CALCULATING GEODESICS ON SURFACES

CALCULATING GEODESICS ON SURFACES

By

ANDRIJANA BURAZIN, H.B.Sc

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Science

McMaster University

MASTER OF SCIENCE (2008) **McMaster University**
(Mathematics) **Hamilton, Ontario**

TITLE: Calculating Geodesics on Surfaces

AUTHOR: Andrijana Burazin, H.B.Sc (University of Toronto)

SUPERVISOR: Dr. Miroslav Lovrić

NUMBER OF PAGES: ix, 72

McMaster University

Department of Mathematics and Statistics

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled **"Calculating Geodesics on Surfaces"** by **Andrijana Burazin** in partial fulfillment of the requirements of the degree of **Master of Science**.

DATED: April 2008

SUPERVISOR: Dr. Miroslav Lovrić

READERS: Dr. Konstantin Khanin

Dr. Maung Min-Oo

Abstract

In this thesis, we mainly study geodesics on various two dimensional surfaces. All the background material needed throughout the thesis is provided, including an explanation of the theory of geodesics. We will calculate geodesics using two numerical methods: Euler's method and Runge-Kutta method of fourth order. Using Maple, we will test the accuracy of the numerical methods on a test case surface, the Poincaré half plane. Later, we proceed to investigate several interesting surfaces by numerically calculating geodesics. From the investigated surfaces, we will draw similiarities between the human cerebral cortex and certain surfaces. The human cerebral cortex is the most intensely studied part of the brain and it is believe that their exists a relation between the function and structure of the cortex. Geodesic analysis can possibly be an essential tool in better understanding the cortical surface as it is in many disciplines of science to understand the nature of physical based problems.

Acknowledgements

I would like to take this opportunity to thank my thesis supervisor, Dr. Miroslav Lovrić for taking me on as his student. He always took time out of his busy schedule to provide me with valuable guidance and advice. It was a pleasure and honour working with him on my Master of Science thesis. Furthermore, I would like to thank Dr. Konstantin Khanin and Dr. Maung Min-Oo for serving on my Master of Science thesis defense committee. Their presence was greatly appreciated. Finally, I would like to thank my mom, my dad and my "loving" brother for their support and encouragement.

Contents

List of Figures

Chapter 1

Introduction

The motivation behind this thesis stems from my interest in studying and understanding the relation between certain measurements performed on the surface of the human brain and geodesics. The surface of the human brain is referred to as the human cerebral cortex. It is the most intensely studied part of the brain. We will define and work with the cortical surface much later.

Geographically, the human cerebral cortex has an irregular shape with many dents and grooves that somewhat resembles the ellipsoid surface. It is believed that the there exists a relationship between the structure and function of the cortex. Several numerical methods have been developed to visually reconstruct the cortical surface; however, due to the complexity of the cortex, each method has drawbacks in minimizing the distortions in topological representations. Thus, to some degree, the drawbacks have placed an impediment in finding a parallel between the function and the structure of the cortex. [11]

According to the paper "Functional and structural mapping of human cerebral cortex: solutions are in the surface" by Van Essen et al., the most efficient way to topologically represent the cortical surface is by using the surface-based warping algorithm. Basically, the algorithm computes distance between points on a selected region of the cortical surface which serve as "coordinates". The "coordinates" are gathered from various regions of the cortex and mapped to a two dimensional flat map. (The interested reader can have a look at the paper [9] by Schwartz et al. to find further details about the algorithm.) However, the authors state that the surface-based method has difficulties in generating accurate representations particularly in higher resolutions. One future suggestion for the algorithm is to enhance the surface-based warping algorithm in order to warp the ellipsoid surface. The paper claims that a three dimensional warping algorithm would possibly limit distortions and would make it easier to study the human cerebral cortex.

The purpose of this thesis is not to study the function of the human cerebral cortex, but rather the geodesics on various two dimensional surfaces and draw similarities between the cortical surface and certain surfaces. Instead of generating a two dimensional coordinate system as used in the surface-based warping algorithm, we want to create a geodesic map by tracing a number of geodesics on a surface. From a geodesic map, we can approximate distances on various regions of surfaces resembling the cortex which can aid in better understanding the cortical structure.

The work done with geodesics in this thesis is not only limited in studying parametrized surfaces and computational geometry. Geodesic analysis branches off into various areas of science such as computer science (computer graphics) and engineering science (computational physics). In these disciplines, many applications require computing a large number of geodesics. Thus, geodesics act as a useful tool in understanding the nature and dynamics of many physical based problems.

All background material needed throughout this thesis will be provided, including an explanation of the theory of geodesics. Since calculating exact geodesics is difficult for most surfaces, we will introduce two numerical methods that will compute approximate geodesics: Euler's method and Runge-Kutta method. In the mathematical software program Maple, we have created various codes which will numerically compute and illustrate geodesics using the numerical methods. We will use the Poincaré half plane surface as a test case to verify the accuracy of the numerical methods, since the exact geodesics and distances can be easily computed and compared with approximate values. After, we will consider several interesting surfaces such as the monkey saddle surface and the bumpy ellipsoidal surface, and later draw correlations to the human cerebral cortex. Finally, we will conclude with possible future avenues of research that could be investigated using the methods in this thesis.

Chapter 2

Geodesics and Calculations

In this chapter, we will introduce some general background material. We will discuss the theory of geodesics and later derive the differential equations of geodesics. Also, the mathematical procedure of two numerical methods, Euler's method and Runge-Kutta method, will be explained in detail. These numerical methods will be used to calculate the solutions of differential equations of geodesics for various surfaces in Maple.

2.1 Background Material

We will define three types of subsets of R^n which we will need: curves, surfaces and patches.

First, we will characterize certain subsets of $\mathbf{R}^n$ that are one dimensional and where methods of differential calculus can be applied. These subsets are defined as images of differentiable functions. "Differentiable" means that the functions are continuous for all derivative orders.

Definition 2. 1. 1 *A parametrized differentiable curve is a differentiable map* $\alpha : I \to \boldsymbol{R}^n$ *of an open interval* $I = (a, b)$ *of the real line* $\boldsymbol{R}$ *into* $\boldsymbol{R}^n$.

We will focus on curves that are mappings of $\mathbf{R}$ into $\mathbf{R}^2$ or $\mathbf{R}^3$. We want to analyze regular curves.

Definition 2. 1. 2 *A parametrized differentiable curve* $\alpha(t) : I \to \boldsymbol{R}^2$ *or* $\boldsymbol{R}^3$

is said to be regular if $\alpha'(t) \neq 0$ for all $t \in I$.

Next, we will consider the subsets of $\mathbf{R}^n$ that are two dimensional.

Definition 2. 1. 3 *A parametrized differentiable surface is a differentiable map $\boldsymbol{x} : U \subseteq \boldsymbol{R}^2 \rightarrow \boldsymbol{R}^n$ of an open subset U of $\boldsymbol{R}^2$ into $\boldsymbol{R}^n$.*

We will focus on surfaces that are mappings of $U \subseteq \mathbf{R}^2$ into $\mathbf{R}^3$. We will work with regular surfaces.

Definition 2. 1. 4 *A subset $M \subset \boldsymbol{R}^3$ is a regular surface if for each $p \in M$ there exists a neighbourhood V of p in $\boldsymbol{R}^3$ and a map $\boldsymbol{x} : U \rightarrow \boldsymbol{R}^n$ of an open set $U \subset \boldsymbol{R}^2$ onto $V \cap M \subset \boldsymbol{R}^3$ such that:*

1. *$\boldsymbol{x}$ is differentiable.*
2. *$\boldsymbol{x} : U \rightarrow V \cap M$ is a homeomorphism. This means that $\boldsymbol{x}$ has a continuous inverse $\boldsymbol{x}^{-1} : V \cap M \rightarrow U$ such that $\boldsymbol{x}^{-1}$ is the restriction to $V \cap M$ of a continuous map $F : W \rightarrow \boldsymbol{R}^2$, where W is an open subset of $\boldsymbol{R}^3$ that contains $V \cap M$.*
3. *Each map $\boldsymbol{x} : U \rightarrow M$ is a regular patch.*

Finally, we will define a special function of two variables.

Definition 2. 1. 5 *A patch or local surface is a differentiable mapping*

$$\boldsymbol{x} : U \rightarrow \boldsymbol{R}^n,$$

where U is an open subset of $\boldsymbol{R}^2$. More generally, if A is any subset of R^2, the map $\boldsymbol{x} : A \to \boldsymbol{R}^3$ is a patch provided that $\boldsymbol{x}$ can be extended to a differentiable mapping from U into R^3, where U is an open set containing A. The trace or image of $\boldsymbol{x}$ is denoted by $\boldsymbol{x}(U)$.

Now, we will characterize two types of patches: regular and injective.

Definition 2. 1. 6 *A regular patch is a patch $\boldsymbol{x} : U \subseteq \boldsymbol{R}^2 \to \boldsymbol{R}^3$ for which the Jacobian matrix $\jmath(\boldsymbol{x})(u, v)$ has rank 2 for all $(u, v) \in U$. An injective patch is a patch such that $\boldsymbol{x}(u_1, v_1) = \boldsymbol{x}(u_2, v_2)$ implies that $u_1 = u_2$ and $v_1 = v_2$.*

Note that there are regular patches which are not injective and vice versa. For example, the cylinder defined parametrically by $\mathbf{x}(u, v) = (\cos(u), \sin(u), v)$ with $u \in (-\infty, \infty)$ and $v \in (-1, 1)$ is a regular patch, but not an injective patch. Also, the function defined parametrically by $\mathbf{x}(u, v) = (u^3, v^3, uv)$ for $u, v \in (-1, 1)$ is an injective patch, but not a regular patch.

A useful criterion for regular patches stems from the following lemma.

Lemma 2. 1. 1 *A patch $\boldsymbol{x} : U \subseteq \boldsymbol{R}^2 \to \boldsymbol{R}^3$ is regular at $(u_0, v_0) \in U$ if and only if $\boldsymbol{x}_u \times \boldsymbol{x}_v$ is nonzero at (u_0, v_0).*

Proof. Refer to Gray p. 192.

The next lemma introduces a property of an injective patch.

Lemma 2. 1. 2 *Let $\boldsymbol{x} : U \to \boldsymbol{R}^3$ be an injective patch. The vector field $(u, v) \mapsto \boldsymbol{x}_u \times \boldsymbol{x}_v$, if nonzero, is everywhere perpendicular to $\boldsymbol{x}(U)$.*

Proof. Refer to Gray p. 193.

Now, we will describe some aspects of the geometry of a surface in $\mathbf{R}^3$ in more detail. We start by calculating how a regular surface M bends in $\mathbf{R}^3$. The bending of a surface is tracked by estimating the change of the surface normal $\mathbf{N}$ from point to point.

Definition 2. 1. 7 *For an injective patch $\boldsymbol{x} : U \longrightarrow \boldsymbol{R}^3$ the unit normal vector field or surface normal $\boldsymbol{N}$ is given by*

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

at those points $(u, v) \in U$ at which $\boldsymbol{x}_u \times \boldsymbol{x}_v$ does not vanish.

We can also describe the geometry of a surface by investigating distance. For a regular patch, the metric is

$$ds^2 = Edu^2 + 2Fdudv + Gdv^2,$$

for some functions E, F and G that we will now define.

Definition 2. 1. 8 *Let $\boldsymbol{x} : U \rightarrow \boldsymbol{R}^3$ be a regular patch, a mapping of an open set U into R^3. The functions $E, F, G : U \rightarrow \boldsymbol{R}$ are given by*

$$E = \boldsymbol{x}_u \cdot \boldsymbol{x}_u = \ |\boldsymbol{x}_u|^2, \quad F = \boldsymbol{x}_u \cdot \boldsymbol{x}_v, \quad G = \boldsymbol{x}_v \cdot \boldsymbol{x}_v = \ |\boldsymbol{x}_v|^2.$$

The metric on the regular patch is defined by $ds^2 = Edu^2 + 2Fdudv + Gdv^2$ and is called the first fundamental form of the patch $\boldsymbol{x}$ induced from $\boldsymbol{R}^3$. The functions E, F and G are called the coefficients of the first fundamental form.

Lemma 2. 1. 3 *Let $\alpha : (a, b) \rightarrow \boldsymbol{R}^3$ be a curve that lies on a regular injective* *patch $\boldsymbol{x}: U \to \boldsymbol{R}^3$. The arc length function l of α starting at $\alpha(c)$ is given by*

$$l(t) = \int_c^t \sqrt{E(\frac{du}{dt})^2 + 2F\frac{du}{dt}\frac{dv}{dt} + G(\frac{du}{dt})^2}\ dt.$$

Proof. Refer to Gray p. 253.

The distance between two points is the length of the shortest curve joining them.

Now, we will define the coefficients of the second fundamental form.

Definition 2. 1. 9 *Let $\boldsymbol{x}: U \to \boldsymbol{R}^3$ be a regular patch. The functions e, f and g defined as*

$$e = -\boldsymbol{N}_u \cdot \boldsymbol{x}_u = \boldsymbol{N} \cdot \boldsymbol{x}_{uu}$$

$$f = -\boldsymbol{N}_v \cdot \boldsymbol{x}_u = \boldsymbol{N} \cdot \boldsymbol{x}_{uv} = \boldsymbol{N} \cdot \boldsymbol{x}_{vu} = -\boldsymbol{N}_u \cdot \boldsymbol{x}_v$$

$$g = -\boldsymbol{N}_v \cdot \boldsymbol{x}_v = \boldsymbol{N} \cdot \boldsymbol{x}_{vv}$$

are called the coefficients of the second fundamental form.

2.2 Theory of Geodesics

Our ultimate goal is to study the "straight lines" of differential geometry. We will begin to describe the theory of geodesics.

Suppose that $\alpha(t)$ is a curve with unit speed, $v(t) = |\alpha'(t)| = 1$, on the surface M in $\mathbf{R}^3$. There are two perpendicular unit vectors: $\mathbf{T} = \alpha'$ (unit tangential vector) and $\mathbf{N}$ (unit normal vector on M). Also, there is a third unit vector obtained by the cross product of $\mathbf{T}$ and $\mathbf{N}$: $\mathbf{T} \times \mathbf{N}$. All three unit vectors are perpendicular to one another and form a basis of $\mathbf{R}^3$. Thus, any vector is a linear combination of these three unit vectors.

