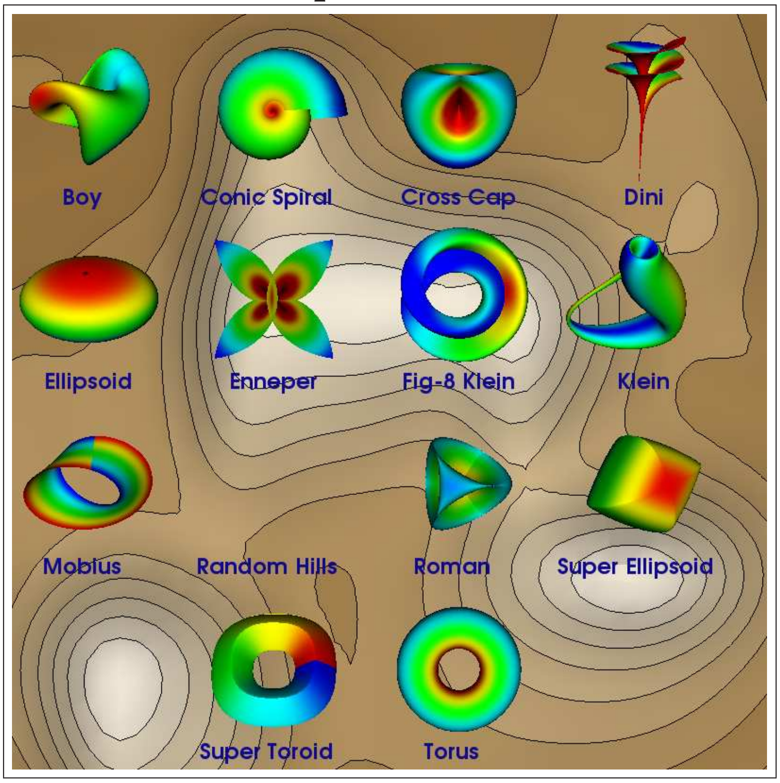
Parametric Equations for Surfaces



Andrew J.P. Maclean *

17 March 2005
Revised 10 September 2006

*Centre for Autonomous Systems, an ARC Centre of Excellence, The University of Sydney

Abstract

This paper describes the parametric surface functions found in VTK. These surfaces are generated by sets of equations describing the deformation of the real plane into the surface.

The introduction provides some background material and definitions of parametric surfaces. Some sample code in TCL and C++ for the Möbuis strip is provided.

In the next chapter, the procedure for creating new classes to generate parametric surfaces is described using the equations for a Figure-8 torus.

The remaining chapters provide a description of each VTK parametric surface. The defining equations and their partial derivatives are also provided.

Pictures of each surface are provided to showcase their features.

Contents

1 Parametric Equations

Introduction

Parametric surfaces are surfaces that are parameterised (usually) by two independent variables. By doing this it is relatively easy to represent surfaces that are self-intersecting, such as Enneper's surface, and surfaces that are non-orientable, such as the Möbius strip. Many of these surfaces are impossible to represent by using implicit functions. Even where implicit functions exist for these surfaces, the tessellated representation is often incorrect. This is because the surfaces are self intersecting and the tessellation generated from the implicit function does not reflect the nearness of points on the surface.

Historically, these surfaces played a powerful and important role in Differential Geometry. They were often used as tools to determine what properties of surfaces are invariant under various transforms. They were also used as tools for investigating the properties of surfaces in higher-dimensions e.g. the Klein bottle. Thus they played an important role in the foundation of, and development of Geometry and General Relativity, underlying modern Mathematics and Physics.

The surfaces have an inherent beauty and were often named after their discoverers. VTK provides an ideal mechanism for users to develop and display their own surfaces.

VTK has a class called vtkFunction() that provides an abstract interface for functions defined by a parametric mapping. This class also has the nice property in that it unifies the spline functions found in VTK, since these are parametric by nature. Thus we can derive from this class and provide the equations that generate the surface in the derived class. To show how this is done, a series of classes have been provided e.g. vtkParametricEllipsoid. Once we have created the class defining the surface, we need to generate it. To do this, we pass the output from the derived class to an instance of vtkParametricFunctionSource(). This will produce the tessellation of the parametric function.

An example - the Möbius strip

A good example is the Möbius strip. Physically we can construct this by taking a long thin strip of paper and joining the ends with a half-twist (180°). If you trace around the surface with a pencil, you will discover that it only has one surface (if we ignore the thickness of the paper). Because there is a half-twist in the surface, the surface is non-orientable.

There is a class in VTK called vtkParametricMobius() that will construct a Möbius strip. In a sense, this is a more faithful representation of a Möbius strip because it only has one side. The non-orientable nature of the surface can be seen by displaying the normals (use vtkHedgeHog() or vtkGlyph3D()). If you so this, you will see a sudden change in direction of the normals where the surface has been joined. You can also investigate curvature by using vtkCurvatures() on this and the other parametric surfaces.

To get you started here is the implementation of the Möbius Strip in TCL and C++, along with a CMake.txt file for C++.

```
# -------------------------------------------------------------
# Call the VTK Tcl packages to make available all VTK commands
# -------------------------------------------------------------
package require vtk
package require vtkinteraction

# -------------------------------------------------------------
# Create a mobius strip
# -------------------------------------------------------------
vtkParametricMobius mobius

vtkParametricFunctionSource mobiusSource
  mobiusSource SetParametricFunction mobius
  mobiusSource SetScalarModeToV

vtkPolyDataMapper mobiusMapper
  mobiusMapper SetInputConnection [mobiusSource GetOutputPort]
  mobiusMapper SetScalarRange -1 1

vtkActor mobiusActor
  mobiusActor SetMapper mobiusMapper
  mobiusActor RotateX 45
  mobiusActor RotateZ -10

# -------------------------------------------------------------
# Create the RenderWindow, Renderer and Interactor
# -------------------------------------------------------------
vtkRenderer ren1
  ren1 SetBackground 0.7 0.8 1
vtkRenderWindow renWin
  renWin AddRenderer ren1
  renWin SetSize 800 800
  renWin SetWindowName "Mobius Strip"
vtkRenderWindowInteractor iren
  iren SetRenderWindow renWin

# add actors
  ren1 AddViewProp mobiusActor

  iren AddObserver UserEvent {wm deiconify .vtkInteract}
  iren AddObserver ExitEvent {exit}

  renWin Render

  mobiusActor SetPosition 0 -0.35 0
  [ren1 GetActiveCamera] Zoom 1.9
```

```
# prevent the tk window from showing up then start the event loop

wm withdraw .
```

Here is the equivalent program in C++:

```
// ---------------------------------------------------------------
// Includes
// ---------------------------------------------------------------
#include "vtkSmartPointer.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"

#include "vtkParametricMobius.h"
#include "vtkParametricFunctionSource.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkCamera.h"

// ---------------------------------------------------------------
// Create a mobius strip
// ---------------------------------------------------------------
int main ( void )
{
  // ---------------------------------------------------------------
  // Select the function and source and then conect them to the
  // mapper and actor.
  // ---------------------------------------------------------------
  vtkSmartPointer<vtkParametricMobius> mobius
      = vtkParametricMobius::New();
  vtkSmartPointer<vtkParametricFunctionSource> mobiusSource
      = vtkSmartPointer<vtkParametricFunctionSource>::New();

  vtkSmartPointer<vtkPolyDataMapper> mobiusMapper
      = vtkSmartPointer<vtkPolyDataMapper>::New();

  vtkSmartPointer<vtkActor> mobiusActor
      = vtkSmartPointer<vtkActor>::New();

    mobiusSource->SetParametricFunction(mobius);
    mobiusSource->SetScalarModeToV();
    mobiusMapper->SetInputConnection(mobiusSource->GetOutputPort());
    mobiusMapper->SetScalarRange(-1, 1);
    mobiusActor->SetMapper(mobiusMapper);
    mobiusActor->RotateX(45);
```

```
    mobiusActor->RotateZ(-10);

  // ------------------------------------------------------------
  // Create the RenderWindow, Renderer and Interactor
  // ------------------------------------------------------------
  vtkSmartPointer<vtkRenderer> ren1
      = vtkSmartPointer<vtkRenderer>::New();
  vtkSmartPointer<vtkRenderWindow> renWin
      = vtkSmartPointer<vtkRenderWindow>::New();
  vtkSmartPointer<vtkRenderWindowInteractor> iren
      = vtkSmartPointer<vtkRenderWindowInteractor>::New();

    renWin->AddRenderer(ren1);
    iren->SetRenderWindow(renWin);

  // ------------------------------------------------------------
  // Add actors and render the scene.
  // ------------------------------------------------------------

  ren1->AddViewProp(mobiusActor);

  ren1->SetBackground(0.7, 0.8, 1);
  renWin->SetSize(800,800);

  renWin->SetWindowName("Mobius Strip");

  iren->Initialize();
  ren1->Render();
  mobiusActor->SetPosition(0, -0.35, 0);
  ren1->GetActiveCamera()->Zoom(1.9);
  iren->Start();

  return 0;
}
```

The corresponding CMake.txt file is:

```
# --------------------------------------------------------------
PROJECT (Mobius)

INCLUDE (${CMAKE_ROOT}/Modules/FindVTK.cmake) IF (USE_VTK_FILE)
  INCLUDE(${USE_VTK_FILE})
ENDIF (USE_VTK_FILE)

SET(EXECUTABLE_OUTPUT_PATH ${PROJECT_BINARY_DIR}/bin
  CACHE PATH
  "Single output directory for all the executables."
  )

ADD_EXECUTABLE(Mobius Mobius.cxx)
```

```
TARGET_LINK_LIBRARIES(Mobius
   vtkCommon vtkGraphics
   vtkHybrid vtkIO vtkRendering)

# ------------------------------------------------------------
```

If you compile and run this code, you should get an image like that on page 32. Use the above code as a template for trying out the other parametric surfaces.

