GEOMETRIC AND NUMERICAL METHODS FOR BONNET PROBLEMS AND SURFACE CONSTRUCTION

by

Zeynep Sultan Kose, B.S., M.S.

A Dissertation

In

Mathematics and Statistics

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Approved

Dr. Magdalena Toda
Chairperson of the Committee

Dr. Eugenio Aulisa
Co-Chairperson of the Committee

Dr. Akif Ibragimov

Dr. Ram Iyer

Dr. Fred Hartmeister
Dean of the Graduate School

December, 2010

ACKNOWLEDGEMENTS

I wish to acknowledge the help and support of my advisors: Dr Magdalena Toda and Dr Eugenio Aulisa. This dissertation couldn't have been written without them who not only served as my supervisors but also encouraged and supported me through my academic program. They were like my family and I would like to thank them because of their guidance, patience, politeness, help and instructions.

A special recognition needs to be given to Dr Ram Iyer and Dr Akif Ibragimov for their innovative ideas, comments, suggestions and support to this research.

I dedicated this dissertation to my Fiance Serkan Kose and my family: Asuman Kose, Elif Akdogan, Oguzhan Kose for their love and support in every way possible throughout the proccess of this dissertation.

TABLE OF CONTENTS

ABSTRACT

This PhD dissertation studies several important problems in differential geometry and its numerical applications. Its novel contributions to these areas include the following:

- solving Bonnet problems using the Cartan theory of moving frames, Cartan structure equations, and numerical analysis;
- studying surface duality in Riemannian geometries, in terms of a certain duality between mean curvature and the Hopf differential factor;
- determining sufficient geometric conditions for a surface to admit isothermic coordinates;
- constructing surfaces that admit isothermic coordinates; determining isothermic coordinate charts when starting from an arbitrary chart; implementing these methods numerically.

The moving frames and Cartan structure equations are written in terms of the first and second fundamental forms, and the Lax system is consequently reinterpreted; orthonormal moving frames are obtained as solutions to the Bonnet-Lax system, via numerical integration. Certain classifications of families of surfaces are studied in terms of the first and second fundamental forms, with respect to certain prescribed invariants. Numerical methods are applied to this theoretical framework in order to solve Bonnet problems, construct isothermic coordinate charts for surfaces that admit them, and construct dual surfaces (Christoffel transforms). Several visual examples are provided, as well as the corresponding numerical methods and code.

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The initial motivation of this project came forward from solving Bonnet problems analytically and numerically, using the Cartan theory of structure equations, and numerical analysis. The second chapter reviews important classical concepts of differential geometry which are extensively used in this work, such as first and second fundamental forms, normal curvature, shape operator, mean and Gaussian curvatures. The third chapter deals with Christoffel transforms (in the sense of families of dual surfaces, that is, surfaces with parallel tangent planes with respect to a 1-1 pointwise correspondence). Such transforms exist exclusively for isothermic coordinate charts, that is, isothermal curvature line parameterizations. Unlike the associate families, dual surfaces do not preserve the metric, but they preserve the unit normal vector field up to a change in orientation.

In the fourth chapter, Bonnet problems are expressed in terms of Cartan theory in conformal coordinates (which exist for all regular surfaces), that is, in the most general context. That is, for any compatible first and second fundamental form, the corresponding families of solutions are obtained (up to roto-translations). Solving Bonnet problems via the Cartan-Lax system lead us in a natural way to analyzing families of surfaces which preserve certain characteristics invariant. It is shown that surface duality is equivalent to a duality between the Hopf differential and the mean curvature function, for any immersion in isothermic coordinates.

Due to the high impact of isothermic coordinates in this study, it became imperative to provide a necessary condition for the existence (and uniqueness) of isothermic coordinates. Such a condition was obtained in Chapter 5, and it contains several important classes of surfaces as particular examples, including CMC surfaces.

We provide a method of constructing isothermic coordinate charts on surfaces that admit them, starting from an arbitrary chart (parameterization). The method is presented both theoretically and numerically. Both Bonnet Problems and changes of coordinate charts are numerically solved, and the solutions are visualized. Certain relevant families of associate surfaces and Christoffel transformations (dual surfaces) are illustrated, in particular for CMC surfaces, which present a particular interest from mathematical and physical view points.

CHAPTER 2
FUNDAMENTAL CONCEPTS OF SURFACE THEORY

This work exclusively involves immersed (respectively, embedded) surfaces in the Euclidean 3-space. One of the primary goals of this work consists of numerical applications and surface visualization, including numerical solutions to Bonnet Problems. Many of these results can be generalized to immersions of 2-manifolds in n-dimensional space forms, but this generalization would not be compatible with our numerical implementation and visualization.