The acceleration vector of α can be written as

$$\alpha'' = A\mathbf{T} + B(\mathbf{T} \times \mathbf{N}) + C\mathbf{N},$$

where the coefficients are given by $A = \alpha'' \cdot \mathbf{T}$, $B = \alpha'' \cdot \mathbf{T} \times \mathbf{N}$ and $C = \alpha'' \cdot \mathbf{N}$.

Substituting in the coefficients A, B and C, we obtain

$$\alpha'' = (\alpha'' \cdot \mathbf{T})\mathbf{T} + (\alpha'' \cdot \mathbf{T} \times \mathbf{N})(\mathbf{T} \times \mathbf{N}) + (\alpha'' \cdot \mathbf{N})\mathbf{N}.$$

Since α has unit speed, differentiating $\mathbf{v}^2 = \alpha' \cdot \alpha' = 1$, it will give us $\alpha'' \cdot \alpha' + \alpha' \cdot \alpha'' = 0$. Thus, $\alpha' \cdot \alpha'' = \mathbf{T} \cdot \alpha'' = 0$.

Now, α'' can be written with no $\mathbf{T}$ component:

$$\alpha'' = (\alpha'' \cdot \mathbf{T} \times \mathbf{N})(\mathbf{T} \times \mathbf{N}) + (\alpha'' \cdot \mathbf{N})\mathbf{N}.$$

We will investigate the first term of α''. We have that $\mathbf{T} \times \mathbf{N}$ is in the tangent plane of M for all $p \in M$ and

$$\begin{aligned} \alpha'' \cdot \mathbf{T} \times \mathbf{N} = \mathbf{N} \cdot \alpha'' \times \alpha' & \\ &= |\mathbf{N}||\alpha'' \times \alpha'| \cos(\theta) \\ &= |\alpha'' \times \alpha'| \cos(\theta) \\ &= \kappa_\alpha \cos(\theta), \end{aligned}$$

where κ_α is the curvature of α and θ is the angle between $\alpha'' \times \alpha'$ and $\mathbf{N}$. The above expression is referred to as the geodesic curvature of α,

$$\kappa_g = \kappa_\alpha cos(\theta).$$

Next, we are can decompose the acceleration α'' into tangential and normal components:

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

Finally, we can state the definition of a geodesic.

Definition 2. 2. 1 *A curve α on the surface M with $\alpha''_{tan} = 0$ is called a geodesic.*

Thus, the geodesics of a surface will be the "straight lines" of differential geometry. Also, the following lemma gives an important fact about geodesics.

Lemma 2. 2. 1 *A geodesic has constant speed.*

Proof. The speed of α is $v = |\alpha'|$, so $v^2 = \alpha' \cdot \alpha'$. Differentiating v, we get

$$2vv' = \alpha'' \cdot \alpha' + \alpha' \cdot \alpha'' = 2\alpha' \cdot \alpha'' = 0,$$

since $\alpha'' = \alpha''_{normal}$ and $\alpha''_{normal} \cdot \alpha' = 0$. Thus, $v' = 0$ meaning that v is a constant. □

Note that if a surface M contains a straight line $\alpha(t) = p + tq$, then that line must be a geodesic because $\alpha'' = 0$.

The reverse is also true in the following context.

Suppose that P is a plane with a unit normal vector $\mathbf{N}$ and α is a geodesic in P. By the definition, $\alpha''_{tan} = 0$; thus, $\alpha'' = (\alpha'' \cdot \mathbf{N})\mathbf{N}$. However, $\alpha' \cdot \mathbf{N} = 0$, since α lies in P.

Differentiating $\alpha' \cdot \mathbf{N} = 0$, it yields

$$0 = (\alpha' \cdot \mathbf{N})' = \alpha'' \cdot \mathbf{N} + \alpha' \cdot \mathbf{N}' = \alpha'' \cdot \mathbf{N},$$

since $\mathbf{N}$ is a constant vector. Both the tangential and normal components of α'' vanish meaning that $\alpha'' = 0$. Thus, α must be a straight line.

2.3 Differential Equations for Geodesics

Now, we will calculate geodesics.

First, we will define define eight functions of the coefficients of the first fundamental form which are called Christoffel symbols.

Definition 2. 3. 1 *Let $\boldsymbol{x} : U \to \boldsymbol{R}^n$ be a regular patch. The Christoffel symbols Γ^i_{jk}, $i, j, k = 1, 2$, corresponding to $\boldsymbol{x}$ are given by*

$$\Gamma^1_{11} = \frac{GE_u - 2FF_u + FE_v}{2(EG - F^2)}, \qquad \Gamma^2_{11} = \frac{2EF_u - EE_v - FE_u}{2(EG - F^2)},$$

$$\Gamma^1_{12} = \frac{GE_v + FG_u}{2(EG - F^2)}, \qquad \Gamma^2_{12} = \frac{EG_u - FE_v}{2(EG - F^2)},$$

$$\Gamma^1_{22} = \frac{2GF_v - 2GG_u - FG_v}{2(EG - F^2)}, \qquad \Gamma^2_{22} = \frac{EG_v - 2FF_v + FG_u}{2(EG - F^2)},$$

and $\Gamma^1_{21} = \Gamma^1_{12}$, $\Gamma^2_{21} = \Gamma^2_{12}$.

Let $\alpha = \mathbf{x}(u(t), v(t))$ be a curve on the surface M. Suppose that $\alpha' = u'\mathbf{x}_u + v'\mathbf{x}_v$. Taking the derivative of α', we get

$$\alpha'' = u''\mathbf{x}_u + u'(\frac{d}{dt}\mathbf{x}_u) + v''\mathbf{x}_v + v'(\frac{d}{dt}\mathbf{x}_v)$$

Now,

$$\frac{d}{dt}\mathbf{x}_u(u, v) = u'\mathbf{x}_{uu} + v'\mathbf{x}_{uv}$$

and similarily,

$$\frac{d}{dt}\mathbf{x}_v(u, v) = u'\mathbf{x}_{vu} + v'\mathbf{x}_{vv}.$$

Using Gauss Equations, we can calculate the geodesic curvature of α.

Theorem 2. 3. 1 *Let $\boldsymbol{x}$ be a regular patch in $\boldsymbol{R}^3$ with the surface normal $\boldsymbol{N}$. Then,*

$$\boldsymbol{x}_{uu} = \Gamma^1_{11}\boldsymbol{x}_u + \Gamma^2_{11}\boldsymbol{x}_v + \boldsymbol{N}e,$$

$$\boldsymbol{x}_{uv} = \Gamma^1_{12}\boldsymbol{x}_u + \Gamma^2_{12}\boldsymbol{x}_v + \boldsymbol{N}f = \boldsymbol{x}_{vu},$$

$$\boldsymbol{x}_{vv} = \Gamma^1_{22}\boldsymbol{x}_u + \Gamma^2_{22}\boldsymbol{x}_v + \boldsymbol{N}g,$$

where the functions e, f and g are the coefficients of the second fundamental form.

Proof. Refer to Gray p. 398-399.

A straightforward calculation shows that

$$\alpha'' = (u'' + u'^2\Gamma^1_{11} + 2u'v'\Gamma^1_{12} + v'^2\Gamma^1_{22})\,\mathbf{x}_u \;+$$

$$(v'' + u'^2\Gamma^2_{11} + 2u'v'\Gamma^2_{12} + v'^2\Gamma^2_{22})\,\mathbf{x}_v \;+\; d\,\mathbf{N},$$

where the coefficient d of $\mathbf{N}$ contains the functions e, f and g. From the above calculations, the following lemma is an immediate consequence.

Lemma 2. 3. 1 *Let $M \subset \boldsymbol{R}^3$ be a surface parametrized by a regular patch $\boldsymbol{x} : U \to \boldsymbol{R}^3$, where $U \subset \boldsymbol{R}^2$. Then, the geodesics on M are determined by two second order differential equations:*

$$u'' + \Gamma^1_{11}u'^2 + 2\Gamma^1_{12}u'v' + \Gamma^1_{22}v'^2 = 0,$$

$$v'' + \Gamma^2_{11}u'^2 + 2\Gamma^2_{12}u'v' + \Gamma^2_{22}v'^2 = 0,$$

where Γ^i_{jk} are Christoffel symbols of $\boldsymbol{x}$.

Next, we can derive one special case.

Definition 2. 3. 1 *Let M be a surface with metric $ds^2 = Edu^2 + 2Fdudv + Gdv^2$. A Clairaut patch (also called a u-Clairaut patch) on M is a patch $\boldsymbol{x} : U \to M$ for which*

$$E_u = G_u = F = 0.$$

The following lemma is a consequence of Definition 2. 3. 1.

Lemma 2. 3. 2 [6] *For a Clairaut patch with $ds^2 = Edu^2 + Gdv^2$, the Christoffel symbols are given by*

$$\Gamma^1_{11} = 0, \qquad \Gamma^2_{11} = \frac{-E_v}{2G},$$

$$\Gamma^1_{12} = \frac{E_v}{2E}, \qquad \Gamma^2_{12} = 0,$$

$$\Gamma^1_{22} = 0, \qquad \Gamma^2_{22} = \frac{G_v}{2G}.$$

In this case, the differential equations of geodesics reduce to

$$u'' + \frac{E_v}{E}u'v' = 0,$$

$$v'' - \frac{E_v}{2G}u'^2 + \frac{G_v}{2G}v'^2 = 0.$$

An example of a Clairaut patch is a sphere. A parametrization of the sphere

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

where $0 \le u \le 2\pi$ and $0 \le v \le \pi$. It is easy to compute that $E = \cos^2(v)$, $F = 0$ and $G = 1$. Furthermore, $E_u = G_u = 0$, $E_v = -2\cos(v)\sin(v)$ and $G_v = 0$.

Thus, the differential equations of geodesics for the sphere become

$$u'' - 2\tan(v)u'v' = 0,$$

$$v'' + \sin(v)\cos(v)u'^2 = 0.$$

All the calculations above are coded in Maple. (See Appendix A and Appendix B)

2.4 Numerical Methods Used to Solve Geodesics Equations

For many surfaces, the differential equations of geodesics are complicated and/or cannot be solved explicitly. We will find numerical approximations of solutions of the differential equations. Two particular methods will be used: Euler's method and Runge-Kutta method of fourth order.

Recall. For the first order problem $y' = f(x, y)$ with the initial value $y(x_0) = y_0$, we want to find approximate values of the solution at $x_{n+1} = x_n + h$, where h is the step size. The iterative Euler's formula is

$$y_{n+1} = y_n + h \cdot f(x_n, y_n),$$

where $n \geq 0$ and (x_0, y_0) is the initial value. The formula calculates the successive approximations y_n to the exact values of $y(x_n)$ of the solution $y = y(x)$ at the points x_n respectively. [3] □

We will take the second order differential equations of geodesics,

$$u'' + \Gamma^1_{11}u'^2 + 2\Gamma^1_{12}u'v' + \Gamma^1_{22}v'^2 = 0,$$

$$v'' + \Gamma^2_{11}u'^2 + 2\Gamma^2_{12}u'v' + \Gamma^2_{22}v'^2 = 0,$$

and transform them into a fourth order system of first order initial value problem,

$$\begin{aligned}
u' &= p \\
v' &= q \\
p' &= -\Gamma^1_{11}u'^2 - 2\Gamma^1_{12}u'v' - \Gamma^1_{22}v'^2 \\
q' &= -\Gamma^2_{11}u'^2 - 2\Gamma^2_{12}u'v' - \Gamma^2_{22}v'^2,
\end{aligned}$$

where $(u(0), v(0)) = (u_0, v_0)$ is the initial point and $(p(0), q(0)) = (p_0, q_0)$ is the initial direction.

Now, we can modify this system of first order equations into the Euler's method iterative formula:

$$\begin{aligned}
u_{n+1} &= u_n + h \cdot p_n \\
v_{n+1} &= v_n + h \cdot q_n \\
p_{n+1} &= p_n + h \cdot (-\Gamma^1_{11}p_n^2 - 2\Gamma^1_{12}p_n q_n - \Gamma^1_{22}q_n^2) \\
q_{n+1} &= q_n + h \cdot (-\Gamma^2_{11}p_n^2 - 2\Gamma^2_{12}p_n q_n - \Gamma^2_{22}q_n^2),
\end{aligned}$$

where $n \geq 0$, (u_0, v_0) is the initial point, (p_0, q_0) is the initial unit vector, and $h > 0$ is the step size. (See Appendix C)

Runge-Kutta method of fourth order (RK4) is similar to Euler's method, but more precise in computation. [1] The general idea of RK4 is to weigh the average of the slopes at the midpoint of each interval, $[x_n, x_{n+1}]$.

Recall. For the first order initial value problem $y' = f(x, y)$ with the initial value $y(x_0) = y_0$, we define y_{n+1} using the following iteration formula:

$$y_{n+1} \simeq y_n + \frac{h}{6}(y'(x_n) + 2y'(x_n + \frac{h}{2}) + 2y'(x_n + \frac{h}{2}) + y'(x_{n+1})),$$

where h is the step size. The formula calculates approximate values of the solution at $x_{n+1} = x_n + h$. We replace $y'(x_n)$, $2y'(x_n + \frac{h}{2})$, $2y'(x_n + \frac{h}{2})$, and $y'(x_{n+1})$ with the following estimates:

- $k_1 = f(x_n, y_n)$ which is the slope at x_n;
- $k_2 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}h \cdot k_1)$ which is the slope at the midpoint of $[x_n, x_{n+1}]$ using k_1 to determine the y-value;
- $k_3 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}h \cdot k_2)$ which is the slope at the midpoint of $[x_n, x_{n+1}]$ using k_2 to determine the y-value;
- $k_4 = f(x_n + h, y_n + h \cdot k_3)$ which is the slope at x_{n+1} using k_3 to determine the y-value.