In the next subsection we will discuss some of the fundamentals needed to construct such a surface. The remaining sections describe and detail the equations used for the various surfaces.

Definition

We can think of parameterising a surface as the process of taking a part of a plane and deforming it into a surface. More formally, this is a mapping from a plane to the space containing the surface. If we let *u*,*v* be the coordinates of some part of a plane and (in three-dimensional space) *x*,*y*,*z* be the usual Cartesian coordinates. Thus we have a mapping:

$$(u, v) \rightarrow (f(u, v), g(u, v), h(u, v))$$

where:

$$\begin{aligned} f(u,v) &= x \\ g(u,v) &= y \\ h(u,v) &= z \end{aligned}$$

are functions of u, v that map (u, v) to the usual Cartesian coordinates (x, y, z).

Or, equivalently, we define a surface S as:

$$S(u, v) = (f(u, v), g(u, v), h(u, v))$$

In order to calculate the normals, we need the partial derivatives of x, y, z with respect to u, v, so accordingly, we define:

$$\begin{aligned} x_u &= \frac{\partial f(u,v)}{\partial u} & x_u &= \frac{\partial f(u,v)}{\partial v} \\ y_u &= \frac{\partial g(u,v)}{\partial u} & y_u &= \frac{\partial g(u,v)}{\partial v} \\ z_u &= \frac{\partial h(u,v)}{\partial u} & z_u &= \frac{\partial h(u,v)}{\partial v} \end{aligned}$$

If we let $\mathbf{X} = (x, y, z)$ be a vector representing the point (x, y, z) then we can define $\mathbf{X}_u = (x_u, y_u, z_u)$ and $\mathbf{X}_v = (x_v, y_v, z_v)$ to be the vectors of the partial derivatives at that point. Hence the normal $\mathbf{n}$ is defined by:

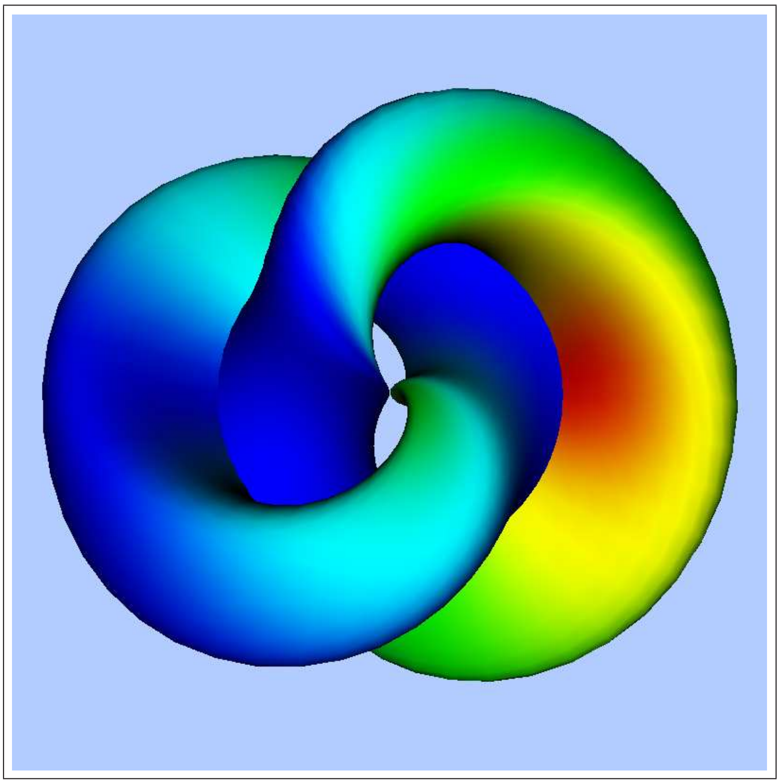
$$\mathbf{n} = \mathbf{X}_u \times \mathbf{X}_v$$

A good development of the brief summary above may be found in [2]. A good online source of information about these surfaces and the geometry associated with them can be found in [10] and [4].

Terminology

In the following discussion for each surface, we will define the equations for each surface and the equations of the partial derivatives of the surface in the above terms. However, note that sometimes we will define auxiliary functions such as $X(u, v)$ to simplify notation. Do not confuse $X(u, v)$ with $\mathbf{X}$.

2 How to create your own surface



Here are some instructions on how to create your own surface. As an example, we will provide equations (and their derivatives) for a Figure-8 torus. Remember that if you do not want to use these equations, search [10], [9] (from which these equations were taken), [4] or [1].

The Figure-8 Torus is constructed by joining the ends of a figure-8 cylinder with a full twist (360°). Contrast this with the figure-8 Klein bottle where the ends of a figure-8 cylinder are joined with a half-twist (180°) on page 27.

Instructions

Step 1: *Determine the parametric equations for your surface and calculate the partial derivatives. For non-orientable surfaces you must calculate the partial derivatives in order to determine the normals on the surface. In this case, VTK is unable to do this automatically. Even for orientable surfaces it is best to calculate the partial derivatives, this seems to give better shading on the surface.*

Step 2: *Create a new class inheriting from vtkParametricFunction().*

Step 3: *For any of the parameters in the equations you are using make them* *protected and use vtkSetMacro() and vtkGetMacro() to set/get values for these parameters.*

Step 4: *Override GetDimension() so that it returns 2.*

Step 5: *Override Evaluate(). In the implementation of this function, drop in the equations that define the surface. Look at one of the existing VTK parametric surface functions to see how this is done.*

Step 6: *Override EvaluateScalar(). This function should return 0 unless you are creating a special scalar for the surface.*

Step 7: *In the constructor, initialise any of the parameters in the equations you are using. In addition, remember to initialise the tessellation parameters: MinimumU, MinimumV, MaximumU, MaximumV, JoinU, JoinV, TwistU, TwistV, ClockwiseOrdering, DerivativesAvailable.*

Equations

Its equations are:

$$\begin{aligned}
f(u,v) &= \cos(u) * (a + \sin(v) * \cos(u) - \sin(2 * v) * \sin(u)/2) \\
g(u,v) &= \sin(u) * (a + \sin(v) * \cos(u) - \sin(2 * v) * \sin(u)/2) \\
h(u,v) &= \sin(u) * \sin(v) + \cos(u) * \sin(2 * v)/2
\end{aligned}$$

Then the figure-8 torus is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ -\pi \leq u \leq \pi,\ -\pi \leq v \leq \pi,\ a > 0$$

At this point, since this is an orientable surface, we have enough information to build a class to calculate the surface, since VTK can automatically calculate the normals. In this exercise, for the sake of completeness, we will calculate the partial derivatives and use them. We can then compare the difference in the appearance of the surface when we calculate the partial derivatives, and hence the normal, to when VTK calculates the normals.

$$
\begin{aligned}
x_u &= -g(u,v) + \cos(u) * (-\sin(v) * \sin(u) - \sin(2 * v) * \cos(u)/2) \\
x_v &= \cos(u) * (\cos(v) * \cos(u) - \cos(2 * v) * \sin(u)) \\
y_u &= f(u,v) + \sin(u) * (-\sin(v) * \sin(u) - \sin(2 * v) * \cos(u)/2) \\
y_v &= \sin(u) * (\cos(v) * \cos(u) - \cos(2 * v) * \sin(u)) \\
z_u &= \sin(v) * \cos(u) - \sin(2 * v) * \sin(u)/2 \\
z_v &= \cos(v) * \sin(u) + \cos(2 * v) * \cos(u)
\end{aligned}
$$

In the above case, vtkParametricTorus.h and vtkParametricTorus.cxx will provide a good model for your class. The tessellation parameters should be set as follows:

