
            "15",
            "15",
            "15",
            "26",
            "15",
            "15",
            "15",
            "15",
            "30",
            "30",
            "30",
            "30"
        ],
        "Zmin": [
            "-15",
            "-15",
            "-15",
            "-15",
            "-15",
            "-26",
            "-30",
            "-30",
            "-30",
            "-30",
            "-30",
            "-30",
            "-30",
            "-30"
        ]
    }
}
```

Appendix II

The ASCII format of a STL file



STL is created by computer 3D Systems. STL is used for rapid prototyping, 3D printing and computer aided manufacturing. The surface geometry of a three-dimensional object would be described by STL without any representation of color and texture. There are two type of STL file: ASCII and binary representations. An STL file describes a raw unstructured tri-angulated surface by the unit normal and vertices (ordered by the right-hand rule) of the triangles using a three-dimensional Cartesian coordinate system. Binary files are more common than ASCII.

a. STL ASCII Format

The ASCII format is primarily intended for testing new CAD interfaces. The large size of its files makes it impractical for general use. The syntax for an ASCII STL file is as follows[36-37]:

solid *name*
  facet normal $n_i\ n_j\ n_k$ +
    outer loop
      vertex $v1_x\ v1_y\ v1_z$
      vertex $v2_x\ v2_y\ v2_z$
      vertex $v3_x\ v3_y\ v3_z$
    endloop
  endfacet
endsolid *name*

which *n* or *v* is a floating point number.

b. STL Binary Format

The binary format uses the IEEE (Institute of Electrical and Electronics Engineers) integer and floating point numerical representation. Below is the syntax for a binary STL file[37]:

Bytes	**Data Type**	**Description**
80	ASCII	Header. No data significance
4	Unsigned long integer	Number of facets in file
4	float	i for normal
4	float	j
4	float	k
4	float	x for vertex 1
4	float	y
4	float	z
4	float	x for vertex 2
4	float	y
4	float	z
4	float	x for vertex 3
4	float	y
4	float	z
2	Unsigned integer	Attribute byte count

Appendix III

The specification of the computer and 3D printer while researching process

a. Computer specification

Windows	: Microsoft Windows 10
Processor	: Intel®Core™2Quad CPU Q9500 @ 2.83GHz
RAM	: 2.00 GB

b. Software specification

(1) MathMod
Version: v3.1 (Win64)

(2) Wolfram Mathematica 10
Version: 10.0.0.0

(3) K3DSurf
Version: v0.6.2

(4) Maplesoft
Version: Maple 18.00

(5) Matlab
Version: R2011a

(6) Blender
Version 2.75a

(7) Netfabb
Version: Basic 6.4.0.252

c. 3D printer specification
Version: UP 3D Printer 2.18

Appendix IV

The printer materials detail

a. ABS (Acrylonitrile Butadiene Styrene)

ABS is a kind of plastics. This material is a common thermoplastic polymer made by polymerizing styrene and acrylonitrile in the presence of poly-butadiene. ABS could be used between -20 and $80°$ C (-4 and $176°$ F) as its mechanical properties vary with temperature. ABS is sold by the different colors like: white, green, red, blue and so forth[38].

b. PLA (Polylactic Acid)

PLA could be produced from the corns. And PLA is biodegradable. PLA is not the recommended plastic material to print the solid object. The material starts to breake down after a few months and printed parts are brittle. The material is very sharp when broken and can be a hazard when removing support material from the printed parts. PLA is harder than ABS material. PLA melts at a lower temperature (around $180° \, C$ to $20° \, C$)[39].