Substituting, we get the following iterative Runge-Kutta formula:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

If $K = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ is the weighted average of the four slopes, then we have the iterative Euler's formula

$$y_{n+1} = y_n + h \cdot K. \qquad [3]\square$$

The differential equations of geodesics is a system of four first order differential equations. We write it as,

$$u' = f_1(t, u, v, p, q) = p$$

$$v' = f_2(t, u, v, p, q) = q$$

$$p' = f_3(t, u, v, p, q) = -\Gamma^1_{11}u'^2 - 2\Gamma^1_{12}u'v' - \Gamma^1_{22}v'^2$$

$$q' = f_4(t, u, v, p, q) = -\Gamma^2_{11}u'^2 - 2\Gamma^2_{12}u'v' - \Gamma^2_{22}v'^2,$$

and convert into the iterative Runge-Kutta formula as follows. In each step, we compute ($n \geq 0$)

$$k_{11} = h \cdot f_1(t_n, u_n, v_n, p_n, q_n)$$

$$k_{12} = h \cdot f_2(t_n, u_n, v_n, p_n, q_n)$$

$$k_{13} = h \cdot f_3(t_n, u_n, v_n, p_n, q_n)$$

$$k_{14} = h \cdot f_4(t_n, u_n, v_n, p_n, q_n)$$

$$k_{21} = h \cdot f_1(t_n + \frac{h}{2}, u_n + \frac{k_{11}}{2}, v_n + \frac{k_{12}}{2}, p_n + \frac{k_{13}}{2}, q_n + \frac{k_{14}}{2})$$
$$k_{22} = h \cdot f_2(t_n + \frac{h}{2}, u_n + \frac{k_{11}}{2}, v_n + \frac{k_{12}}{2}, p_n + \frac{k_{13}}{2}, q_n + \frac{k_{14}}{2})$$
$$k_{23} = h \cdot f_3(t_n + \frac{h}{2}, u_n + \frac{k_{11}}{2}, v_n + \frac{k_{12}}{2}, p_n + \frac{k_{13}}{2}, q_n + \frac{k_{14}}{2})$$
$$k_{24} = h \cdot f_4(t_n + \frac{h}{2}, u_n + \frac{k_{11}}{2}, v_n + \frac{k_{12}}{2}, p_n + \frac{k_{13}}{2}, q_n + \frac{k_{14}}{2})$$

$$k_{31} = h \cdot f_1(t_n + \frac{h}{2}, u_n + \frac{k_{21}}{2}, v_n + \frac{k_{22}}{2}, p_n + \frac{k_{23}}{2}, q_n + \frac{k_{24}}{2})$$
$$k_{32} = h \cdot f_2(t_n + \frac{h}{2}, u_n + \frac{k_{21}}{2}, v_n + \frac{k_{22}}{2}, p_n + \frac{k_{23}}{2}, q_n + \frac{k_{24}}{2})$$
$$k_{33} = h \cdot f_3(t_n + \frac{h}{2}, u_n + \frac{k_{21}}{2}, v_n + \frac{k_{22}}{2}, p_n + \frac{k_{23}}{2}, q_n + \frac{k_{24}}{2})$$
$$k_{34} = h \cdot f_4(t_n + \frac{h}{2}, u_n + \frac{k_{21}}{2}, v_n + \frac{k_{22}}{2}, p_n + \frac{k_{23}}{2}, q_n + \frac{k_{24}}{2})$$

$$k_{41} = h \cdot f_1(t_n + h, u_n + k_{31}, v_n + k_{32}, p_n + k_{33}, q_n + k_{34})$$

$$k_{42} = h \cdot f_2(t_n + h, u_n + k_{31}, v_n + k_{32}, p_n + k_{33}, q_n + k_{34})$$

$$k_{43} = h \cdot f_3(t_n + h, u_n + k_{31}, v_n + k_{32}, p_n + k_{33}, q_n + k_{34})$$

$$k_{44} = h \cdot f_4(t_n + h, u_n + k_{31}, v_n + k_{32}, p_n + k_{33}, q_n + k_{34}),$$

Then,

$$u_{n+1} = u_n + \frac{1}{6} \cdot (k_{11} + 2 \cdot k_{21} + 2 \cdot k_{31} + k_{41})$$

$$v_{n+1} = v_n + \frac{1}{6} \cdot (k_{12} + 2 \cdot k_{22} + 2 \cdot k_{32} + k_{42})$$

$$p_{n+1} = p_n + \frac{1}{6} \cdot (k_{13} + 2 \cdot k_{23} + 2 \cdot k_{33} + k_{43})$$

$$q_{n+1} = q_n + \frac{1}{6} \cdot (k_{14} + 2 \cdot k_{24} + 2 \cdot k_{34} + k_{44}),$$

where $t_{n+1} = t_n + h$ and $n \geq 0$. (See Appendix D)

Throughout this thesis, Euler's method and Runge-Kutta of fourth order method will be used to present several numerical results of differential equations of geodesics. All the numerical results are coded in Maple 11 and can be referenced in the Appendices. All computational results were obtained using a MacIntosh computer with a 2.33GHz Intel Core 2 Duo CPU and 2 GB of memory. We will test the accuracy and speed of the numerical methods in the next couple of chapters of this thesis.

Chapter 3

Test Case: Poincaré Half Plane

In this chapter, we introduce the Poincaré half plane and its metric. It will serve as a test case to check the accuracy of Euler's method and Runge-Kutta method of fourth order when computing geodesics in Maple. Later, these numerical methods will provide us with an efficient way of calculating geodesics on different surfaces. After defining the Poincaré metric, we will study geodesics and derive the distance formula. Being able to draw geodesic circles in Maple, we will illustrate a case of metric fibration.

3.1 Definition of the Poincaré Half Plane

In non-Euclidean geometry, the Poincaré half plane is the upper half-plane,

$$\mathbf{R}^2_+ = \{(u,v) \in \mathbf{R}^2 : v > 0\},$$

with the Poincaré metric,

$$ds^2 = \frac{du^2 + dv^2}{v^2}.$$

A straightforward calculation shows that the coefficients of the first fundamental form are $E = G = \frac{1}{v^2}$ and $F = 0$. Furthermore, $E_u = G_u = 0$ and $E_v = G_v = -\frac{2}{v^3}$. Thus, $(\mathbf{R}^2_+, ds^2)$ is a Clairaut patch.

3.2 Geodesics in the Poincaré Half Plane

In the Poincaré half plane, there are only two types of curves that can be geodesics.

Theorem 3. 2. 1 [8] *The geodesics of the Poincaré metric on the half plane* $\boldsymbol{R}_+^2$ *are the following: vertical lines and circular arcs centred on the u-axis.*

Proof. Let $\alpha(t) = (u(t), v(t))$ be a unit speed geodesic, ie. $|\alpha'(t)|^2 = Eu'^2 + Fv'^2 = 1$.

Since $G_u = E_u = F = 0$, the Poincaré half plane is a Clairaut patch and the differential equations of geodesics are given by Lemma 2.3.2:

$$u'' - \frac{2}{v}u'v' = 0$$

$$v'' + \frac{1}{v}u'^2 - \frac{1}{v}v'^2 = 0.$$

From the first geodesic equation, assuming $u' \neq 0$, we obtain

$$\int \frac{u''}{u'}\mathrm{d}t = \int \frac{2}{v}v'\mathrm{d}t$$

$$\ln(u') = 2\ \ln(v) + c,$$

where c is a constant. Thus,

$$e^{\ln(u')} = 2\, e^{\ln(v)+c}$$

$$u' = Av^2,$$

where $A = e^c$.

If $u' = 0$, then $u =$ constant $(v > 0)$ which is a vertical line. Thus, this proves that a geodesic can be a vertical line.

Now, from the unit speed relation, we obtain

$$\frac{u'^2 + v'^2}{v^2} = 1 \qquad \text{or} \qquad u'^2 + v'^2 = v^2.$$

Plugging in $u' = Av^2$ into the unit speed relation, we have

$$A^2v^4 + v'^2 = v^2$$

$$v'^2 = v^2 - A^2v^4$$
$$v' = v\sqrt{1 - A^2v^2}.$$

Dividing v' by u', it yields a seperable differential equation which we will integrate by substitution.

$$\frac{v'}{u'} = \frac{v\sqrt{1-A^2v^2}}{Av^2} = \frac{\sqrt{1-A^2v^2}}{Av}$$
$$u' = \frac{Av}{\sqrt{1-A^2v^2}}v'$$
$$\int \mathrm{d}u = \int \frac{Av}{\sqrt{1-A^2v^2}}\mathrm{d}v.$$

Substituting $z = 1 - A^2v^2$, $\mathrm{d}z = -2A^2v\ \mathrm{d}v$, we get

$$u - d = -\frac{1}{A}\sqrt{1-A^2v^2},$$

where d is a constant.

Simplifying the result gives us

$$(u-d)^2 + v^2 = \frac{1}{A^2},$$

which is an equation of a circle centred on the u-axis. Thus, in this case, the geodesic is a circular arc centred on the u-axis. □

3.3 Distance Formula

We now can calculate the distance between two points on a geodesic.

Let $A_1(u_1, v_1)$ and $A_2(u_2, v_2)$ be the two given points on a geodesic. By integrating the Poincaré metric along the geodesic, we obtain the distance formula:

$$d(A_1, A_2) = \begin{cases} \left|\ln \frac{v_2}{v_1}\right| & \text{if } u_2 = u_1 \\ \left|\ln \frac{\tan(\frac{t_1}{2})}{\tan(\frac{t_2}{2})}\right| & \text{if } u_2 \neq u_1 \end{cases}$$

where t_i is the angle between the vectors $\overrightarrow{QA_i}$ and the positive u-axis, $i = 1, 2$. The point Q lies on the u-axis and is the centre of the Euclidean circle through the points A_1 and A_2. [8]

We will derive the distance formula.

Case 1. Assume that $u_2 = u_1$. This means that the value of u is fixed and the geodesic is a vertical line.

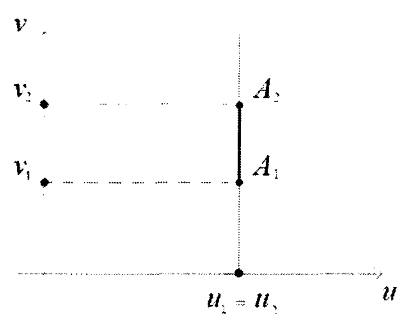

Figure 1: Representation of Case 1.

Let $\alpha(t) = (u_1, v_1) + t(u_2 - u_1, v_2 - v_1)$ be a parametrization of a straight line segment between A_1 and A_2, where $0 \leq t \leq 1$. By the assumption $u_2 = u_1$, we obtain

$$\alpha(t) = (u_1, v_1 + t(v_2 - v_1)).$$

It follows that

$$\alpha'(t) = (0, v_2 - v_1),$$

and

$$|\alpha'(t)|^2 = \frac{(v_2 - v_1)^2}{v^2} = \frac{(v_2 - v_1)^2}{(v_1 + t(v_2 - v_1))^2}.$$

The length of the line segment is calculated by

$$
\begin{aligned}
l &= \int_0^1 |\alpha'(t)| \ \mathrm{d}t \\
&= \int_0^1 \frac{(v_2 - v_1)}{v_1 + t(v_2 - v_1)} \ \mathrm{d}t,
\end{aligned}
$$

where $z = v_1 + t(v_2 - v_1)$, $\mathrm{d}z = (v_1 - v_1) \ \mathrm{d}t$,

$$
\begin{aligned}
&= \ln |v_1 + t(v_2 - v_1)| \Big|_0^1 \\
&= \ln |v_2| - \ln |v_1| = \ln \left|\frac{v_2}{v_1}\right|,
\end{aligned}
$$

assuming that $v_2 > v_1$. In general, distance in this case is $\left| \ln \frac{v_2}{v_1} \right|$. □

Case 2. Assume that $u_2 \neq u_1$. This means that A_1 and A_2 lie on the a circular arc.

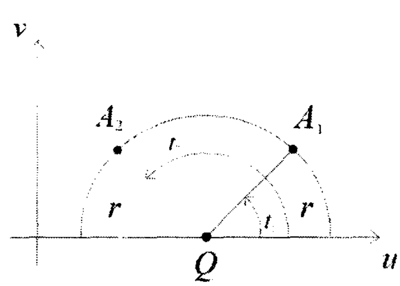

Figure 2: Representation of Case 2.

Let $\alpha(t) = (q + r\cos(t), r\sin(t))$ be a circular arc connecting A_1 and A_2, where $t_1 \leq t \leq t_2$, r is the radius and $Q = (q, 0)$ is the centre of a circular arc on the u-axis (See Figure 2). Then,

$$\alpha'(t) = (-r\sin(t), r\cos(t)).$$

Thus, we obtain

$$|\alpha'(t)|^2 = \frac{(-r\sin(t))^2 + (r\cos(t))^2}{(r\sin(t))^2}$$

$$= 1 + \cot^2(t) = \csc^2(t).$$

The length of the circular segment is computed to be

$$\begin{aligned}
l &= \int_{t_1}^{t_2} |\alpha'(t)| \ \mathrm{d}t \\
&= \int_{t_1}^{t_2} \csc(t) \ \mathrm{d}t \\
&= \ln|\csc(t) - \cot(t)|\Big|_{t_1}^{t_2} \\
&= \ln\left(\frac{1-\cos(t)}{\sin(t)}\right)\Big|_{t_1}^{t_2} = \ln\left(\tan\left(\frac{t}{2}\right)\right)\Big|_{t_1}^{t_2} \\
&= \ln\left(\tan\left(\frac{t_2}{2}\right)\right) - \ln\left(\tan\left(\frac{t_1}{2}\right)\right) \\
&= \ln\left|\frac{\tan(\frac{t_2}{2})}{\tan(\frac{t_1}{2})}\right|,
\end{aligned}$$

assuming that $t_2 > t_1$. In general, the distance is $\left|\ln\frac{\tan(\frac{t_2}{2})}{\tan(\frac{t_1}{2})}\right|$. $\square$

This exact geodesic distance will be used to test the accuracy of our numerical methods.

3.4 Accuracy of the Numerical Methods

Now that we know what geodesics are in the Poincaré half plane, we are ready to compare numerically calculated geodesics with exact geodesics. We will examine a few cases.