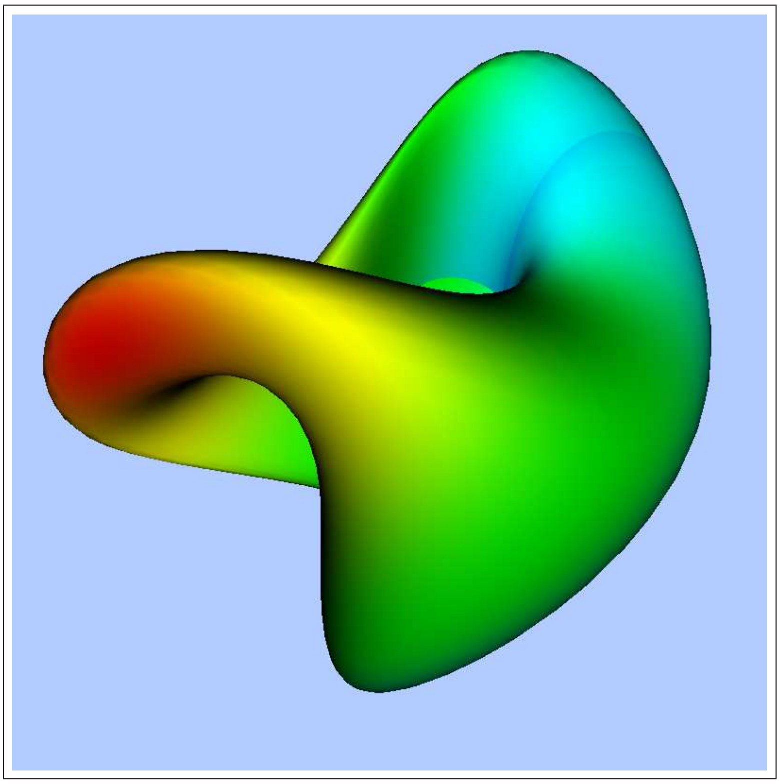
$$
\begin{aligned}
MinimumU &= -vtkMath :: Pi() \\
MinimumV &= -vtkMath :: Pi() \\
MaximumU &= vtkMath :: Pi() \\
MaximumV &= vtkMath :: Pi() \\
JoinU &= 1 \\
JoinV &= 1 \\
TwistU &= 0 \\
TwistV &= 0 \\
ClockwiseOrdering &= 1 \\
DerivativesAvailable &= 1
\end{aligned}
$$

When you get it all working, you should get a surface that looks similar to that at the beginning of this chapter. Hint: Use ($a = 2$).

As an exercise, save the image you have created. Then recalculate the image but let VTK calculate the normals in place of our calculated ones (hint: use DerivativesAvailableOff()), compile, run and compare the images. You should see a difference.

Remember, you can use EvalauateScalar() to create your own personalised scalar for the surface. However, note that UResolution and VResolution are not directly available because these attributes are in vtkParametricFunctionSource.

3 Boy's Surface



This is a Model of the projective plane without singularities. It was found by Werner Boy on assignment from David Hilbert in 1901. The Boy Surface is a self-intersecting continuous surface possessing three-fold symmetry. It is homeomorphic to the real projective plane $\mathbb{R}P^2$. More information about the surface and alternative representations can be found in [5].

In [8] there is an interesting description of the history of Boy's surface, reproduced here:

> Boy's surface was named after Werner Boy who discovered it in 1901. Originally, David Hilbert, assigned Boy the task of proving or disproving if it is possible to immerse a closed form of the real projective plane $\mathbb{R}P^2$ in 3-space $\mathbb{R}^3$ without the formation of a singularity. Other realizations of $\mathbb{R}P^2$ are notorious for the presence of singularities when represented in $\mathbb{R}^3$, and though the subject of projective geometry had been studied for 300 years prior, nobody had yet discovered a closed surface that was singularity free. It had not been mathematically proven such a surface could exist; however, it was generally believed that there was no singularity free closed form of real projective plane. Hilbert was rather surprised when his student proved that such a surface did exist if a continuous double point curve intersects with itself at a triple point. The triple point is located in the centre of the image above. Boy's surface also has the interesting property of having a three fold symmetry axis.
>
> During Boy's time he didn't have a computer to plot out the surface for him, nor did he have the parameterisation needed to mathematically realize his newly found singularity

free surface. How did he come up with such an odd surface, and furthermore how could he defend his claim that it was singularity free? The answer is that he just visualised it, and through this visualisation he was able to make several sketches of the projective plane and found that there were no singularities. Boy never did figure out what the correct parameterisation for his famous surface was, a parametric model of Boy's surface was not derived until in 1978 when B. Morin solved the problem. Ironically Morin was blind and never did get to see what his solution really looked like. Topologically Boy's surface is very similar to the Roman surface, noting that they both are realizations of $\mathbb{R}P^2$ immersed in $\mathbb{R}^3$, and they both have a triple point. In fact the only real difference is that the Roman surface has six Whitney singularities while Boy's surface has no singularities. The surfaces are so similar that a homotopy exists between them.

If you want to construct a paper model of Boy's Surface, look at [3]

Parametric Equations

Define:

$$
\begin{aligned}
X(u,v) &= \cos u \sin v \\
Y(u,v) &= \sin u \sin v \\
Z(u,v) &= \cos v
\end{aligned}
$$

Let:

$$
\begin{aligned}
f(u,v) &= \frac{1}{2}(2X^2 - Y^2 - Z^2 + 2YZ(Y^2 - Z^2) + ZX(X^2 - Z^2) + XY(Y^2 - X^2)) \\
g(u,v) &= \frac{\sqrt{3}}{2}(Y^2 - Z^2 + (ZX(Z^2 - X^2) + XY(Y^2 - X^2))) \\
h(u,v) &= (X + Y + Z)((X + Y + Z)^3 + 4(Y - X)(Z - Y)(X - Z))
\end{aligned}
$$

Then Boy's surface is defined by:

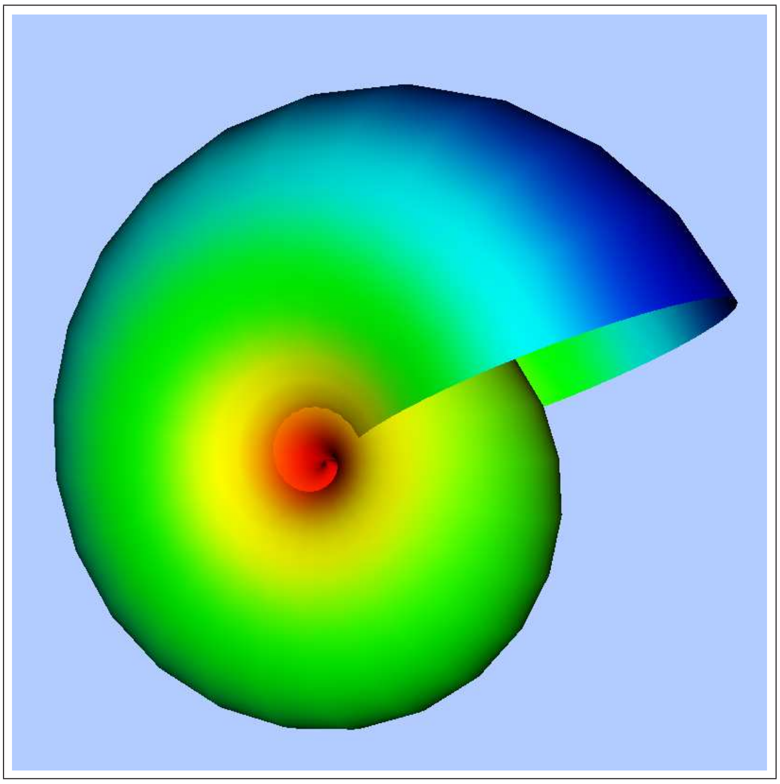
$$S(u,v) = (f(u,v), g(u,v), h(u,v)), \ \textit{where } 0 \le u \le \pi, 0 \le v \le \pi$$

A good representation is found by scaling the x,y,z directions by (1,1,1/8).

The derivatives are given by:

$$
\begin{aligned}
x_u &= -\frac{1}{2}X^4 - Z^3X + 3Y^2X^2 - \frac{3}{2}ZX^2Y + 3ZXY^2 - 3YX - \frac{1}{2}Y^4 + \frac{1}{2}Z^3Y \\
x_v &= (\frac{3}{2}Z^2X^2 + 2ZX - \frac{1}{2}Z^4)\cos u + (-2ZX^3 + 2ZXY^2 + 3Z^2Y^2 \\
&\quad - ZY - Z^4)\sin u + (-\frac{1}{2}X^3 + \frac{3}{2}Z^2X - Y^3 + 3Z^2Y + Z)\sin v \\
y_u &= \sqrt{3}(-\frac{1}{2}X^4 + 3Y^2X^2 + \frac{3}{2}ZX^2Y + YX - \frac{1}{2}Y^4 - \frac{1}{2}Z^3Y) \\
y_v &= \sqrt{3}((-\frac{3}{2}Z^2X^2 + \frac{1}{2}Z^4)\cos u + (-2ZX^3 + 2ZY^2X + ZY)\sin u \\
&\quad + (\frac{1}{2}X^3 - \frac{3}{2}Z^2X + Z)\sin v) \\
z_u &= X^4 + \frac{3}{2}ZX^3 + \frac{3}{2}Z^2X^2 + X^3Y - 3X^2Y^2 + 3ZX^2Y - Y^3X \\
&\quad - \frac{3}{2}ZY^3 - \frac{3}{2}Z^2Y^2 - Z^3Y \\
z_v &= (\frac{1}{2}ZX^3 + \frac{3}{2}Z^3X + Z^4)\cos u \\
&\quad + (4ZX^3 + 3ZX^2Y + \frac{9}{2}Z^2X^2 + \\
&\quad \frac{9}{2}Z^2XY + 3Z^3X + \frac{1}{2}ZY^3 + 3Z^2Y^2 + \frac{3}{2}Z^3Y)\sin u \\
&\quad + (-\frac{3}{2}X^2Y - \frac{3}{2}ZX^2 - \frac{3}{2}XY^2 - 3ZXY - 3Z^2X - Y^3 \\
&\quad - \frac{3}{2}ZY^2 - \frac{1}{2}Z^3)\sin v
\end{aligned}
$$

4 Conic Spiral



A conic spiral may be thought of as a tube of varying radius that has been twisted in three-dimensional space to form a spiral shape. They often resemble sea-shells. Accordingly there are extra parameters to control this appearance. The parametrisation used here is from [7].