Definition 2.0.1. *A* **local surface** *is a differentiable mapping from an open, simply connected subset of the Euclidean plane, to* $\mathbb{R}^3$*. Various references use the name of* **patch of a surface***, or* **local patch** *instead of local surface. We will call such a local surface a* **surface immersion***, if the Jacobian of the differential mapping has rank two at every point. This is equivalent to the differential map being 1-1. Remark that such a surface is generally not embedded. If the local surface (immersion) represents an injective mapping in itself, then we will call such a mapping an* **embedding***, or an* **injective patch***.*

In a more global sense, a connected set M *in* $\mathbb{R}^3$ *is said to be a* **regular 2-dimensional surface** *if for an arbitrary point* $p \in M$ *there exist an open ball* U_p *in* $\mathbb{R}^3$ *with center at* p *and a homeomorphism* ψ *which maps* $M \cap U_p$ *onto an open disk on some plane in the space* $\mathbb{R}^3$*. Such a homeomorphism is frequently called a* **chart***, or a* **system of local coordinates on a surface***.*

It is worth noting that, in the previous definition, the word disk can be replaced by an arbitrary open set of a plane, diffeomorphic to a disk.

2.1 First Fundamental Form of a Surface

Let us consider an immersion $r = r(u, v) = (X(u, v), Y(u, v), Z(u, v))$ (of an open, simply connected disk) in $\mathbb{R}^3$. Note that the property of this map $r = r(u, v)$ being an immersion is equivalent to the property that the vectors $\frac{\partial r}{\partial u}$ and $\frac{\partial r}{\partial v}$ are linearly independent at every point. These span a tangent plane at each point, that is, generate a tangent bundle. We will denote $N = \frac{r_u \times r_v}{|r_u \times r_v|}$ and call it the unit normal vector field, or Gauss map; it represents a map $N : D \to S^2$.

Definition 2.1.1. *We will call the following quadratic form*

$$ds^2(u,v) = < r_u, r_u > du^2 + 2 < r_u, r_v > dudv + < r_v, r_v > dv^2 \tag{2.1}$$

a naturally induced metric, or **first fundamental form**.

Remark that the first fundamental form is nothing but the inner product between tangent vectors. It can be also defined as a map that acts as follows:

$I(v_p, w_p) = < v_p, w_p >$, where the vectors v_p and w_p represent tangent vectors to the surface at the point p.

Recall that locally, every smooth surface can be represented as a graph (Monge chart) $z = f(u,v)$ over an open domain $D \subset \mathbb{R}^2$. Parametrically, this is written as $r(u,v) = (u, v, f(u,v))$. The Cartesian coordinates in D are then mapped by f onto coordinate lines on the surface.

In the **particular** case of a Monge chart (graph), the induced metric (first fundamental form) can be written as

$$ds^2 = (1 + f_u^2)du^2 + 2f_u f_v dudv + (1 + f_v^2)dv^2 \tag{2.2}$$

In general, we will not work with Monge charts, but with generic immersions.

If we define $g_{11} = |r_u|^2$, $g_{12} = < r_u, r_v >$, $g_{22} = |r_v|^2$, then we can identify the naturally induced metric (2-form)

$$ds^2 = g_{11}du^2 + 2g_{12}dudv + g_{22}dv^2 \tag{2.3}$$

with a symmetric and positive definite real-valued matrix.

Besides the equation (2.3), the following notation, which was introduced by

Gauss, can also be found in the literature as:

$$I : ds^2 = Edu^2 + 2Fdudv + Gdv^2. \tag{2.4}$$

The first fundamental form is often considered in matrix form, that is:

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

Theorem 2.1.1. *It is easy to see that at regular points on a surface, the first fundamental form (2.4) is positive definite. [5]*

Note that, if the actual immersion a surface is unknown and only the first fundamental form is given, then we do not possess much information about the geometric properties of the surface.

On the other hand, the first fundamental form of a surface yields information about the intrinsic geometry of the surface. The metric determines the Gaussian curvature (*Theorema Egregium - The Remarkable Theorem of Gauss*), length of a curve, area of a region, geodesics.

By a classical result of Bonnet, the geometry of surfaces exclusively depends on two quadratic differential forms. One of them is the metric, and the other is the second fundamental form.

2.2 Second Fundamental Form of a Surface

We will review a few basic notions of surface theory, such as normal curvature and second fundamental form, using [4].

Definition 2.2.1. *For a local immersion in mathbbR^3, with image $M = r(D)$, let N represent a normal vector field. For a tangent vector $v = v_p$ to M at p, we put*

$S_p(v) = -dN_p(v) = -D_vN(p)$.