Euler's Method

Pick a point $A(0,1)$ and an initial vector $\overrightarrow{v}(\cos(\theta), \sin(\theta))$, where $\theta = \frac{2\pi}{5}$. The figure below shows the comparison of the exact geodesic and the numerically calculated geodesic using Euler's method with 1000 steps and a step size 0.01.

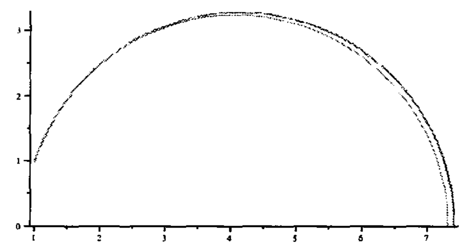

Figure 3: Comparison of exact geodesic (solid line) and numerically calculated geodesic (dotted line) using Euler's method. (h = 0.01, n = 1000 steps)

By decreasing the value of h and increasing the value of n, we obtain a better approximation. The figure below is obtained by iterating 10000 steps with a step size of 0.005.

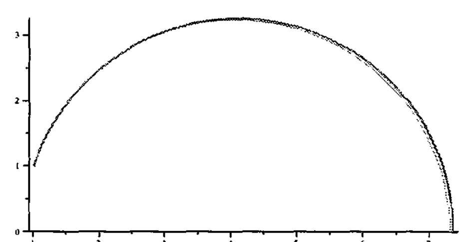

Figure 4: Comparison of exact geodesic (solid line) and numerically calculated geodesic (dotted line) using Euler's method. (h = 0.005, n = 10000 steps)

Runge-Kutta Fourth Order Method (RK4)

RK4 provides better approximations of numerically calculated geodesics than Euler's method. We can see this is true in the case of the figure below where we used the same initial conditions as in Figure 3. The numerically calculated geodesic sits upon top of the exact geodesic.

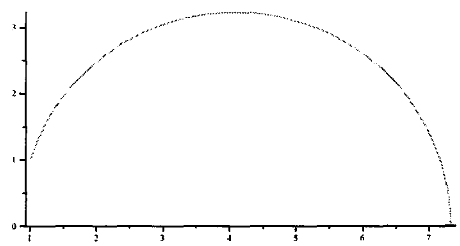

Figure 5: Comparison of exact geodesic (solid line) and numerically calculated geodesic (dotted line) using RK4 method. (h = 0.01, n = 1000 steps)

Both Euler's method and Runge-Kutta method give fairly good approximations of geodesics. RK4 method is more precise than Euler's method and is therefore used in various calculations. [3] Thus, whenever possible, we will use RK4 method.

Unfortunately, we have limitations in hardware (MacIntosh) and software (Maple) that restrict us to perform certain numerical calculations because the system either breaks down due to the exceeding memory allocation or takes a very long time to run numerical calculations which at times does not finish at all. These restrictions will be mentioned throughout this thesis when we investigate various surfaces and the various strategies to overcome computer limitations.

Distance

Now, we will show that the numerical methods can be used to estimate the distance between two points. In general, there is no explicit way to find the distance between two points on a surface. What we can do is use Euler's and RK4 methods to obtain an approximation.

Pick two points $A_1(u_1, v_1)$ (initial point) and $A_2(u_2, v_2)$. We construct a sequence of concentric geodesic circles centred at A_1 of increasing radii $h, 2h, 3h, ..., nh, ...$ where h is the step size and n is the number of steps used in either Euler's method or RK4 method. We stop when we reach the point A_2, ie. when A_2 is the region between circles of radii nh and $(n+1)h$.

To construct each circle, we do the following procedure. We generate a number of geodesics starting at A_1, where the directions are given by $(cos(\theta_i), sin(\theta_i))$, where $\theta_i = \frac{2\pi}{N}i$ and $i = 1, ..., N$. Thus, the speed of each geodesic is $\frac{1}{v_1}$. By connecting the points on each geodesic that are generated by the first iteration step, we obtain the circle of radius h. Similarly, by connecting the points on all geodesics generated in the second iteration step, we obtain the circle of radius $2h$, etc. We want to ensure that we have a sufficient number of steps to form a family of geodesic circles reaching the point A_2. If A_2 is between the geodesic circles obtained by n and $n+1$ iterations, then the approximate distance is

$$\frac{nh}{v_1} \leq d_{approx}(A_1, A_2) \leq \frac{(n+1)h}{v_1}.$$

This inequality holds true because geodesics have constant speed by Lemma 2. 2. 1.

As an example, let us estimate the distance between the points $A_1(1, 2)$ (initial point) and $A_2(1.4, 2.4)$ in the Poincaré half plane using RK4 method. The following conditions are used to construct the circles: the initial direction angles $\theta_i = \frac{2\pi}{60}i$, $i = 1, ..., 60$, the step size $h = 0.03$ and the number of steps $n = 17$.

Thus, Figure 6 shows that the approximate distance is $(n = 17)$

$$\frac{nh}{v_1} \leq d_{approx}(A_1, A_2) \leq \frac{(n+1)h}{v_1},$$

ie.

$$0.2550 \leq d_{approx}(A_1, A_2) \leq 0.2700.$$

which was calculated in Maple. (See Appendix E)

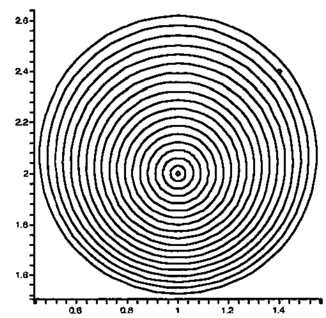

Figure 6: Distance between two points (1,2) and (1.4, 2.4) using RK4 method.

Using the distance formula from Section 3.4 and Maple to calculate it, we obtain that

$$d(A_1, A_2) = 0.257487.$$

We performed another accuracy test of the numerical methods. Using $A(1, 2)$ as an initial point, we used both Euler's method and RK4 method to calculate the geodesic in the initial direction $\theta = \frac{\pi}{3}$ with the step size $h = 0.02$ and the number of steps $n = 1000$.

In the case of Euler's method, the geodesic ended at a point $(8.535549, 0.001226)$; its distance from A is $\frac{(1000)(0.02)}{2} = 10$, compared to the exact distance of 10.119285. In the case of RK4 method, the geodesic ended at a point $(8.464101, 0.001355)$;

its distance from A is $\frac{(1000)(0.02)}{2} = 10$, compared to the exact distance of 9.999271.

We will use the numerically calculated geodesic circles to estimate the distance on other surfaces, since the approximations of Euler's method and Runge-Kutta method seem to be fairly accurate. Unfortunately, we will not be able to compare the approximate distance with the exact distance on an arbitrary surface, since there is no explicit way to calculate the distance between two points.

3.5 Metric Fibration

Disgressing a bit, we use our tools to illustrate a case of a metric fibration in the Poincaré half plane.

Definition 3. 5. 1 *A partition of a metric space by a family of congruent and mutually equidistant sets is called a metric fibration.*

According to the paper [5], there are three types of metric fibrations of hyperbolic geometry: fibration by fifth lines, fibration by horocycles, and fibration by broken horocycles. We will only focus on the fibration by horocycles. In general, a horocycle is defined as the limit of circles with the same tangent line at a given point as their radii approaches infinity. In Euclidean geometry, the limit is "the circle of infinite radius" or a straight line, and, in hyperbolic geometry, the limit is a curve represented by a circle tangent to the u-axis. [8] Thus, horocyles are a family of circles tangent to the u-axis.

Since we have developed a way to sketch geodesic circles, we can illustrate horocycles in the Poincaré half plane.

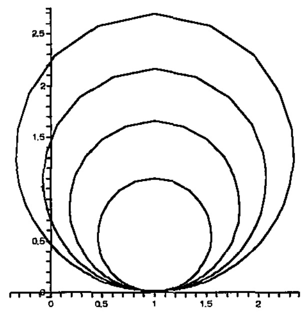

Figure 7: Representation of Horocycles.

The figure above shows a family of circles where all the their radii lie on the line $u = 1$. Each circle is obtained by shooting geodesics in a number of equally spaced directions for the same distance.

In this case, we generated four sets of geodesics, each starting at a different point on the line $u = 1$. In general, instead of using a single initial point, we can shoot many geodesics from many initial points to study a surface and to investigate the relation of every geodesics to one another. This approach is useful when we mention the phase flow method in chapter 5.

Chapter 4

Using Maple to Study Geodesics on Some Interesting Surfaces

Having developed the necessary methods to analyze geodesics, we will investigate two particular surfaces: the monkey saddle surface and the corkscrew surface. We will show that certain regions on these surfaces are similar to regions of the human cerebral cortex. We will calculate and sketch the geodesics of the monkey saddle surface and the corkscrew surface.

4.1 The Monkey Saddle Surface

The monkey saddle is a regular surface parameterized by

$$\mathbf{x}(u,v) = (u, v, u^3 - 3uv^2),$$

where $u, v \in \mathbf{R}$.

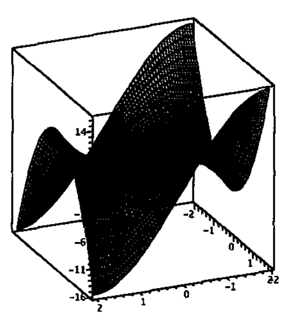

Figure 8: The Monkey Saddle Surface.

A generalized version of the monkey saddle surface is given by

$$\mathbf{x}_n(u,v) = (u, v, \Re e(u+iv)^n),$$

where $n \geq 3$ and $u, v \in \mathbf{R}$.

As mentioned earlier, we can use certain parts of the monkey saddle surface to simulate parts of the human cerebral cortex in order to better understand its structure. The figure below shows one part of the surface that can be used for that purpose.

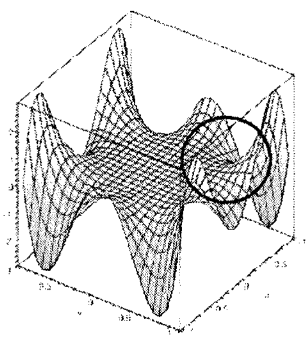

Figure 9: The Generalized Monkey Saddle Surface. $(n = 6)$

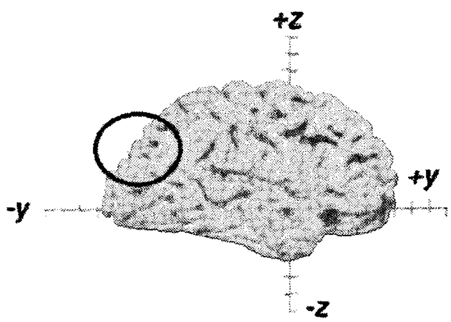

Figure 10: A 'bumpy' region of the human cerebral cortex. [12]

We will work with the monkey saddle surface parametrization

$$\mathbf{x}_3(u, v) = \mathbf{x}(u, v) = (u, v, u^3 - 3uv^2),$$

where $u, v \in \mathbf{R}$, to calculate and illustrate geodesics. In a similar way, we could calculate a generalized version of the monkey saddle surface using stronger computational tools.

The coefficients of the first fundamental form are

$$E = 1 + 9(u^2 - v^2)^2, \;\; F = -18uv(u^2 - v^2), \;\; G = 1 + 36u^2v^2,$$

which give the metric

$$ds^2 = (1 + 9(u^2 - v^2)^2)du^2 - (18uv(u^2 - v^2))du^2dv^2 + (1 + 36u^2v^2)dv^2.$$

From the Christoffel symbols,

$$\Gamma^1_{11} = 36u(u^2 - v^2), \;\; \Gamma^1_{12} = -36v(u^2 - v^2),$$

$$\Gamma^1_{22} = -18v(3u^2 - v^2), \;\; \Gamma^2_{11} = 18u(3v^2 - u^2),$$

$$\Gamma^2_{12} = 72uv^2, \;\; \Gamma^2_{22} = 72u^2v,$$

we obtain the differential equations of geodesics for a monkey saddle:

$$u'' + \frac{18u(u^2 - v^2)}{1 + 9(u^2 + v^2)^2}u'^2 - \frac{36v(u^2 - v^2)}{1 + 9(u^2 + v^2)^2}u'v' - \frac{18u(u^2 - v^2)}{1 + 9(u^2 + v^2)^2}v'^2 = 0$$

$$v'' - \frac{36u^2v}{1 + 9(u^2 + v^2)^2}u'^2 + \frac{72uv^2}{1 + 9(u^2 + v^2)^2}u'v' + \frac{36u^2v}{1 + 9(u^2 + v^2)^2}v'^2 = 0.$$

In order to solve geodesic equations numerically, we need to rewrite the equations as a fourth order system of first order equations:

$$u' = p$$

$$v' = q$$

$$p' = -\frac{18u(u^2 - v^2)}{1 + 9(u^2 + v^2)^2}p^2 + \frac{36v(u^2 - v^2)}{1 + 9(u^2 + v^2)^2}pq + \frac{18u(u^2 - v^2)}{1 + 9(u^2 + v^2)^2}q^2$$

$$q' = +\frac{36u^2v}{1 + 9(u^2 + v^2)^2}p^2 - \frac{72uv^2}{1 + 9(u^2 + v^2)^2}pq + \frac{36u^2v}{1 + 9(u^2 + v^2)^2}q^2.$$

Using Maple, we numerically calculate and illustrate the geodesics on the monkey saddle surface. Below are the results using RK4 method with the initial point $(u_0, v_0) = (0, 0)$ and the initial direction vector $\overrightarrow{v} = (\cos(\theta), \sin(\theta))$ where the initial direction angles are $\theta = \frac{2\pi}{50}i$, $i = 1, ..., 50$.

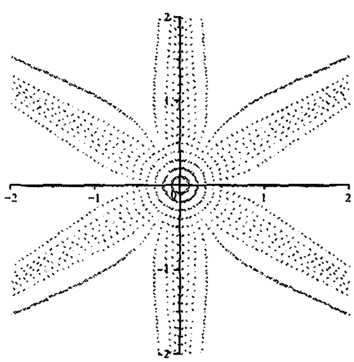

Figure 11: (The Monkey Saddle Surface) Geodesics in the uv-plane, showing actual points calculated by using RK4 method. $(h = 0.1, n = 70$ steps)

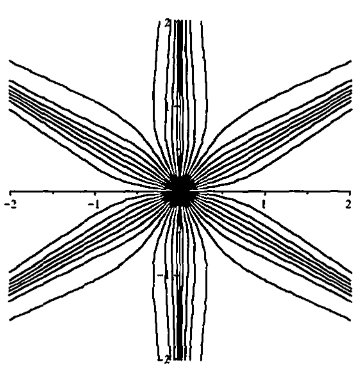

Figure 12: (The Monkey Saddle Surface) Geodesics in the uv-plane as curves using RK4 method. $(h = 0.1, n = 70$ steps)

The next figure shows the geodesics on the surface.