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= a\left(1-\frac{v}{2\pi}\right)\cos nv(1+\cos u)+c\cos nv \\ g(u,v) &= a\left(1-\frac{v}{2\pi}\right)\sin nv(1+\cos u)+c\sin nv \\ h(u,v) &= \frac{bv+a\left(1-\frac{v}{2\pi}\right)\sin u}{2\pi} \end{aligned}$$

Where $a, b, c > 0$.

The parameters a, b, c and n need to be carefully selected. A good starting set of parameters are: $a = 0.2$, $b = 1$, $c = 0.1$, $n = 2$, this yields a conic spiral shape. A Nautilus shape can be made with the parameters $a = 0.2$, $b = 0.1$, $c = 0$, $n = 2$.

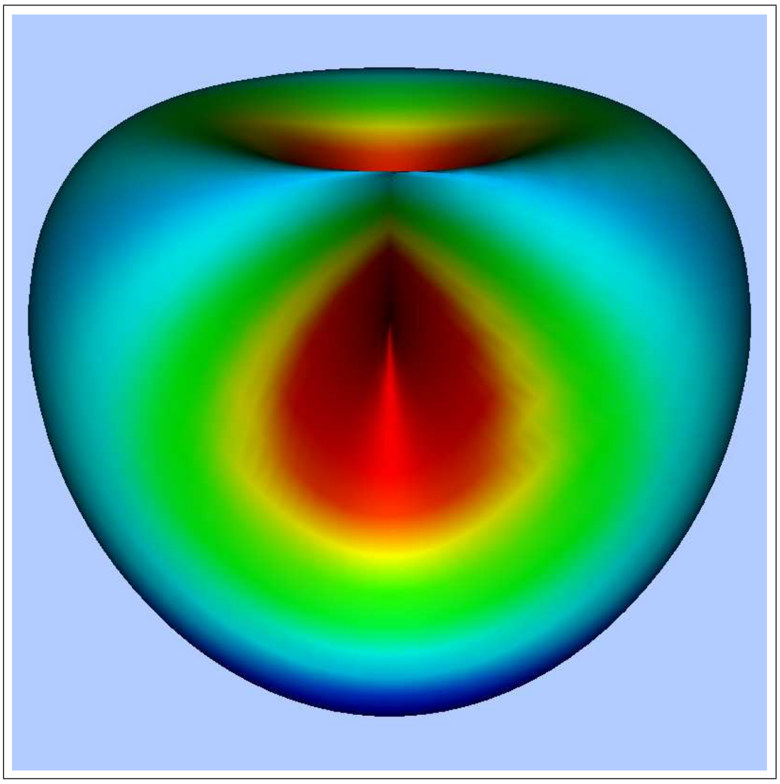
Then the conic spiral is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \ where\ 0 \le u \le 2\pi,\ \ 0 \le v \le 2\pi$$

The derivatives are given by:

$$\begin{aligned}
x_u &= -a * (1 - \frac{v}{2\pi}) \cos nv \sin u \\
x_v &= -\frac{a}{2\pi} \cos nv(1 + \cos u) - an(1 - \frac{v}{2\pi}) \sin nv(1 + \cos u) - cn \sin nv \\
y_u &= -a(1 - \frac{v}{2\pi}) \sin nv \sin u \\
y_v &= -\frac{a}{2\pi} \sin nv(1 + \cos u) + an(1 - \frac{v}{2\pi}) \cos nv(1 + \cos u) + cn \cos nv \\
z_u &= a(1 - \frac{v}{2\pi}) \cos u \\
z_v &= \frac{b - a \sin u}{2\pi}
\end{aligned}$$

5 Cross-Cap



The cross-cap is an image of the real projective plane $\mathbb{R}P^2$. It has a segment of double points, which terminate in two "pinch" points, or Whitney singularities. Every neighbourhood of the pinch point intersects itself.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= \cos u \sin 2v \\ g(u,v) &= \sin u \sin 2v \\ h(u,v) &= \cos v \cos v - \cos^2 u \sin^2 v \end{aligned}$$

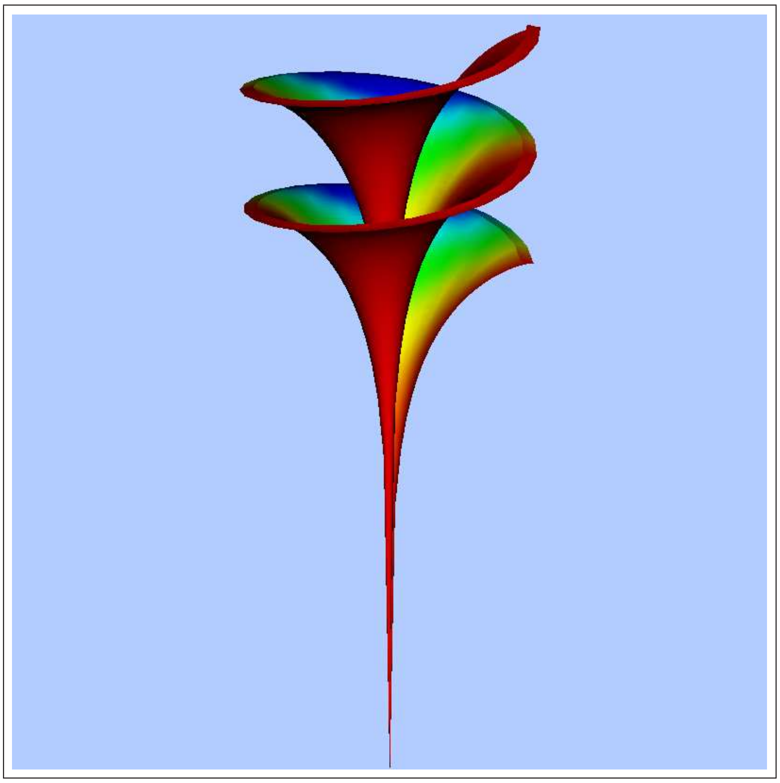
Then the cross-cap is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ 0 \le u \le \pi,\ 0 \le v \le \pi$$

The derivatives are given by:

$$
\begin{aligned}
x_u &= -y \\
x_v &= 2\cos u \cos 2v \\
y_u &= x \\
y_v &= 2\sin u \cos 2v \\
z_u &= 2\cos u \sin u \sin^2 v \\
z_v &= -2\cos v \sin v(1 + \cos^2 u)
\end{aligned}
$$

6 Dini's Surface



This is a surface obtained by taking a pseudosphere and twisting it. Note that a pseudosphere has constant negative Gaussian curvature, hence the name "pseudo-sphere" as an analogy to a sphere which has constant positive Gaussian curvature. Like a pseudosphere, this surface has constant negative curvature. It was named after Ulisse Dini.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= a\cos u \sin v \\ g(u,v) &= a\sin u \sin v \\ h(u,v) &= a(\cos v + \log\tan\frac{v}{2}) + bu \end{aligned}$$

Where $a, b > 0$.

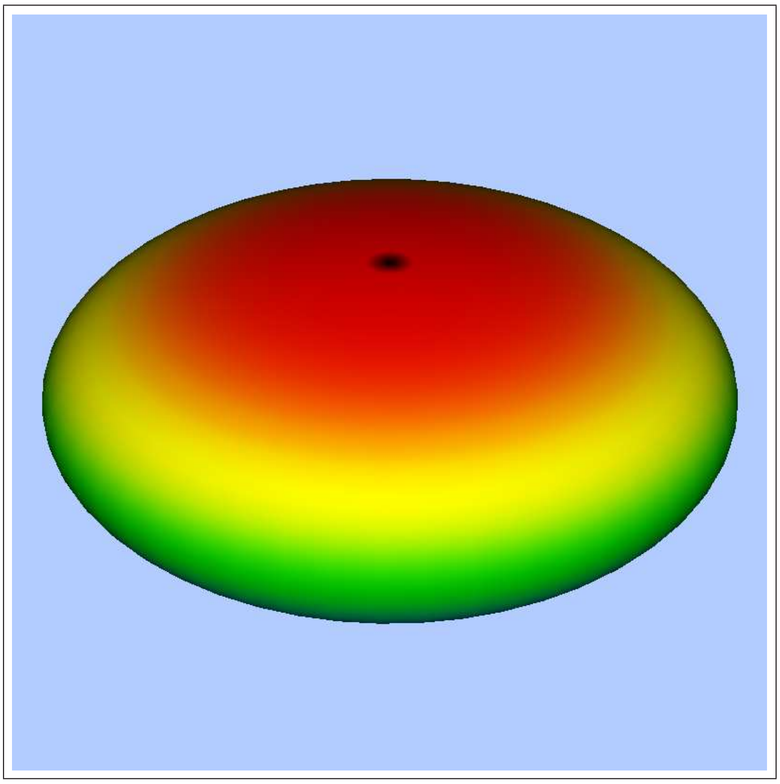
Then Dini's surface is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)), \ \textit{where}\ 0 \le u,\ 0 < v$$

The derivatives are given by:

$$
\begin{aligned}
x_u &= g(u,v) \\
x_v &= a\cos u\cos v \\
y_u &= f(u,v) \\
y_v &= a\sin u\cos v \\
z_u &= b \\
z_v &= a\left[-\sin v + \frac{1 + tan^2\frac{v}{2}}{2tan\frac{v}{2}}\right]
\end{aligned}
$$

7 Ellipsoid



The ellipsoid can be used to represent a sphere or a spheroid.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= a\cos u \sin v \\ g(u,v) &= b\sin u \sin v \\ h(u,v) &= c\cos v \end{aligned}$$

Where $a, b, c > 0$.