Then S_p, viewed as a linear map from the tangent plane to itself, defines the shape operator S at every point.

We define the normal curvature as:

$$\kappa_n(v_p) = \frac{< S(v_p), v_p >}{< v_p, v_p >}$$

Clearly, if v_p is unitary, then the denominator is equal to 1.

Definition 2.2.2. *The second fundamental form of a regular surface M in $\mathbb{R}^3$ represents a symmetric and bilinear form on the tangent plane to M at the point p given by:*

$II(v_p, w_p) =< S(v_p), w_p >$ *for any arbitrary tangent vectors on the tangent plane at p.*

Based on the previous computations, another way of writing the second fundamental form is:

$$II = ldu^2 + 2mdudv + ndv^2 \tag{2.5}$$

where the coefficients l, m and n are given by: $l =< r_{uu}, N >$, $m =< r_{uv}, N >$, $n =< r_{vv}, N >$.

The second fundamental form is often denoted by II and in matrix form we have

$$II = \begin{pmatrix} l & m \\ m & n \end{pmatrix}$$

The second fundamental form was first introduced by Gauss in 1827 and unlike the first fundamental form, the second fundamental form is not necessarily positive definite. However, it is a symmetric bilinear form.

Properties of the surface expressible in terms of the second fundamental form are called *extrinsic properties* of the surface such as describing the normal components of curvature vectors, and the deviation of the surface from its tangent plane, and it keeps track of the twisting of the unit normal vector field.

2.3 Principal Curvatures

We study the normal curvature as a function:

$$\kappa_n(v_p) = \frac{< S(v_p), v_p >}{< v_p, v_p >}. \tag{2.6}$$

Since all directions in a tangent plane T_pM form a compact set homeomorphic to a circle, κ has at least one minimum and one maximum, i.e., at least two extremal values.[6] The maximum normal curvature κ_1 and the minimum normal curvature κ_2 are called principal curvatures. They represent the eigenvalues of the shape operator, and the eigenvectors corresponding to them are called principal directions.

Definition 2.3.1. *The average and the product of the two principal curvatures are the mean curvature H and Gaussian curvature K, respectively.*

$$H = \frac{\kappa_1 + \kappa_2}{2}, \tag{2.7}$$
$$K = \kappa_1 \kappa_2. \tag{2.8}$$

In the classical literature, surfaces whose mean curvature is a constant have been studied extensively.

Definition 2.3.2. *A surface with mean curvature $H = 0$ is called a minimal surface.*

Such surfaces include but are not limited to, surface of minimum area. In any case they represent critical "points" of an energy functional. No volume constraints are involved.

Definition 2.3.3. *Surfaces that satisfy the condition $H = constant \neq 0$ are called CMC surfaces.*

Both minimal and CMC surfaces can be expressed as solutions of nonlinear PDEs. CMC surfaces minimize area under a volume constraint. A special functional is introduced for this purpose, and the proof is straight-forward; this is a classical result of Variational Calculus.

We are hereby interested in certain properties such as *surface duality* (i.e., existence of a Christoffel transform). These geometric properties are not characteristic only to CMC surfaces, but to very few types of surfaces. Known examples of surfaces that admit isothermic surfaces, and hence admit a dual, include CMC surfaces, quadric surfaces and Bonnet surfaces.

CHAPTER 3
DUAL SURFACES AND DUAL ISOTHERMIC IMMERSION

We have shown that any local immersion can be endowed with a Riemanian metric g with entries g_{11}, g_{12} and g_{22}.

If the metric g is a (nonconstant) multiple of the flat metric, then the immersion is said to be **conformal**, or **isothermal**, or **in conformal coordinates**, i.e.

$\|r_u\| = \|r_v\| = F(u,v)$ and $< r_u, r_v >= 0$ at every point.

A surface in $\mathbb{R}^3$-space is said to be parameterized in **isothermic coordinates** if the given parameterization is conformal **and** in curvature line coordinates, i.e. it is given as an immersion $f : D \subset R^2 \longrightarrow R^3$ such that $< f_u, f_v >= 0$,

$$\|f_u\|^2 =< f_u, f_u >= \|f_v\|^2 =< f_v, f_v >= e^{2u} \tag{3.1}$$

for a smooth function over D and $< f_{uv}, N >\equiv 0$.

Remark that not every isothermal parameterization is isothermic. The *curvature line coordinate condition* is essential, and highly non-trivial to obtain on a given surface.