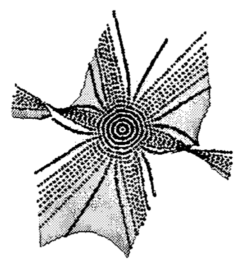

Figure 13: (The Monkey Saddle Surface) Geodesics on the surface using RK4 method. ($h = 0.1, n = 70$ steps)

By connecting all the iterated points of the same step on all the geodesics, we can produce geodesic circles on the surface. These configurations are useful in calculating distances on a surface. We can show them in two or three dimensions.

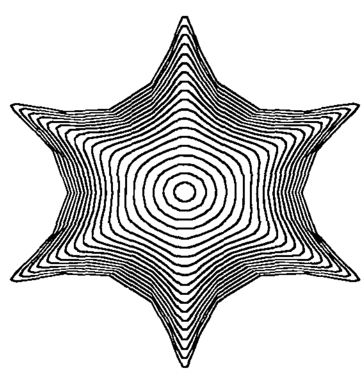

Figure 14: (The Monkey Saddle Surface) Geodesic circles in the uv-plane using RK4 method. ($h = 0.1, n = 15$ steps)

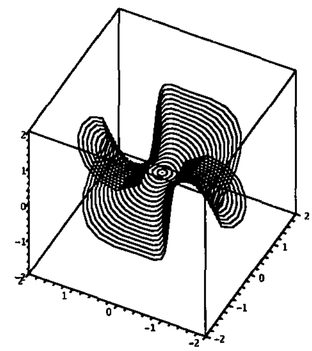

Figure 15: (The Monkey Saddle Surface) Geodesic circles on the surface using RK4 method. ($h = 0.1, n = 15$ steps)

For the purpose of using surfaces that resemble certain parts of the human cerebral cortex, we need to use initial points other than $(u_0, v_0) = (0, 0)$. The figures below show geodesics and geodesic circles with the initial point $(u_0, v_0) = (1, 1)$ and the initial direction angles $\theta = \frac{2\pi}{50}i$, $i = 1, ..., 50$.

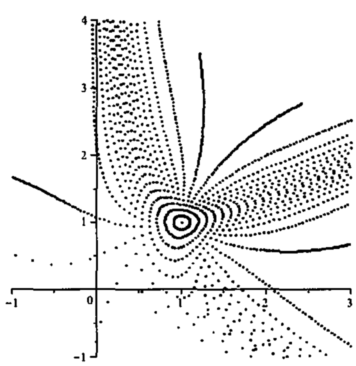

Figure 16: (The Monkey Saddle Surface) Geodesics in the uv-plane, showing actual points calculated by using RK4 method. ($h = 0.1, n = 70$ steps)

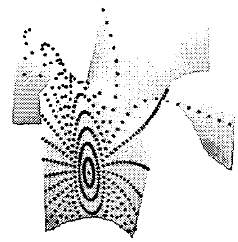

Figure 17: (The Monkey Saddle Surface) Geodesics, shown as points, on the surface using RK4 method. ($h = 0.1, n = 70$ steps)

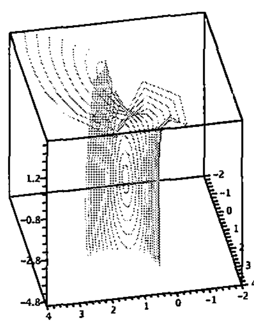

Figure 18: (The Monkey Saddle Surface) Geodesic circles on the surface using RK4 method. ($h = 0.1, n = 15$ steps)

4.2 The Corkscrew Surface

The corkscrew surface is a regular surface parametrized by

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

where a, b are constants, $u \in [0, 2\pi)$ and $v \in [-\pi, \pi]$. The surface is obtained by extending the sphere along its diameter, and then twisting it.

To obtain the differential equations of geodesics for the general corkscrew surface parametrization, we calculate the coefficients of the first fundamental form,

$$E = b^2 + a^2\cos^2(v), \;\; F = ab\cos(v), \;\; G = a^2,$$

which gives the metric,

$$ds^2 = (b^2 + a^2\cos^2(v))du^2 + (ab\cos(v))du^2dv^2 + (a^2)dv^2.$$

The Christoffel symbols are

$$\Gamma^1_{11} = -a^3b\cos^2(v)\sin(v), \;\; \Gamma^1_{12} = a^2\cos(v)\sin(v)(a^2\cos^2(v) + b^2),$$

$$\Gamma^1_{22} = -a^4\cos(v)\sin(v), \;\; \Gamma^2_{11} = a^3b\cos^2(v)\sin(v),$$

$$\Gamma^2_{12} = -a^3b\sin(v), \;\; \Gamma^2_{22} = a^2b^2\cos(v)\sin(v),$$

from which we obtain the following geodesic equations,

$$u'' - \frac{ab\cos^2(v)\sin(v)}{a^2\cos^2(v) + b^2 - b^2\cos^2(v)}u'^2 - \frac{2a^2\cos(v)\sin(v)}{a^2\cos^2(v) + b^2 - b^2\cos^2(v)}u'v'$$

$$-\frac{ab\sin(v)}{a^2\cos^2(v) + b^2 - b^2\cos^2(v)}v'^2 = 0$$

$$v'' + \frac{(a^2\cos^2(v) + b^2)\sin(v)\cos(v)}{a^2\cos^2(v) + b^2 - b^2\cos^2(v)}u'^2 - \frac{2ab\cos^2(v)\sin(v)}{a^2\cos^2(v) + b^2 - b^2\cos^2(v)}u'v'$$

$$+\frac{b^2\cos(v)\sin(v)}{a^2\cos^2(v) + b^2 - b^2\cos^2(v)}v'^2 = 0$$

We modify the above differential equations in order to solve them by numerical methods as a fourth order system of first order equations,

$$u' = p$$

$$v' = q$$

$$\begin{aligned} p' = &+\frac{ab\cos^2(v)\sin(v)}{a^2\cos^2(v)+b^2-b^2\cos^2(v)}p^2+\frac{2a^2\cos(v)\sin(v)}{a^2\cos^2(v)+b^2-b^2\cos^2(v)}pq \\ &+\frac{ab\sin(v)}{a^2\cos^2(v)+b^2-b^2\cos^2(v)}q^2 \\ q' = &-\frac{(a^2\cos^2(v)+b^2)\sin(v)\cos(v)}{a^2\cos^2(v)+b^2-b^2\cos^2(v)}p^2+\frac{2ab\cos^2(v)\sin(v)}{a^2\cos^2(v)+b^2-b^2\cos^2(v)}pq \\ &-\frac{b^2\cos(v)\sin(v)}{a^2\cos^2(v)+b^2-b^2\cos^2(v)}q^2. \end{aligned}$$

The above calculations apply to the corkscrew surface obtained from a sphere. As we can see, the equations are quite complex. For our investigation of the corkscrew surface, we will slightly modify the parametrization and so the differential equations of geodesics will be even more complex and too long to be included here. In Maple, we will calculate and illustrate geodesics using the following parametrization

$$\mathbf{x}(u,v) = (2\cos(u)\cos(v), 4\sin(u)\cos(v), \sin(v)+0.5u),$$

$u \in [0, 2\pi]$ and $v \in [-\pi, \pi]$. This corkscrew surface modification twists an ellipsoid rather than sphere to create a surface similar locally in nature to the cortical surface.

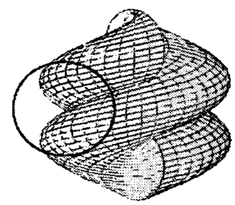

Figure 19: The Corkscrew Surface. $(a = 2, b = 4, c = 1, d = 0.5)$

The twists on the modified corkscrew surface resemble certain regions of the human cerebral cortex. We believe that this this particular surface is suitable to investigate in order to simulate parts of the cortical surface.

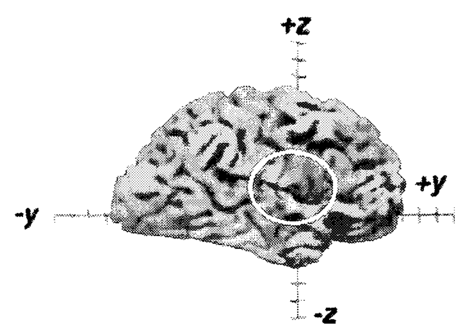

Figure 20: A 'twisted' region of the human cerebral cortex. [12]

Using RK4 method, the figures below relate to geodesics on the modified corkscrew surface generated at the initial point $(u_0, v_0) = (\pi, \pi)$ with the initial direction angles $\theta = \frac{2\pi}{50}i$, $i = 1, ..., 50$.

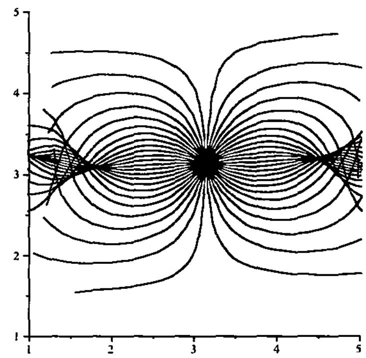

Figure 21: (The Corkscrew Surface) Geodesic curves in the uv-plane using RK4 method. ($h = 0.1, n = 30$ steps)

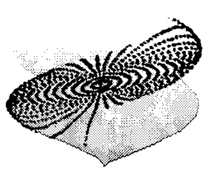

Figure 22: (The Corkscrew Surface) Geodesics on the surface using RK4 method. ($h = 0.1, n = 30$ steps)

We can illustrate geodesic circles by shooting geodesics in many directions from the same initial point. Also, a better representation of the self-intersecting geodesics will be shown in the following figures.

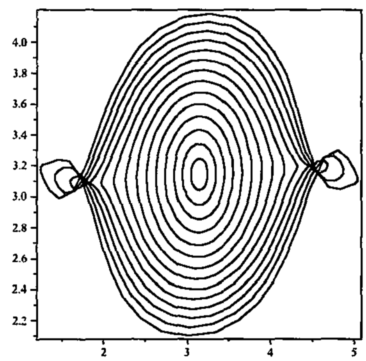

Figure 23: (The Corkscrew Surface) Geodesic circles in the uv-plane using RK4 method. $(h = 0.1, n = 15$ steps)

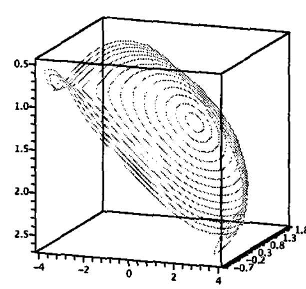

Figure 24: (The Corkscrew Surface) Geodesic circles on the surface using RK4 method. $(h = 0.1, n = 15$ steps)

Since we are interested analyzing certain regions of the corkscrew surface that resemble the human cerebral cortex, we need to use different initial points and possibly initial direction angles. Below, we have illustrated the geodesics starting at the initial point $(\frac{\pi}{9}, \frac{\pi}{9})$ with the initial direction angles $\theta = \frac{2\pi}{50}i$, $i = 1, ..., 50$, using RK4 method.

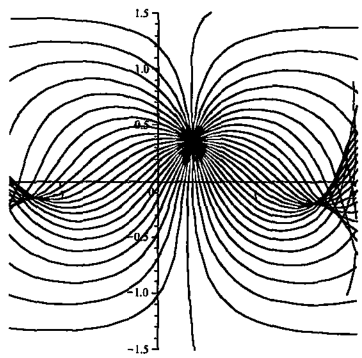

Figure 25: (The Corkscrew Surface) Geodesic curves in the uv-plane using RK4 method. $(h = 0.1, n = 30$ steps)

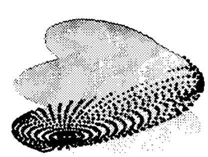

Figure 26: (The Corkscrew Surface) Geodesics on the surface using RK4 method. $(h = 0.1, n = 30$ steps)

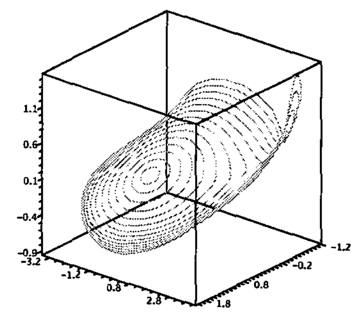

Figure 27: (The Corkscrew Surface) Geodesic circles on the surface using RK4 method. ($h = 0.1, n = 15$ steps)

Chapter 5

Calculating Geodesics on Surfaces Resembling the Human Cerebral Cortex

In this chapter, we will describe the human cerebral cortex and the importance of the studying it. The most common method used to reconstruct the cortical surface will be briefly discussed; then we will present an alternative method which calculates geodesics on the surface itself. Our agenda is to analyze two surfaces that resemble globally a larger part of the cortex which might aid in finding an approximate parametrization of the cortical surface.

5.1 Defining the Human Cerebral Cortex

In human beings, the brain governs the central nervous system which is responsible for human behaviour. It is comprised of two parts: gray matter and white matter. Gray matter, also referred to as the human cerebral cortex, is the outer layer consisting of nerve cells and fibers. The cortical surface is often visually represented as the ellipsoid surface (see Figure 28; the picture is take from *en.wikipedia.org/wiki/Cerebral_cortex*). The surface of the brain is an extremely complex structure to study, but very important, as it plays a crucial role in controlling sensations, thought and action.

Figure 28: The Human Cerebral Cortex.

White matter is the inner part, also comprised of nerve cells and fibers. It connects different regions of the cortex to make the brain systematically function as one unit. [10]

The human cerebral cortex is the most intensely studied part of the brain and can be visually reconstructed for analysis. We will suggest certain methods for this analysis. Literature shows that studying the surface of the brain is important, since it is believed that there exists a relation between the function and structure of the human cerebral cortex.