Then the ellipsoid is defined by:

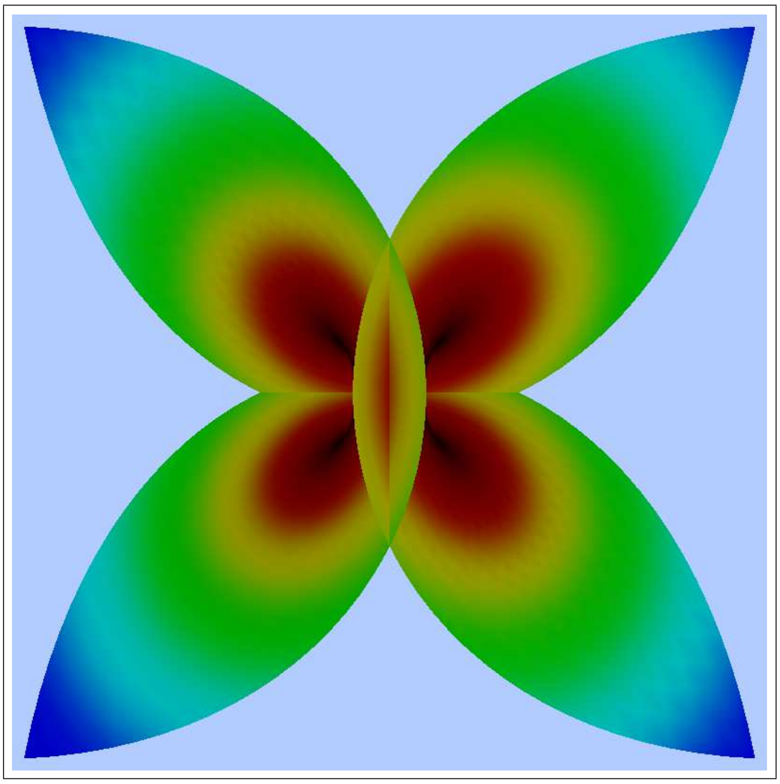
$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ 0 \le u < 2\pi,\ 0 \le v \le 2\pi$$

If the lengths of two axis of the ellipsoid are the same then the figure is called a spheroid. In this case, assuming $a = b$, it is an oblate spheroid if $c < a$ and a prolate spheroid if $c > a$. If the lengths of all three axes are the same $a = b = c$ then the figure is a sphere. u is known as the azimuthal angle, and v is known as the polar angle.

The derivatives are given by:

$$\begin{aligned}
x_u &= -a \sin u \sin v \\
x_v &= a \cos u \cos v \\
y_u &= b \cos u \sin v \\
y_v &= b \sin u \cos v \\
z_u &= 0 \\
z_v &= -c \sin v
\end{aligned}$$

8 Enneper's Surface



Enneper's surface is an example of a self-intersecting minimal surface. It is named after the German mathematician Alfred Enneper who constructed the surface in 1863. Although it has a fairly uncomplicated parametrisation, it is somewhat hard to visualise because of its self-intersections. By keeping the plot range small, the plot suggests the self-intersections exhibited by the surface and the structure of the surface's centre.

Note that the self-intersection curves are subsets of the planes y = 0 and x = 0. The surface is a special case of the more general Enneper's surface of degree n. These surfaces tend to be even more complicated and difficult to visualise.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= u - \frac{u^3}{3} + uv^2 \\ g(u,v) &= v - \frac{v^3}{3} + vu^2 \\ h(u,v) &= u^2 - v^2 \end{aligned}$$

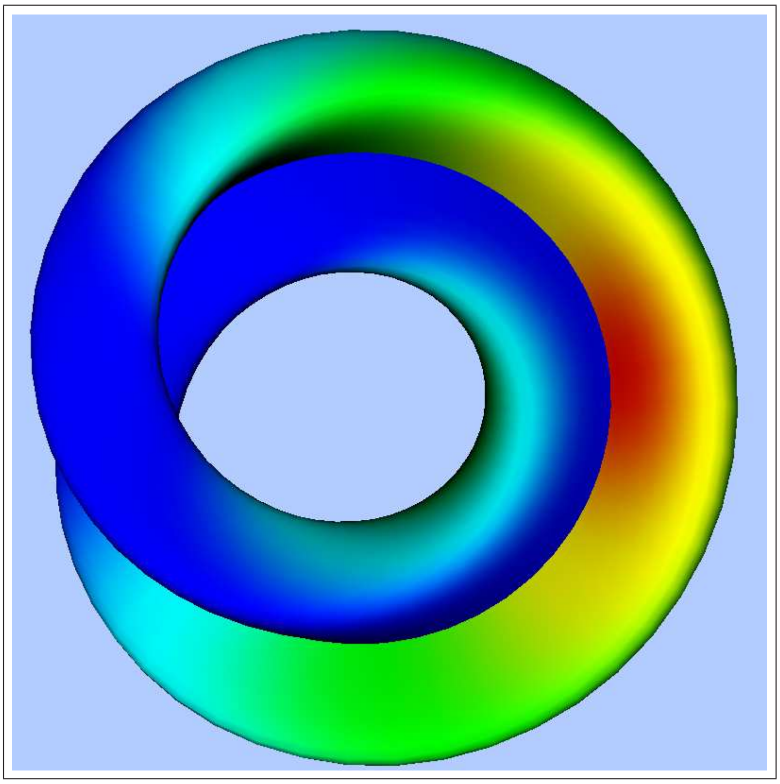
Then Enneper's Surface is defined by:

$$S(u, v) = (f(u, v), g(u, v), h(u, v)),\ \ where\ u, v \in \mathbb{R}$$

The derivatives are given by:

$$\begin{aligned}
x_u &= 1 - u^2 + v^2 \\
x_v &= 2uv \\
y_u &= 2uv \\
y_v &= 1 - v^2 + u^2 \\
z_u &= 2u \\
z_v &= -2v
\end{aligned}$$

9 Figure 8 Klein Bottle



A Klein bottle is a closed surface with no interior and only one surface. It is cannot be realised in 3 dimensions without intersecting surfaces. It can be realised in 4 dimensions by considering the map $G : \mathbb{R}^2 \to \mathbb{R}^4$ given by:

$$G(u, v) = ((r \cos v + a) \cos u, (r \cos v + a) \sin u, r \sin v \cos \frac{u}{2}, r \sin v \sin \frac{u}{2})$$

This representation of the immersion in $\mathbb{R}^3$ is formed by taking two Möbius strips and joining them along their boundaries, this is the so-called “Figure-8 Klein Bottle”. Alternatively, imagine joining the ends of a figure-8 cylinder with a half-twist (180°). This particular representation is not what Felix Klein first envisaged in 1882. The Klein bottle has the properties of being nonorientable and having no volume.

The main utility of this surface lies in the fact that the equations are simpler than those used to represent the classical Klein bottle.

Parametric Equations

Define:

$$
\begin{aligned}
f(u,v) &= \cos u(a + \sin v \cos\frac{u}{2} - \frac{\sin 2v \sin\frac{u}{2}}{2}) \\
g(u,v) &= \sin u(a + \sin v \cos\frac{u}{2} - \frac{\sin 2v \sin\frac{u}{2}}{2}) \\
h(u,v) &= \sin\frac{u}{2}\sin v + \frac{\cos\frac{u}{2}\sin 2v}{2}
\end{aligned}
$$

Where $a > 0$

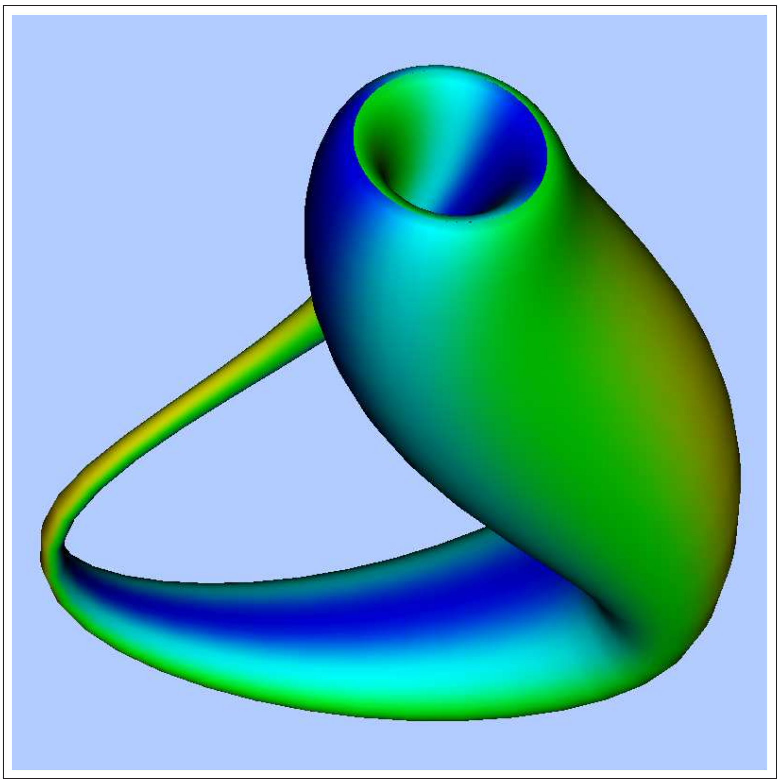
Then the Figure8 Klein Bottle is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ where\ \pi \leq u < \pi,\ \pi \leq v \leq \pi$$