5.2 Geodesics and the Human Cerebral Cortex

According to Van Essen et al. [11] and Schwartz et al. [9], the most common type of surface reconstruction of the human cerebral cortex is represented by a two dimensional flat map. To construct a flat map, a region of the cerebral cortex is selected and marked with geographical landmarks or points. Since the cortex is bumpy with many dents and grooves, the region is smoothened out so that it can be represented in two dimensions. Using the surface-based warping algorithm, the minimal distance to each point is calculated. The points are mapped onto a two dimensional flat map. From the flat map, distances at each point can be computed using the two dimensional coordinate system.

Flat maps are deemed to be the most compact way of reconstructing the cortical surface, since a selected region can be visualized in a single view. However, in the process of converting the three dimensional cortex onto a two dimensional flat map, distortions to the shape of the cortical surface are inevitable, since there is no isometry between the cortex and the corresponding flat map. Also, the smoothening and the edges of the selected region cause distortions in the two dimensional representation. The distortions made by using the surface-base warping algorithm are not major setbacks in the method, but refinements of the algorithm are needed due to inaccurate representations of flat maps in higher resolutions. The paper [11] by Van Essen et al. does claim that measurements of distance are best made by calculating geodesics along three dimensional surfaces and that ellipsoidal maps would have less inaccurate representations than a flat map.

Instead of using the surface-based warping algorithm to generate a representation of the human cerebral cortex, we can focus on the entire cortical surface and numerically calculate geodesics directly on it without changing the shape of the surface. The challenge to this method is producing an approximate parametrization of the cortical surface due to the structural complexity. What we can do is to investigate the surfaces that resemble the cortex. By analyzing various parametrizations of surfaces, we could possibly find an approximate cortical surface parametrization.

We will investigate two particular surfaces that resemble the cortical surface:

the ellipsoid surface and the bumpy ellipsoid surface. As mentioned before, the cortical surface can be represented (approximated) as the ellipsoid surface. Also, as suggested in the paper by Van Essen et al., the ellipsoid surface can be used to create a topological representation of the cortex. We will deform the ellipsoid surface to produce the bumpy ellipsoidal surface which better resembles the cortical surface. We will discuss these surfaces and numerically calculate geodesics on them.

The Ellipsoid Surface

The ellipsoid is a regular surface parameterized by

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

where the coefficients a, b, c are positive numbers, $u \in [0, 2\pi)$ and $v \in [0, \pi]$. The numbers $2a$, $2b$, $2c$ are the diameters of the ellipsoid surface.

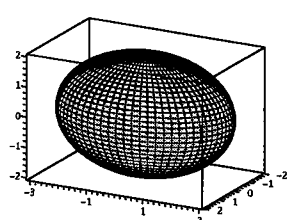

Figure 29: The Ellipsoid Surface. ($a = c = 2$, $b = 3$)

We use various initial points and initial direction angles to draw geodesics. Below, we have illustrated the geodesics at the initial point $(\frac{\pi}{3}, \frac{\pi}{3})$ and initial direction angles $\theta = \frac{2\pi}{12}i$, $i = 1, ..., 12$, using RK4 method.

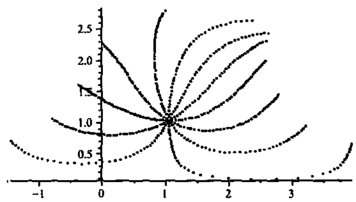

Figure 30: (The Ellipsoid Surface) Geodesics in the uv-plane, showing actual points calculated by using RK4 method. ($h = 0.05, n = 40$ steps)

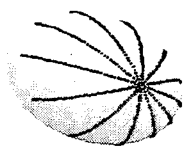

Figure 31: (The Ellipsoid Surface) Geodesics on the surface using RK4 method. ($h = 0.05, n = 40$ steps)

As described earlier, we construct geodesic circles. Using these circles, we can estimate distances on the ellipsoid surface.

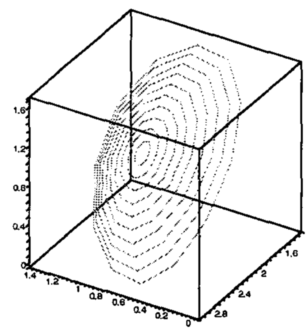

Figure 32: (The Ellipsoid Surface) Geodesic circles on the surface using RK4 method. ($h = 0.05, n = 11$ steps)

There were computation obstacles in generating geodesics on the ellipsoid surface. Only a limited number of steps and geodesics could be calculated at once. One reason being is that there was a break down in memory allocation in Maple.

The Bumpy Ellipsoid Surface

The bumpy ellipsoid surface is a regular surface parameterized by

$$\mathbf{x}(u,v) = [f(a+b\cos(mu)\sin(nv))\sin(v)\cos(u), g(a+b\cos(mu)\sin(nv))\sin(u)\sin(v),$$

$$h(a+b\cos(mu)\sin(nv))\cos(v)]$$

where $u, v \in [0, 2\pi)$, the coefficients a, b, f, g, h are positive numbers, and the coefficients m, n are positive integers. The coefficients a, b govern the depth of the bumps and grooves and the coefficients m, n change the "bumpness" and rigidity of the ellipsoid. It is interesting to note that in research, similar surfaces are used to model tumors. [4]

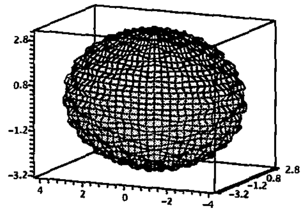

Figure 33: The Bumpy Ellipsoid Surface. (a = 1, b = 0.05, m = 10, n = 20, f = 3, g = 4, h = 3)

Thus, by deforming the ellipsoid surface, we achieve a closer representation to the human cerebral cortex. We will plot the geodesics on the surface at the initial point $(\frac{\pi}{9}, \frac{\pi}{9})$ and initial direction angles $\theta = \frac{2\pi}{10}i$, $i = 1, ..., 10$, using RK4 method.

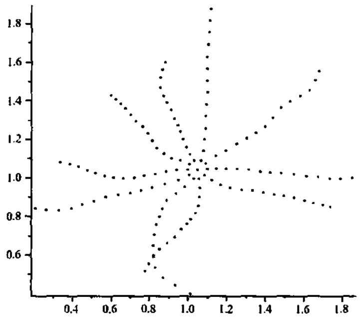

Figure 34: (The Bumpy Ellipsoid Surface) Geodesics in the uv-plane, showing actual points calculated by using RK4 method. ($h = 0.05, n = 15$ steps)

Again, due to the constraints on hardware and software, we can only show a few geodesics. It is interesting to note how the geodesics in Figure 31 cross one another. On the surface, the geodesics look scattered and the direction of the geodesics looks rather unpredictable.

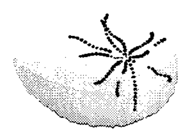

Figure 35: (The Bumpy Ellipsoid Surface) Geodesics on the surface using RK4 method. ($h = 0.05, n = 15$ steps)

Constructing geodesic circles allows us to estimate distances on the surface.

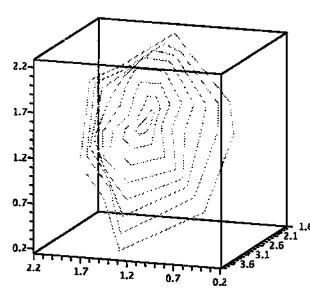

Figure 36: (The Bumpy Ellipsoid Surface) Geodesic circles on the surface using RK4 method. ($h = 0.05, n = 10$ steps)

Computational difficulties arose again when we were calculating geodesics on the bumpy ellipsoid surface in Maple. Only a limited number of steps and geodesics could be generated. Producing only 10 geodesics on the surface took a long time to compute. When we tried to draw more than 10 geodesics, Maple would loose its memory allocation and freeze.

The best way to study the surface of the human brain, in particular distances, may be to calculate geodesics directly on it. If an approximate parametrization is developed, then through geodesic analysis, we can yield fairly good approximations of distance. The accuracy of approximation is discussed in the paper [10] by Teo et al. If an approximate parametrization cannot be found, then surfaces that resemble the cortical surface locally can be used to conduct similar work as shown in Chapter 4.

Chapter 6

Conclusion

As mentioned in the previous chapter, a parametrization of a large part of the cortical surface would be ideal to perform geodesic calculations on; however, due to the complexity of the structure, it may not be possible. In finding a global parametrization of the cortex, one obstacle is computer power - the hardware, in particular, but also software. In our research on relatively "simple" surfaces, we experienced some obstacles in computing geodesics numerically. Definitely, the computer hardware should be more powerful and the computer software more advanced (or we could have tired application like Mathematica) in order to handle complex calculations.

In practice, the bumpy ellipsoidal surface which is used to analyze tumors was most likely developed by investigating different modifications to the usual sphere parametrizations. Similarly, since the human cerebral cortex resembles the ellipsoid surface, modifications to the ellipsoid surface seem like a fairly logical way of analyzing it. However, if a global cortical parametrization cannot be found, then local regions of the cortex can be investigated, as seen in Chapter 4, to aid in understanding the structure of the entire human cerebral cortex.

One avenue of future research is implementing the phase flow method to numerically compute a large number of geodesics on the human cerebral cortex. The phase flow method, which has been developed recently, is claimed to be efficient and accurate. [13] In the case of geodesics, instead of solving differential equations of geodesics at one initial condition as in RK4 method, the system is calculated at many initial conditions at once. This method constructs a geodesic flow map which is a trace of all the geodesics with various initial conditions.

In various disciplines, many physical based problems are better understood using geodesics. For instance, the study of a large number of geodesics is important in computational physics where the phase flow method is applied. In analyzing high-frequency wave propagation, trajectories which are also referred to as geodesics are computed on a surface from various initial points, producing many trajectories. The trajectories create a "wavefield" from which the dynamics of wave propagation is analyzed. [13] Thus, calculating geodesics on surfaces can serve as a useful tool in learning the nature of various physical problems.

Appendix A: General Calculations of a Surface

```
> restart:
  with(plots):
```

```
> dp:= proc(X,Y) X[1]*Y[1]+X[2]*Y[2]+X[3]*Y[3];
  end:
```

Calculation of the dot product.

```
> nrm:= proc(X)
  sqrt(dp(X,X));
  end:
```

Calculation of the norm.

```
> W:= proc(X,Y)
  [X[2]*Y[3] - X[3]*Y[2], X[3]*Y[1] - X[1]*Y[3], X[1]*Y[2] - X[2]*
  Y[1]];
  end:
```

Calculation of the cross product.

```
> X:=[ insert surface parametrisation here ]:
```

```
> Xu:= simplify([diff(X[1],u),diff(X[2],u),diff(X[3],u)]);
  Xv:= [diff(X[1],v),diff(X[2],v),diff(X[3],v)];
  Xuu:= simplify([diff(Xu[1],u),diff(Xu[2],u),diff(Xu[3],u)]);
  Xuv:= [diff(Xu[1],v),diff(Xu[2],v),diff(Xu[3],v)];
  Xvv:= ([diff(Xv[1],v),diff(Xv[2],v),diff(Xv[3],v)]);
```

The first and second derivative of the surface X with respect to the variables u and v..

```
> E:= dp(Xu,Xu);
  F:= dp(Xu,Xv);
  G:= dp(Xv,Xv);
  E:= simplify(dp(Xu,Xu));
  F:= simplify(dp(Xu,Xv));
  G:= simplify(dp(Xv,Xv));
```

The coefficients of the first fundamental form.

```
> Eu:= simplify(diff(E,u));
  Ev:= simplify(diff(E,v));
  Gu:= simplify(diff(G,u));
  Gv:= simplify(diff(G,v));
  Fu:= simplify(diff(F,u));
  Fv:= simplify(diff(F,v));
```

The first derivative of the coefficients of the first fundamental form with respect to the variables u and v.

```
> GAMMA(1,1,1):= simplify((G*Eu - 2*F*Fu + F*Ev)/2(E*G-F^2));
  GAMMA(1,1,2):= simplify((G*Ev - F*Gu)/2(E*G-F^2));
  GAMMA(1,2,2):= simplify((2*G*Fv - G*Gu - F*Gv)/2(E*G-F^2));
  GAMMA(2,1,1):= simplify((2*E*Fu - E*Ev - F*Eu)/2(E*G-F^2));
  GAMMA(2,1,2):= simplify((E*Gu - F*Ev)/2(E*G-F^2));
  GAMMA(2,2,2):= simplify((E*Gv - 2*F*Fv + F*Gu)/2(E*G-F^2));
```

The Christoffel symbols.