The derivatives are given by:

$$
\begin{aligned}
x_u &= -g(u,v) - \cos u\frac{2\sin v \sin\frac{u}{2} + \sin 2v \cos\frac{u}{2}}{4} \\
x_v &= \cos u(\cos v \cos\frac{u}{2} - \cos 2v \sin\frac{u}{2}) \\
y_u &= f(u,v) - \sin u\frac{2\sin v \sin\frac{u}{2} + \sin 2v \cos\frac{u}{2}}{4} \\
y_v &= \sin u(\cos v \cos\frac{u}{2} - \cos 2v \sin\frac{u}{2}) \\
z_u &= \frac{\cos\frac{u}{2}\sin v}{2} - \frac{\sin\frac{u}{2} sin 2v}{4} \\
z_v &= \frac{\sin\frac{u}{2}\cos v}{2} + \cos\frac{u}{2}\cos 2v
\end{aligned}
$$

10 Klein Bottle



A Klein bottle is a closed surface with no interior and only one surface. It is cannot be realised in 3 dimensions without intersecting surfaces. It can be realised in 4 dimensions by considering the map $G : \mathbb{R}^2 \rightarrow \mathbb{R}^4$ given by:

$$G(u, v) = ((r\cos v + a)\cos u, (r\cos v + a)\sin u, r\sin v\cos\frac{u}{2}, r\sin v\sin\frac{u}{2})$$

This representation of the immersion in $\mathbb{R}^3$ is formed by taking two Möbius strips and joining them along their boundaries. This is the form imagined by the topologist Kelix Klein in 1882, when he imagined sewing two Möbius strips together to create a single sided bottle with no boundary. The Klein bottle has the properties of being nonorientable and having no volume.

Thanks to Robert Israel[6] for providing the equations defining this classical representation of the immersion in $\mathbb{R}^3$.

Parametric Equations

Define:

$$\begin{aligned}
f(u,v) &= -\frac{2}{15}\cos u\,(3\cos v + 5\sin u\cos v\cos u - 30\sin u \\
&\quad - 60\sin u\cos^6 u + 90\sin u\cos^4 u) \\
g(u,v) &= \frac{-1}{15}\sin u\,(80\cos v\cos^7 u\sin u + 48\cos v\cos^6 u \\
&\quad - 80\cos v\cos^5 u\sin u - 48\cos v\cos^4 u \\
&\quad - 5\cos v\cos^3 u\sin u - 3\cos v\cos^2 u \\
&\quad + 5\sin u\cos v\cos u + 3\cos v - 60\sin u) \\
h(u,v) &= \frac{2}{15}\sin v\,(3 + 5\sin u\cos u)
\end{aligned}$$

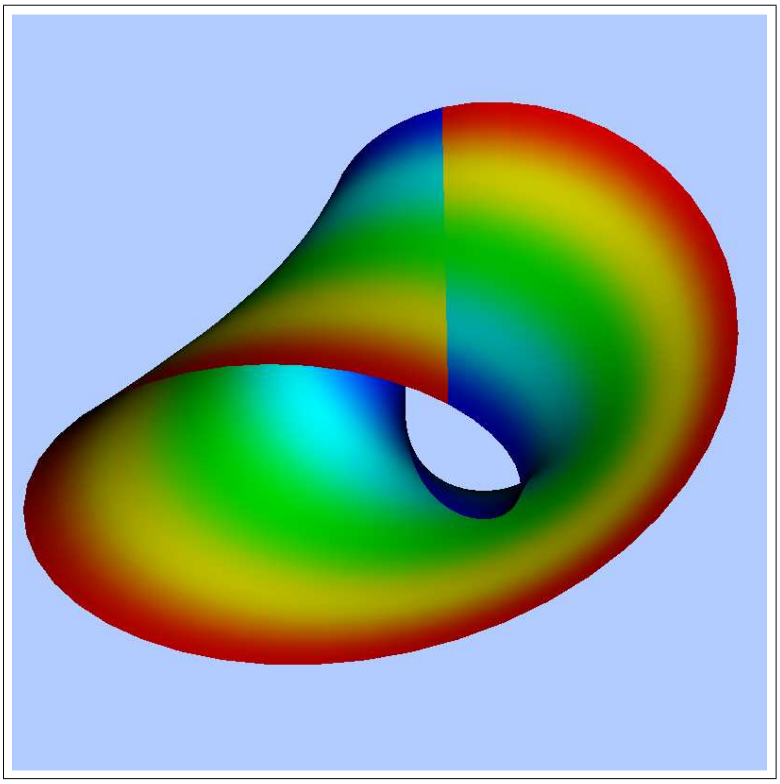
Then the Klein Bottle is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ \pi \le u < \pi,\ \pi \le v \le 2\pi$$

The derivatives are given by:

$$
\begin{aligned}
x_u &= -\tan u\, f(u,v) \\
&\quad - \frac{2}{15}\cos u\,(5\cos v\cos^2 u - 5\sin^2 u\cos v - 30\cos u - 60\cos^7 u \\
&\quad + 360\sin^2 u\cos^5 u + 90\cos^5 u - 360\sin^2 u\cos^3 u) \\
x_v &= -\frac{2}{15}\cos u\,(-3\sin v - 5\sin u\sin v\cos u) \\
y_u &= -\frac{1}{15}\cos u\,(80\cos v\cos^7 u\sin u + 48\cos v\cos^6 u - 80\cos v\cos^5 u\sin u \\
&\quad - 48\cos v\cos^4 u - 5\cos v\cos^3 u\sin u - 3\cos v\cos^2 u + 5\sin u\cos v\cos u \\
&\quad + 3\cos v - 60\sin u) - \frac{1}{15}\sin u(-560\cos v\cos^6 u\sin^2 u \\
&\quad + 80\cos v\cos^8 u - 288\cos v\cos^5 u\sin u + 400\cos v\cos^4 u\sin^2 u \\
&\quad - 80\cos v\cos^6 u + 192\cos v\cos^3 u\sin u + 15\sin^2 u\cos v\cos^2 u \\
&\quad - 5\cos v\cos^4 u + 6\sin u\cos v\cos u + 5\cos v\cos^2 u \\
&\quad - 5\sin^2 u\cos v - 60\cos u) \\
y_v &= \frac{-1}{15}\sin u\,(-80\sin v\cos^7 u\sin u - 48\sin v\cos^6 u \\
&\quad + 80\sin v\cos^5 u\sin u + 48\sin v\cos^4 u + 5\sin v\cos^3 u\sin u \\
&\quad + 3\sin v\cos^2 u - 5\sin u\sin v\cos u - 3\sin v) \\
z_u &= \frac{2}{15}\sin v\,(5\cos^2 u - 5\sin^2 u) \\
z_v &= \frac{2}{15}\cos v\,(3 + 5\sin u\cos u)
\end{aligned}
$$

11 Möbius strip



The Möbius strip is a single-sided non-orientable surface. This surface was discovered by August Ferdinand Möbius in 1858. This is the easiest non-orientable surface to visualise because you can construct it from a strip of paper. This is an important surface in topology, often one or more of these surfaces are added to other surfaces to build new ones. For example, Klein used two of them to build his Klein bottle, and, by gluing a Möbuis strip to the edge of a disk, we can construct three possible surfaces, namely Boy's surface, the Cross Cap and the Roman Surface . It is also depicted in artists representations, for example, Möbius strip II, 1962, by M.C. Escher, Escher Foundation, Gemeentemuseum Den Haag, The Hague.

The picture, above, clearly shows the half-twist in the surface.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= (a - v\sin\frac{u}{2})\sin u \\ g(u,v) &= (a - v\sin\frac{u}{2})\cos u \\ h(u,v) &= v\cos\frac{u}{2} \end{aligned}$$

Where $a > 0$.

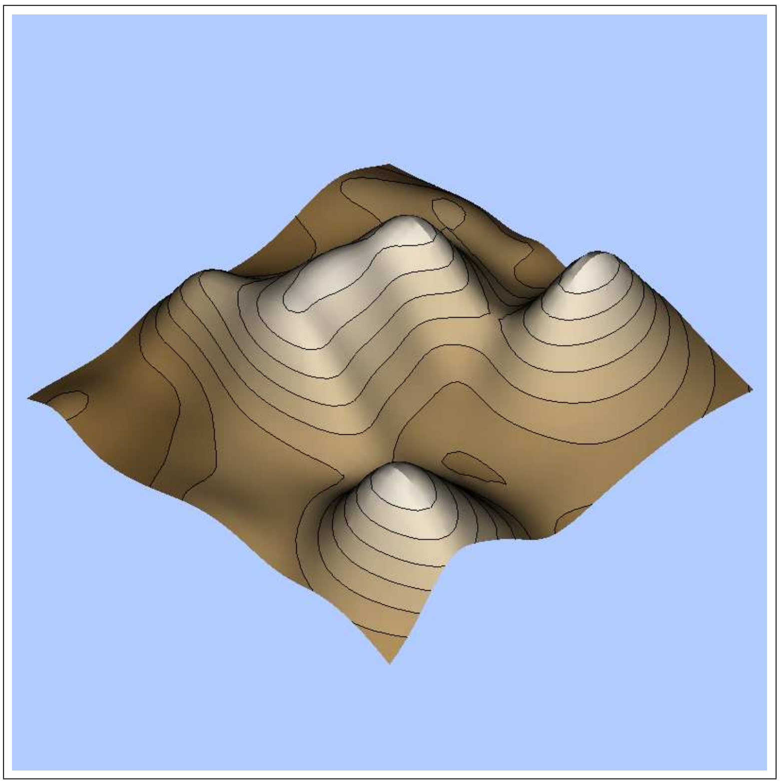
Then the Möbius strip is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ 0 \le u \le 2\pi,\ -1 < v < -1, a > 0$$

The derivatives are given by:

$$\begin{aligned} x_u &= -\frac{v\cos\frac{u}{2}\sin u}{2} + g(u,v) \\ x_v &= -\sin\frac{u}{2}\sin u \\ y_u &= -\frac{v\cos\frac{u}{2}\cos u}{2} - f(u,v) \\ y_v &= -\sin\frac{u}{2}\cos u \\ z_u &= -\frac{v\sin\frac{u}{2}}{2} \\ z_v &= \cos\frac{u}{2} \end{aligned}$$

12 Random Hills



This function produces random hills on a surface. We take a series of points that have been assigned an amplitude or height along with variances in the $x-$, $y-$ directions and sum the Gaussian contributions from all the hills to get the height of the point.

Parametric Equations

Define:

The number of hills as N. The position of the i'th hill as $(\overline{x_i}, \overline{y_i})$ the variance of the hill as $(\mathbf{v}_{x_i}, \mathbf{v}_{y_i})$ and its amplitude as a. The contribution from the i'th hill at position (u, v) on the plane is:

$$
\begin{aligned}
x_i &= \frac{(u - \overline{x_i})}{\mathbf{v}_{x_i}} \\
y_i &= \frac{(v - \overline{y_i})}{\mathbf{v}_{y_i}}
\end{aligned}
$$

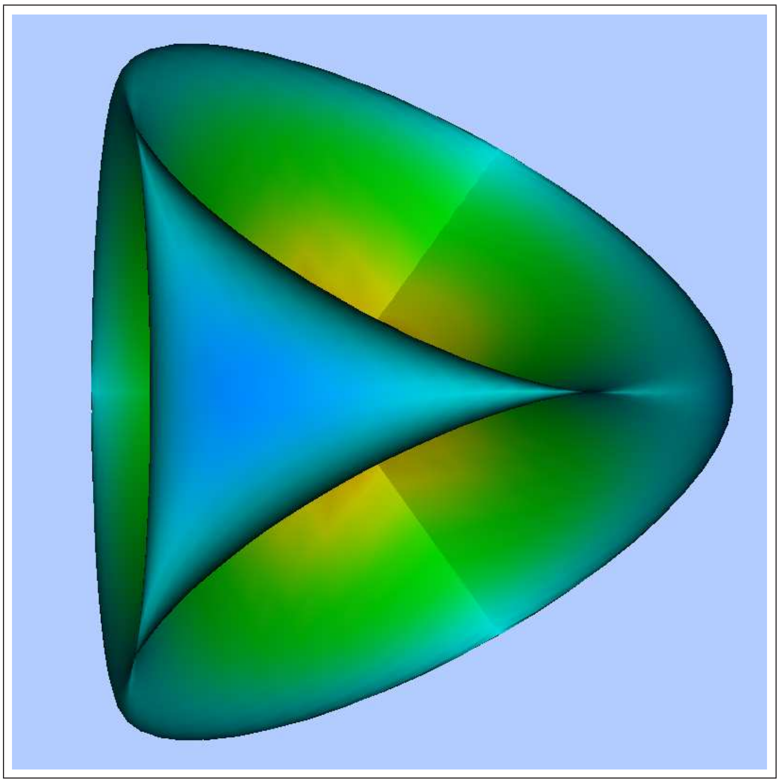
Thus:

$$
\begin{aligned}
f(u,v) &= u \\
g(u,v) &= v \\
h(u,v) &= \sum_{i=1}^{N} a e^{-\frac{x_i^2 + y_i^2}{2}}
\end{aligned}
$$

Then the Random Hills surface is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v))$$

13 Steiner's Roman Surface



The Roman surface, also known as Steiner's Roman surface, was discovered in 1844 by Jacob Steiner, it is called the Roman Surface because he was in Rome at the time. While Steiner was working on the solution he was having some difficulty with solving a complicated polynomial, so he asked his friend Karl Weierstrass to see if he could find a solution, and the solution was:

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

The Roman surface is a realization of the real projective plane $\mathbb{R}P^2$ in 3-space $\mathbb{R}^3$. As such, because it is non-orientable and closed, it will have self-intersections. The Roman Surface has six Whitney singularities, these are points that have no tangent plane which means that these points are not immersed in $\mathbb{R}^3$. Essentially, the Roman surface consists of six cross caps stuck together.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= \frac{a^2\cos^2 v\sin 2u}{2} \\ g(u,v) &= \frac{a^2\sin u\sin 2v}{2} \\ h(u,v) &= \frac{a^2\cos u\sin 2v}{2} \end{aligned}$$

Where $a > 0$.

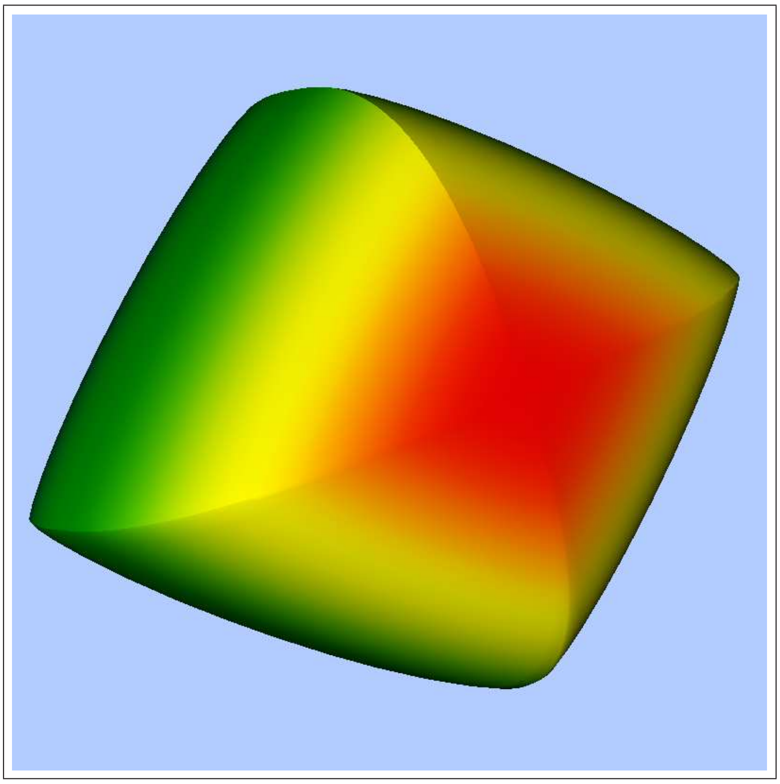
Then the Roman Surface is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ 0 \le u \le \pi,\ \ 0 \le v \le \pi$$

The derivatives are given by:

$$\begin{aligned} x_u &= a^2\cos^2 v\cos 2u \\ x_v &= -a^2\cos v\sin 2u\sin v \\ y_u &= \frac{a^2\cos u\sin 2v}{2} \\ y_v &= a^2\sin u\cos 2v \\ z_u &= -\frac{a^2\sin u\sin 2v}{2} \\ z_v &= a^2\cos u\cos 2v \end{aligned}$$

14 Super Ellipsoid



A superellipsoid is an ellipsoid where power terms have been added to the trigonometric functions that define the surface. This allows us to generate a wide variety of shapes because power terms < 1 generate rounded squarish convex surfaces whilst power terms > 1 generate surfaces that are concave with sharp edges.

Parametric Equations

Define:

$$
\begin{aligned}
f(u,v) &= a \sin^{n_1} v \cos^{n_2} u \\
g(u,v) &= b \sin^{n_1} v \sin^{n_2} u \\
h(u,v) &= c \sin^{n_1} v
\end{aligned}
$$

The shape of the ellipsoid in the $z-$ direction is controlled by n_1 and the shape in the $x-y$ plane is controlled by n_2 where $n_1, n_2 > 0$. There may be numerical issues with very small or large values of n_1 or n_2. The scale factors for each axis ($x-, y-, z-$) are a, b and c where $a, b, c > 0$.