Appendix B: Differential Equations of Geodesics for a Surface

```
> restart:
  with(plots):
```

```
> dp:= proc(X,Y)
  X[1]*Y[1]+X[2]*Y[2]+X[3]*Y[3];
  end:
```

Calculation of the dot product.

```
> Jacf:= proc(X)
  local Xu, Xv;
  Xu:= [diff(X[1],u), diff(X[2],u), diff(X[3],u)];
  Xv:= [diff(X[1],v), diff(X[2],v), diff(X[3],v)];
  simplify([Xu,Xv]);
  end:
```

The first derivative of the surface X with respect to the variables u and v.

```
> EFG:= proc(X)
  local E,F,G,Y;
  Y:= Jacf(X);
  E:= dp(Y[1],Y[1]);
  F:= dp(Y[1],Y[2]);
  G:= dp(Y[2],Y[2]);
  simplify([E,F,G]);
  end:
```

The coefficients of the first fundamental form.

```
> DEFG:= proc(X)
  local Eu,Ev,Fu,Fv,Gu,Gv,S;
  S:= EFG(X);
  Eu:= simplify(diff(S[1],u));
  Ev:= simplify(diff(S[1],v));
  Fu:= simplify(diff(S[2],u));
  Fv:= simplify(diff(S[2],v));
  Gu:= simplify(diff(S[3],u));
  Gv:= simplify(diff(S[3],v));
  simplify([Eu,Ev,Fu,Fv,Gu,Gv]);
  end:
```

The first derivative of the coefficients of the first fundamental form with respect to the variables u and v.

```
> geoeq:= proc(X)
  local eq1,eq2,GAMMA111,GAMMA112,GAMMA122,GAMMA211,GAMMA212,
  GAMMA222,S,T;
  S:= EFG(X);
  T:= DEFG(X);
  GAMMA111:= simplify((S[3]*T[1]-2*S[2]*T[3]+S[2]*T[2])/(2*(S[1]*S
  [3]-(S[2]^2))));
  GAMMA112:= simplify((S[3]*T[2]-S[2]*T[5])/(2*(S[1]*S[3]-S[2]^2))
  );
  GAMMA122:= simplify((2*S[3]*T[4]-S[3]*T[5]-S[2]*T[6])/(2*(S[1]*S
  [3]-S[2]^2)));
  GAMMA211:= simplify((2*S[1]*T[3]-S[1]*T[2]-S[2]*T[1])/(2*(S[1]*S
  [3]-S[2]^2)));
  GAMMA212:= simplify((S[1]*T[5]-S[2]*T[2])/(2*(S[1]*S[3]-S[2]^2))
  );
  GAMMA222:= simplify((S[1]*T[6]-2*S[2]*T[4]+S[2]*T[5])/(2*(S[1]*S
  [3]-S[2]^2)));
  eq1:= diff(u(t),t$2) + subs({u=u(t),v=v(t)}, GAMMA111)*diff(u(t)
  ,t)^2 + 2*subs({u=u(t),v=v(t)}, GAMMA112)*diff(u(t),t)*diff(v(t)
  ,t) + subs({u=u(t),v=v(t)}, GAMMA122)*diff(v(t),t)^2 = 0;
  eq2:= diff(v(t),t$2) + subs({u=u(t),v=v(t)}, GAMMA211)*diff(u(t)
  ,t)^2 + 2*subs({u=u(t),v=v(t)}, GAMMA212)*diff(u(t),t)*diff(v(t)
  ,t) + subs({u=u(t), v=v(t)}, GAMMA222)*diff(v(t),t)^2 = 0;
  eq1, eq2;
  end:
```

Calculation of the differential equations of geodesics of the given surface.

```
> X:= [ , , ];
```

Insert the surface parametrization with three components.

```
> geoeq(X);
```

Displays the calculated differential equations of geodesics.

Appendix C: Euler's Method

```
restart:
with(plots):
with(plottools):
```

```
> a:= (u,v) -> :
  b:= (u,v) -> :
  c:= (u,v) -> :
  X:= [a(u,v),b(u,v),c(u,v)]:
```

Insert surface parametrization with three components.

```
> dp:= proc(X,Y)
  X[1]*Y[1] + X[2]*Y[2] + X[3]*Y[3];
  end:

  Xu:= simplify([diff(X[1],u),diff(X[2],u),diff(X[3],u)]):
  Xv:= simplify([diff(X[1],v),diff(X[2],v),diff(X[3],v)]):
  Xuu:= simplify([diff(Xu[1],u),diff(Xu[2],u),diff(Xu[3],u)]):
  Xuv:= simplify([diff(Xu[1],v),diff(Xu[2],v),diff(Xu[3],v)]):
  Xvv:= simplify([diff(Xv[1],v),diff(Xv[2],v),diff(Xv[3],v)]):

  E:= simplify(dp(Xu,Xu)):
  F:= simplify(dp(Xu,Xv)):
  G:= simplify(dp(Xv,Xv)):

  Eu:= simplify(diff(E,u)):
  Ev:= simplify(diff(E,v)):
  Fu:= simplify(diff(F,u)):
  Fv:= simplify(diff(F,v)):
  Gu:= simplify(diff(G,u)):
  Gv:= simplify(diff(G,v)):

  GAMMA111:= subs({u=w[1], v=w[2]}, simplify((G*Eu-2*F*Fu+F*Ev)/
  (2*(E*G-(F^2))))):
  GAMMA112:= subs({u=w[1], v=w[2]}, simplify((G*Ev-F*Gu)/(2*(E*G-
  F^2)))):
  GAMMA122:= subs({u=w[1], v=w[2]}, simplify((2*G*Fv-G*Gu-F*Gv)/
  (2*(E*G-F^2)))):
  GAMMA211:= subs({u=w[1], v=w[2]}, simplify((2*E*Fu-E*Ev-F*Eu)/
  (2*(E*G-F^2)))):
  GAMMA212:= subs({u=w[1], v=w[2]}, simplify((E*Gu-F*Ev)/(2*(E*G-
  F^2)))):
  GAMMA222:= subs({u=w[1], v=w[2]}, simplify((E*Gv-2*F*Fv+F*Gu)/
  (2*(E*G-F^2)))):
```

Calculations of dot product, the first and second derivative of the surface X with respect to the variables u and v, The coefficients of the first fundamental form, the first derivative of the coefficients of the first fundamental form with respect to the variables u and v and Christoffel symbols.

```
> iterate_geodesic:= proc(X)
local u,v,p,q;
u:= evalf(w[1] + h*w[3]);
v:= w[2] + h*w[4];
p:= w[3] + h*(-1*GAMMA111*w[3]^2 - 2*GAMMA112*w[3]*w[4] -
GAMMA122*w[4]^2);
q:= w[4] + h*(-1*GAMMA211*w[3]^2 - 2*GAMMA212*w[3]*w[4] -
GAMMA222*w[4]^2);
[u,v,p,q];
end:
```

Euler's method iteration formula.

```
h:= 0.01:
steps:= 100:
initialpoint:= [0,0]:
num_geo:= 100:
```

Setting up initial conditions to calculate geodesics.

```
> iterate_geodesic(X);
```

Displays Euler's method iteration formula.

```
> for j from 1 to num_geo do
initialangle[j]:= (2*Pi/num_geo)*j;
initialdirection[j]:= [evalf(cos(initialangle[j])),evalf(sin
(initialangle[j]))];
w:= [initialpoint[1],initialpoint[2],initialdirection[j][1],
initialdirection[j][2]];
u[0]:= w[1];
v[0]:= w[2];
p[0]:= w[3];
q[0]:= w[4];
for i from 1 by 1 to steps do
ww:= iterate_geodesic(w);
u[i]:= w[1];
v[i]:= w[2];
p[i]:= w[3];
q[i]:= w[4];
w:= ww
end do:
geodesic[j]:= plot([seq([u[i],v[i]],i=0..steps)],color=blue,
style=point,symbol=point):
points:= {seq([a(u[i],v[i]), b(u[i],v[i]), c(u[i],v[i])],i=0..
steps)}:
geo[j]:= pointplot3d(points,color=red):
end do:
```

```
> for i from 1 to num_geo do
initialangle[i]:= (2*Pi/num_geo)*i:
initialdirection[i]:= [evalf(cos(initialangle[i])),evalf(sin
(initialangle[i]))];
w:= [initialpoint[1],initialpoint[2],initialdirection[i][1],
initialdirection[i][2]];
u[0]:= w[1];
v[0]:= w[2];
p[0]:= w[3];
q[0]:= w[4];
g[i,0]:= [u[0],v[0]];
for j from 1 by 1 to steps do
ww:= iterate_geodesic(w);
u[j]:= w[1];
v[j]:= w[2];
p[j]:= w[3];
q[j]:= w[4];
w:= ww;
g[i,j]:= [u[j],v[j]];
end do:
geod[i]:= plot([seq([g[i,j][1],g[i,j][2]],j=0..steps)],color=
blue,style=point,symbol=point):
points:= {seq([a(u[j],v[j]), b(u[j],v[j]), c(u[j],v[j])],j=0..
steps)}:
geo[i]:= pointplot3d(points,color=red);
end do:
```

The first and second loop calculates geodesics in the uv-plane, showing actual points calculated.

```
> for i from 1 by 1 to num_geo do
for j from 0 by 1 to steps-1 do
linesegment[i][j]:= plots[display](line([g[i,j][1],g[i,j][2]],
[g[i,j+1][1],g[i,j+1][2]])):
end do:
end do:

for i from 0 by 1 to num_geo do
for j from 1 by 1 to steps-1 do
circlesegment[i][j]:= plots[display](line([g[i,j][1],g[i,j][2]],
[g[i+1,j][1],g[i+1,j][2]])):
circlesegment[0][j]:= plots[display](line([g[1,j][1],g[1,j][2]],
[g[num_geo,j][1],g[num_geo,j][2]])):
end do:
end do:

for i from 0 by 1 to num_geo do
for j from 1 by 1 to steps-1 do
circlesegment3d[i][j]:= plots[display](line([a(g[i,j][1],g[i,j]
[2]), b(g[i,j][1],g[i,j][2]), c(g[i,j][1],g[i,j][2])], [a(g[i+1,
j][1],g[i+1,j][2]), b(g[i+1,j][1],g[i+1,j][2]), c(g[i+1,j][1],g
[i+1,j][2])]])):
circlesegment3d[0][j]:= plots[display](line([a(g[1,j][1],g[1,j]
[2]), b(g[1,j][1],g[1,j][2]), c(g[1,j][1],g[1,j][2])], [a(g
[num_geo,j][1],g[num_geo,j][2]), b(g[num_geo,j][1],g[num_geo,j]
[2]), c(g[num_geo,j][1],g[num_geo,j][2])])):
end do:
end do:
```

The first loop calculates geodesics in the uv-plane as curves. The second and third loop calculate geodesics as circles in the uv-plane and on the surface respectively.

```
> geo[num_geo+1]:= plot3d([a(u,v), b(u,v), c(u,v)], u=0..Pi, v=-2*
  Pi..2*Pi):
  display(geo[num_geo+1]);

  display(seq([geodesic[j]],j=1..num_geo),scaling=constrained,
  view=[-5..5, -5..5]);

  display(seq([geo[j]],j=1..num_geo+1),scaling=constrained,view=
  [-5..5, -5..5, -5..5]);

  display(seq(seq([linesegment[i][j]],j=0..steps-1),i=1..num_geo))
  ;

  display(seq(seq([circlesegment[i][j]],i=0..num_geo-1),j=2..30));

  display(seq(seq([circlesegment3d[i][j]],i=0..num_geo-1),j=2..30)
  );
```

The first command displays the surface.

The second command displays geodesics as points in the uv-plane.

The third command displays geodesics as points on the surface.

The fourth command displays geodesics as curves in the uv-plane.

The fifth command displays geodesics as circles in the uv-plane.

The sixth command displays the geodesic circles on the surface.

Appendix D: Runge-Kutta Method of Fourth Order

```
> restart:
  with(plots):
  with(plottools):

> a:= (u,v) -> u:
  b:= (u,v) -> v:
  c:= (u,v) -> u^3-3*u*v^2:
  X:= [a(u,v),b(u,v),c(u,v)];
```

Insert surface parametrization with three components.

```
> dp:= proc(X,Y)
  X[1]*Y[1] + X[2]*Y[2] + X[3]*Y[3];
  end:

  Xu:= simplify([diff(X[1],u),diff(X[2],u),diff(X[3],u)]):
  Xv:= simplify([diff(X[1],v),diff(X[2],v),diff(X[3],v)]):
  Xuu:= simplify([diff(Xu[1],u),diff(Xu[2],u),diff(Xu[3],u)]):
  Xuv:= simplify([diff(Xu[1],v),diff(Xu[2],v),diff(Xu[3],v)]):
  Xvv:= simplify([diff(Xv[1],v),diff(Xv[2],v),diff(Xv[3],v)]):

  E:= simplify(dp(Xu,Xu)):
  F:= simplify(dp(Xu,Xv)):
  G:= simplify(dp(Xv,Xv)):

  Eu:= simplify(diff(E,u)):
  Ev:= simplify(diff(E,v)):
  Fu:= simplify(diff(F,u)):
  Fv:= simplify(diff(F,v)):
  Gu:= simplify(diff(G,u)):
  Gv:= simplify(diff(G,v)):

  GAM111:= subs({u=uu, v=vv}, simplify((G*Eu-2*F*Fu+F*Ev)/(2*(E*G-
  (F^2))))):
  GAM112:= subs({u=uu, v=vv}, simplify((G*Ev-F*Gu)/(2*(E*G-F^2))))
  :
  GAM122:= subs({u=uu, v=vv}, simplify((2*G*Fv-G*Gu-F*Gv)/(2*(E*G-
  F^2)))):
  GAM211:= subs({u=uu, v=vv}, simplify((2*E*Fu-E*Ev-F*Eu)/(2*(E*G-
  F^2)))):
  GAM212:= subs({u=uu, v=vv}, simplify((E*Gu-F*Ev)/(2*(E*G-F^2))))
  :
  GAM222:= subs({u=uu, v=vv}, simplify((E*Gv-2*F*Fv+F*Gu)/(2*(E*G-
  F^2)))):

  GAMMA111:=unapply(GAM111,uu,vv):
  GAMMA112:=unapply(GAM112,uu,vv):
  GAMMA122:=unapply(GAM122,uu,vv):
  GAMMA211:=unapply(GAM211,uu,vv):
  GAMMA212:=unapply(GAM212,uu,vv):
  GAMMA222:=unapply(GAM222,uu,vv):
```

Calculations of dot product, the first and second derivative of the surface X with respect to the variables u and v, The coefficients of the first fundamental form, the first derivative of the coefficients of the first fundamental form with respect to the variables u and v and Christoffel symbols.

```
> f1:=(t,uu,vv,pp,qq)->pp:
  f2:=(t,uu,vv,pp,qq)->qq:
  f3:=(t,uu,vv,pp,qq)->-1*GAMMA111(uu,vv)*pp^2 - 2*GAMMA112(uu,vv)
  *pp*qq - GAMMA122(uu,vv)*qq^2:
  f4:=(t,uu,vv,pp,qq)->-1*GAMMA211(uu,vv)*pp^2 - 2*GAMMA212(uu,vv)
  *pp*qq - GAMMA222(uu,vv)*qq^2:


  geodesic_by_rk4:= proc(w)
  local u, v, p, q, k11,k12,k13,k14,k21,k22,k23,k24,k31,k32,k33,
  k34,k41,k42,k43,k44,tt;

  k11:=h*f1(tt,w[1],w[2],w[3],w[4]);
  k12:=h*f2(tt,w[1],w[2],w[3],w[4]);
  k13:=h*f3(tt,w[1],w[2],w[3],w[4]);
  k14:=h*f4(tt,w[1],w[2],w[3],w[4]);

  k21:=h*f1(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
  k22:=h*f2(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
  k23:=h*f3(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
  k24:=h*f4(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);

  k31:=h*f1(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
  k32:=h*f2(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
  k33:=h*f3(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
  k34:=h*f4(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);

  k41:=h*f1(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
  k42:=h*f2(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
  k43:=h*f3(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
  k44:=h*f4(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);

  u:= evalf(w[1] + 1/6*(k11 + 2*k21 + 2*k31 + k41));
  v:= evalf(w[2] + 1/6*(k12 + 2*k22 + 2*k32 + k42));
  p:= evalf(w[3] + 1/6*(k13 + 2*k23 + 2*k33 + k43));
  q:= evalf(w[4] + 1/6*(k14 + 2*k24 + 2*k34 + k44));
  tt:=w[5]+h;
  [u,v,p,q,tt];
  end:
```

Runge-Kutta method of fourth order iteration formula.

```
> h:= 0.1:
  steps:= 20:
  num_geo:= 10:
  initialpoint:= [0,0]:
```

Setting up initial conditions to calculate geodesics.

```
> for j from 1 to num_geo do
  initialangle[j]:= (2*Pi/num_geo)*j:
  initialdirection[j]:= [evalf(cos(initialangle[j])),evalf(sin
  (initialangle[j]))];
  w:= [initialpoint[1],initialpoint[2],initialdirection[j][1],
  initialdirection[j][2],0]:
  u[0]:= w[1];
  v[0]:= w[2];
  p[0]:= w[3];
  q[0]:= w[4];
  g[0,j]:= [u[0],v[0]];
  for i from 1 by 1 to steps do
  ww:= geodesic_by_rk4(w);
  u[i]:= w[1];
  v[i]:= w[2];
  p[i]:= w[3];
  q[i]:= w[4];
  w:= ww;
  g[i,j]:= [u[i],v[i]];
  end do:
  geodesic[j]:= plot([seq([u[i],v[i]],i=0..steps)],color=black,
  style=point,symbol=point):
  points:= {seq([a(u[i],v[i]), b(u[i],v[i]), c(u[i],v[i])],i=0..
  steps)}:
  geo[j]:= pointplot3d(points,style=point,symbol=circle,
  symbolsize=4,color=black);
  end do:
```

The first and second loop calculates geodesics in the uv-plane, showing actual points calculated.

```
> geo[num_geo+1]:=plot3d([a(u,v),b(u,v),c(u,v)],u=-2..2,v=-2..2):
  display(seq([geo[j]],j=1..num_geo+1),scaling=unconstrained,view=
  [-2..2,-2..2,-4..3]);
  display(seq([geodesic[j]],j=1..num_geo),scaling=unconstrained,
  view=[-1..3,-1..4] );
```

Displays the geodesics as points on the surface and uv-plane respectively.

```
> for j from 1 to num_geo do
  for i from 0 by 1 to steps-1 do
  linesegment[i][j]:= plots[display](line([g[i,j][1],g[i,j][2]],
  [g[i+1,j][1],g[i+1,j][2]])):
  end do:
  end do:
  display(seq(seq([linesegment[i][j]],i=0..steps-1),j=1..num_geo),
  view=[-2..2,-2..2]);
```

The first loop calculates and displays the geodesics in the uv-plane as curves.

```
> for i from 1 by 1 to steps-1 do
  for j from 1 by 1 to num_geo do
  circlesegment[i][j]:= plots[display](line([g[i,j][1],g[i,j][2]],
  [g[i,j+1][1],g[i,j+1][2]])):
  circlesegment[i][0]:= plots[display](line([g[i,1][1],g[i,1][2]],
  [g[i,num_geo][1],g[i,num_geo][2]])):

  circlesegment3d[i][j]:= plots[display](line([a(g[i,j][1],g[i,j]
  [2]), b(g[i,j][1],g[i,j][2]), c(g[i,j][1],g[i,j][2])], [a(g[i,
  j+1][1],g[i,j+1][2]), b(g[i,j+1][1],g[i,j+1][2]), c(g[i,j+1][1],
  g[i,j+1][2])]),color=red):
  circlesegment3d[i][0]:= plots[display](line([a(g[i,1][1],g[i,1]
  [2]), b(g[i,1][1],g[i,1][2]), c(g[i,1][1],g[i,1][2])], [a(g[i,
  num_geo][1],g[i,num_geo][2]), b(g[i,num_geo][1],g[i,num_geo][2])
  , c(g[i,num_geo][1],g[i,num_geo][2])]),color=red):
  end do:
  end do:

  for i from 1 by 1 to steps-1 do
  circlesegment3d[i][num_geo]:=plot3d([a(u,v),b(u,v),c(u,v)],u=-5.
  .5,v=-5..5,style=patchnogrid):
  end do:

  display(seq(seq([circlesegment[i][j]],j=0..num_geo-1),i=1..15),
  axes=none);
  display(seq(seq([circlesegment3d[i][j]],j=0..num_geo-1),i=1..15)
  ,view=[-2..4,-2..4,-5..3]);
```

The first loop calculates the geodesics as circles in the uv-plane and then displays them.

The second loop calculates the geodesic circles on the surface and then displays them.

Appendix E: Distance Formula for Poincaré Half Plane

```
> restart:
  with(plots):
  with(plottools):
```

```
> distance:= proc(A,B) local x0, R, t1, t2, F;
  if A[1]=B[1] then RETURN(abs(log(B[2]/A[2]))) else
  x0:= solve((B[1]-x)^2+B[2]^2 = (A[1]-x)^2+A[2]^2,x);
  R:= sqrt((B[1]-x0)^2+B[2]^2); t1:= arccos ((B[1]-x0)/R);
  t2:= arccos((A[1]-x0)/R); F:=t -> ln(tan(t/2));
  RETURN (evalf (abs (F(t2)-F(t1)))) fi end:
```

Calculation of distance on the Poincare half plane surface.

```
> segment:= proc (A,B) local x0, R, t1, t2, X, Y;
  if A[1]=B[1] then
  plot ({[A,B], [[A[1],0], [A[1], 2*max(A[2], B[2])]]},
  scaling = constrained, thickness= [2,1], linestyle= [1,3])
  else x0:= solve ((B[1]-x)^2+B[2]^2 = (A[1]-x)^2+A[2]^2, x);
  R:= sqrt ((B[1]-x0)^2+B[2]^2); X:= x0+R*cos(t); Y:= R*sin(t);
  t1:= arccos((B[1]-x0)/R); t2:= arccos ((A[1]-x0)/R);
  plot({[X,Y,t=t1..t2], [X,Y,t=0..Pi/2]}, color=blue, scaling=
  constrained,
  thickness=[2,1], linestyle= [1,2]) fi end:
```

Command to plot line segments on the Poincare half plane surface.

```
> geodesic_by_euler:=proc(w)
  local u,v,p,q;
  u:= evalf(w[3]*h+w[1]);
  v:= w[4]*h+w[2];
  p:= 2*w[3]*w[4]*h/w[2]+w[3];
  q:= ((w[4])^2-(w[3])^2)*h/w[2]+w[4];
  [u,v,p,q];
  end:
```

Euler's method iteration formula for the Poincare half plane.

```
f1:=(t,uu,vv,pp,qq)->pp:
f2:=(t,uu,vv,pp,qq)->qq:
f3:=(t,uu,vv,pp,qq)->2*pp*qq/vv:
f4:=(t,uu,vv,pp,qq)->(qq^2-pp^2)/vv:

geodesic_by_rk4:= proc(w)
local u, v, p, q, k11,k12,k13,k14,k21,k22,k23,k24,k31,k32,k33,
k34,k41,k42,k43,k44,tt;
k11:=h*f1(tt,w[1],w[2],w[3],w[4]);
k12:=h*f2(tt,w[1],w[2],w[3],w[4]);
k13:=h*f3(tt,w[1],w[2],w[3],w[4]);
k14:=h*f4(tt,w[1],w[2],w[3],w[4]);
k21:=h*f1(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
k22:=h*f2(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
k23:=h*f3(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
k24:=h*f4(tt+h/2,w[1]+k11/2,w[2]+k12/2,w[3]+k13/2,w[4]+k14/2);
k31:=h*f1(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
k32:=h*f2(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
k33:=h*f3(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
k34:=h*f4(tt+h/2,w[1]+k21/2,w[2]+k22/2,w[3]+k23/2,w[4]+k24/2);
k41:=h*f1(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
k42:=h*f2(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
k43:=h*f3(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
k44:=h*f4(tt+h,w[1]+k31,w[2]+k32,w[3]+k33,w[4]+k34);
u:= evalf(w[1] + 1/6*(k11 + 2*k21 + 2*k31 + k41));
v:= evalf(w[2] + 1/6*(k12 + 2*k22 + 2*k32 + k42));
p:= evalf(w[3] + 1/6*(k13 + 2*k23 + 2*k33 + k43));
q:= evalf(w[4] + 1/6*(k14 + 2*k24 + 2*k34 + k44));
tt:=w[5]+h;
[u,v,p,q,tt];
end:
```

Runge-Kutta method of fourth order iteration formula for the Poincare half plane.

```
num_geo:=48;
h:= 0.09;
steps:= 20:
initialpoint:= [1,3]:given_point:=[1.4,3.4]:
```

Setting the initial conditions to calculate the geodesics.

```
> for i from 1 to num_geo do
initialangle[i]:= (2*Pi/num_geo)*i;
initialdirection[i]:= [evalf(cos(initialangle[i])),evalf(sin
(initialangle[i]))];
w:= [initialpoint[1],initialpoint[2],initialdirection[i][1],
initialdirection[i][2],0];
u[0]:= w[1];
v[0]:= w[2];
p[0]:= w[3];
q[0]:= w[4];
g[i,0]:= [u[0],v[0]];

for j from 1 by 1 to steps do
ww:= geodesic_by_rk4(w);
u[j]:= ww[1];
v[j]:= ww[2];
p[j]:= ww[3];
q[j]:= ww[4];
w:= ww;
g[i,j]:= [u[j],v[j]];
end do:
geod[i]:= plot([seq([g[i,j][1],g[i,j][2]],j=0..steps)],color=
blue,style=point,symbol=point):
end do:
```

The first loop calculates the geodesics starting at the initial point and the initial direction vector and iterates the geodesics by the step size as per the number of steps indicated. The second loop plots the iterated steps of the geodesic as points.

```
> for i from 1 by 1 to num_geo do
for j from 0 by 1 to steps-1 do
linesegment[i][j]:= plots[display](line([g[i,j][1],g[i,j][2]],
[g[i,j+1][1],g[i,j+1][2]])):
end do:
end do:

for i from 0 by 1 to num_geo do
for j from 1 by 1 to steps-1 do
circlesegment[i][j]:= plots[display](line([g[i,j][1],g[i,j][2]],
[g[i+1,j][1],g[i+1,j][2]])):
circlesegment[0][j]:= plots[display](line([g[1,j][1],g[1,j][2]],
[g[num_geo,j][1],g[num_geo,j][2]])):
end do:
end do:

for j from 1 by 1 to steps-1 do
circlesegment[num_geo][j]:= pointplot([given_point[1],
given_point[2]]):
end do:circlesegment[num_geo][1]:= pointplot([initialpoint[1],
initialpoint[2]]):
```

The first loop calculates the geodesics starting at the initial point and the initial direction vector and connects the points (calculated in the above procedure) to generate geodesic lines. The second loop plots geodesic circles. The third loop plots the a given point in order to calculate the distance between the initial point and given point.

```
> display(seq(seq([circlesegment[i][j]],i=0..num_geo),j=1..
  steps-1));
```

Displays the geodesic circles starting at the initial point and the given point.

```
> lower_estimate:=5*h/2.;
  upper_estimate:=6*h/2.;
  actual_distance:=distance ([initialpoint[1],initialpoint[2]],
  [given_point[1],given_point[2]]);
```

Calculates the geodesic circle radius of "nh/v" and the radius of "(n+1)h/v" and the actual distance of the geodesic.

Bibliography

[1] Richard L. Burden, J. Douglas Faires, *Numerical Analysis*. Prentice-Hall, [1976].

[2] Manfredo P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall Inc, [1976].

[3] C. Henry Edwards, David E. Penny, *Differential Equations and Boundary Value Problems: Computing and Modeling*, Pearson Education, Third Edition [2004].

[4] C. Henry Edwards, David E. Penny, *Multivariable Calculus*, Pearson Education, Sixth Edition [2002].

[5] V. Efremovic, A. Vainstein, E. Gorelik, E, Loginov, *Metric fibrations on Riemannian manifolds*, Colloq. Math. Soc. J. Bolyai, 195-198 [1979].

[6] Alfred Gray, *Modern Differential Geometry of Curves and Surfaces*, CRC Press, [1997].

[7] John Oprea, *Differential Geometry and Its Applications*, Prentice Hall, [2003].

[8] Vladimir Rovenski, *Geometry of Curves and Surfaces with Maple*, Brikhauser. [2002].

[9] E. L. Schwartz, A. Shaw, E. Wolfson, *A Numerical Solution to the Generalized Mapmaker's Problem: Flattening Nonconvex Polyhedral Surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 9, 1005-1008 [1989].

[10] P. C. Teo, G. Sapiro, B. A. Wandell, *Creating Connected Representations of Cortical Gray Matter for Functional MRI Visualization*, IEEE Transactions on Medical Imaging, Vol. 16, No. 6, 852-863 [2004].

[11] D. C. Van Essen, H. A. Drury, S. Joshi, M. I. Miller, *Functional and structural mapping of human cerebral cortex: Solutions are in the surfaces*, Proc. Natl. Acad. Sci., Vol. 95, 788-795 [1998].

[12] D. C. Van Essen, H. A. Drury, S. Joshi, M. I. Miller, *Structural and Functional Analyses of Human Cerebral Cortex Using a Surface-Based Atlas*, The Journal of Neuroscience, Vol. 17, No. 18, 7079-7102 [1997].

[13] L. Xing, E. J. Candes, *Fast geodesics computation with the phase flow method*, Journal of Computational Physics, Vol. 220, 6-18 [2006].