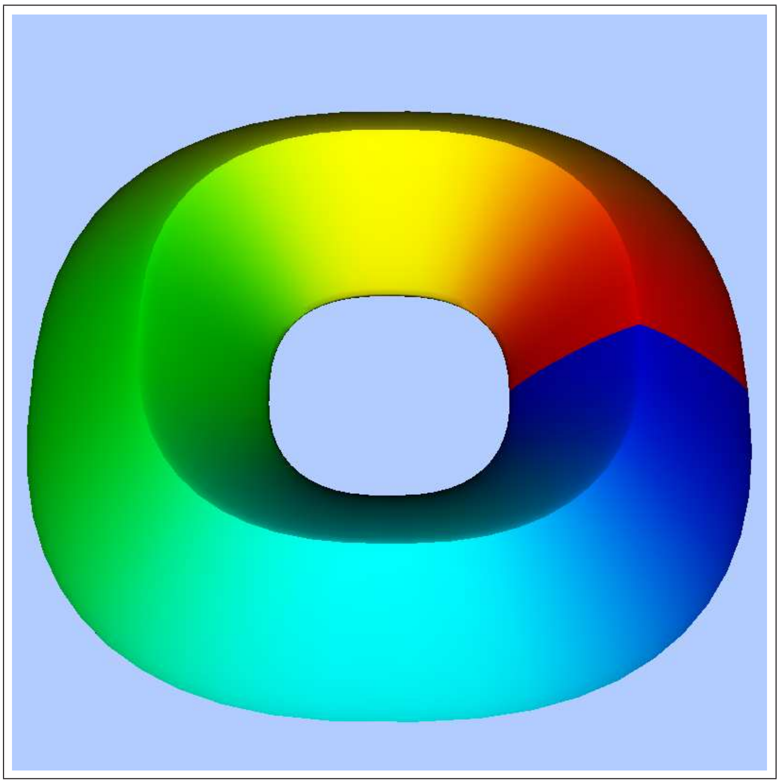
Then the superellipsoid is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ where\ 0 \le u \le 2\pi,\ 0 \le v \le \pi$$

The derivatives are given by:

$$\begin{aligned}
x_u &= -a\, n_2 \tan u \sin^{n_1} v \cos^{n_2} u \\
x_v &= a\, n_1 \cot v \sin^{n_1} v \cos^{n_2} u \\
y_u &= b\, n_2 \cot u \sin^{n_1} v \ \sin^{n_2} u \\
y_v &= b\, n_1 \cot v \sin^{n_1} v \ \sin^{n_2} u \\
z_u &= 0 \\
z_v &= c\, n_1 \cot v \sin^{n_1} v
\end{aligned}$$

15 Super Toroid



A supertoroid is a torus where power terms have been added to the trigonometric functions that define the surface. This allows us to generate a wide variety of shapes because power terms < 1 generate rounded squarish convex surfaces whilst power terms > 1 generate surfaces that are concave with sharp edges.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= r_x(c + a\cos^{n_2} v)\cos^{n_1} u \\ g(u,v) &= r_y(c + a\cos^{n_2} v)\sin^{n_1} u \\ h(u,v) &= r_z a \sin^{n_2} v \end{aligned}$$

The radius from the centre to the middle of the ring of the torus is c, and a is the radius of the cross-section of the ring of the torus, where $c, a > 0$. Generally $c > a$, giving the usual torus shape. The shape of the torus ring is controlled by n_1 and the shape of the cross-section of the ring is controlled by n_2 where $n_1, n_2 > 0$. There may be numerical issues with very small or large values of n_1 or n_2. The scale factors for each axis $(x-, y-, z-)$ are r_x, r_y and r_z where $r_x, r_y, r_z > 0$.

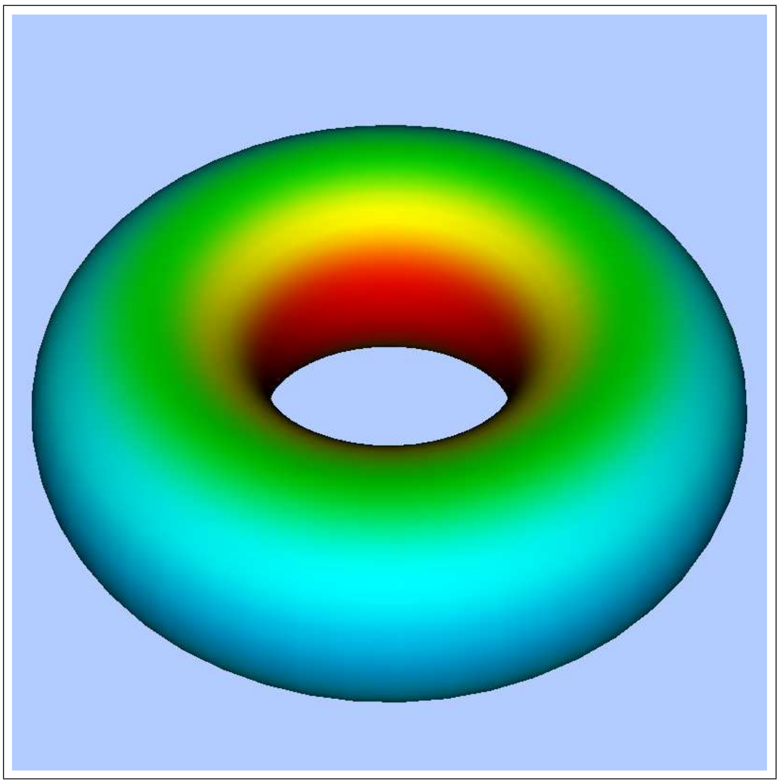
Then the supertoroid is defined by:

$$S(u, v) = (f(u, v), g(u, v), h(u, v)),\ \textit{where}\ 0 \le u \le 2\pi,\ 0 \le v \le 2\pi$$

The derivatives are given by:

$$\begin{aligned}
x_u &= -r_x\, n_1\, (c + a \cos^{n_2} v)\ \tan u\ \cos^{n_1} u \\
x_v &= -r_x\, n_2\, a\ \cos^{n_1} u\ \cos^{n_2} v\ \tan v \\
y_u &= r_y\, n_1\, (c + a \cos^{n_2} v) \cot u\ \sin^{n_1} u \\
y_v &= -r_y\, n_2\, a \sin^{n_1} u\ \cos^{n_2} v\ \tan v \\
z_u &= 0 \\
z_v &= r_z\, n_2\, a\ \sin^{n_2} v\ \cot v
\end{aligned}$$

16 Torus



This is a surface having a genus of one and thus posesses a single hole. This is generated by revolving a circle about an axis coplanar with the circle but not touching the circle.

Parametric Equations

Define:

$$\begin{aligned} f(u,v) &= (c + a\cos v)\cos u \\ g(u,v) &= (c + a\cos v)\sin u \\ h(u,v) &= a\sin v \end{aligned}$$

The radius from the centre to the middle of the ring of the torus is c and a is the radius of the cross-section of the ring of the torus where $c, a > 0$. Generally $c > a$, giving the usual torus shape.

Then the torus is defined by:

$$S(u,v) = (f(u,v), g(u,v), h(u,v)),\ \textit{where}\ 0 \le u \le 2\pi,\ 0 \le v \le 2\pi$$

The derivatives are given by:

$$
\begin{aligned}
x_u &= -g(u, v) \\
x_v &= -a \sin v \cos u \\
y_u &= f(u, v) \\
y_v &= -a \sin v \sin u \\
z_u &= 0 \\
z_v &= a \cos v
\end{aligned}
$$

References

[1] Paul Bourke, *Surfaces*, Has a wide, varied and interesting collection of surfaces at http://astronomy.swin.edu.au/~pbourke/surfaces/.

[2] Manfredo P. do Carmo, *Differential geometry of curves and surfaces*, Prentice-Hall, 1976, An excellent introduction to differential geometry.

[3] Joe Fields, *Constructing boy's surface*, Step by step instructions for constructing a paper model of Boy's Surface are found at http://www.southernct.edu/~fields/topology/Boys_surface.html.

[4] Weiqing Gu, *Curves and surfaces, a digital library of mathematically interesting and important curves and surfaces.*, Department of Mathematics, Harvey Mudd College,, This digital library at http://www.math.hmc.edu/~gu/curves_and_surfaces/index.html attempts to collect together some mathematically interesting curves and surfaces, along with formulas describing some of their properties, and text highlighting special features.

[5] Amy Hawthorne and Jenny Posson-Brown, *Boy's surface.*, This is found at http://maven.smith.edu/~patela/boysurface/AmyJenny/boy_surface.html.

[6] Robert Israel, *Robert b. israel*, This is found at http:www.math.ubc.ca/~israel.

[7] T. Nordstrand, *Conic spiral or seashell.*, This is found at http://www.uib.no/people/nfytn/shelltxt.htm.

[8] Mark Sudduth, *Cool physics*, This is found at http://www.coolphysics.com/4d/.

[9] The University of Minnesota Science and Technology Center, The Geometry Center (now closed), *The topological zoo*, An on-line compendium of topological and geometrical properties of curves and surfaces at http://www.geom.umn.edu/zoo/.

[10] Eric Weisstein, *Eric weisstein's, world of mathematics*, Wolfram Research, An extensive on-line reference collection of mathematical definitions, formulae, curves, surfaces and their properties at http://mathworld.wolfram.com.