Definition 3.0.4. *Given an arbitrary immersion $f : D \longrightarrow R^3$, we call **dual** immersion f^d any surface which is obtained via a one-to-one pointwise corresponce with f, such that the corresponding tangent planes are parallel, up to an eventual change of orientation.*

Note that, if a dual f^d exists, it is not unique. It was shown by Christoffel (1867) that f^d exists if and only if f is isothermic.

Corollary 3.0.1. *If f^d exists, then f^d will also be isothermic.*

We like to make f^d unique by imposing additional conditions. Otherwise, there are infinitely many duals corresponding to an isothermic immersion f.

The dual immersion is unique up to translation and dilation (homothety) except in the totally umbilic case. [7].

Question: What is the most common way to designate a unique dual to a surface in isothermic coordinates?

The question is answered by the following definition, and the unique representative chosen out of the entire equivalence class of duals will be referred to as *normalized dual*. Although the procedure existed before and was presented in works by A. Bobenko and T. Hoffman, the term of **normalized dual surface** has not been used anywhere in the literature: it is used in this work for the first time.

Definition 3.0.5. *If $f : D \subset R^2 \to R^3$ is an isothermic immersion, then the* **normalized dual** *isothermic immersion is defined* ***uniquely*** *by*

$$df^* = e^{-2u}\left(f_u du - f_v dv\right). \tag{3.2}$$

Given the isothermic immersion f, let us study the characteristics of its normalized dual f^* in terms of:

1. Metric (first fundamental form);
2. Gauss normal map;
3. Second fundamental form.

3.0.1 Metric

From the definition 2 and definition 4 , note that we can rewrite:

$$df^* = f_u^* du + f_v^* dv = \frac{f_u}{\|f_u\|^2} du - \frac{f_v}{\|f_v\|^2} dv \tag{3.3}$$

The first fundamental form of the dual f^* is:

$$< df^*, df^* >= \|f_u^*\|^2 du^2 + 2 < f_u^*, f_v^* > dudv + \|f_v^*\|^2 dv^2 =$$
$$= \frac{1}{\|f_u\|^2} du^2 + \frac{1}{\|f_v\|^2} dv^2 = e^{-2u}\left(du^2 + dv^2\right).$$

3.0.2 Gauss Normal Map

$$N^* = -N. \tag{3.4}$$

Note that dual families do have the same Gauss map up to change in sign/orientation, but they are not isometric, and hence, *in general*, do not belong to the same associate family.

3.0.3 Second Fundamental Form

The second fundamental form for the immersion f is:

$$d\sigma^2 = ldu^2 + 2mdudv + ndv^2, \tag{3.5}$$

where

$$\begin{aligned} l &=< f_{uu}, N >; \\ m &=< f_{uv}, N >= 0; \\ n &=< f_{vv}, N > . \end{aligned}$$

The second fundamental form of the normalized dual surface f^* follows after a straight-forward computation, and can be written as:

$$(d\sigma^*)^2 = l^*du^2 + 2m^*dudv + n^*dv^2, \tag{3.6}$$

where

$$\begin{aligned} l^* &=< \frac{f_{uu}}{\|f_u\|^2}, -N >= -e^{-2u}l \\ m^* &=< \left(\frac{1}{\|f_u\|^2} - \frac{1}{\|f_v\|^2}\right) f_{uv}, -N >= 0 \\ n^* &=< \frac{-f_{vv}}{\|f_v\|}, -N >= e^{-2u}n. \end{aligned}$$

The mean curvature of the immersion f is:

$$H = \frac{e^{-2u}}{2}(l+n). \tag{3.7}$$

The corresponding mean curvature of the normalized dual f^* is immediate:

$$H^* = \frac{E^*n^* - 2F^*m^* + G^*l^*}{2\,(E^*G^* - (F^*)^2)}$$
$$= \frac{e^{2u}}{2}(n^* + l^*) = \frac{n-l}{2}. \tag{3.8}$$

3.1 Hopf Differential

The Hopf differential of f and respectively, f^* can be expressed as Qdz^2, and respectively, Q^*dz^2, where

$$Q =< f_{zz}, N >= \frac{1}{4}(l-n). \tag{3.9}$$

and

$$Q^* =< f^*_{zz}, N^* >= \frac{1}{4}(l^* - n^* - 2im^*) = -\frac{e^{-2u}}{4}(l+n) \tag{3.10}$$

Using equations (3.7) and (3.10), we can write the following relation

$$Q^* = -\frac{H}{2} \tag{3.11}$$

and by equations (3.8) and (3.9), it follows that

$$Q = -\frac{H^*}{2}. \tag{3.12}$$

The duality between mean curvature and Hopf differentials

Proposition 3.1.1. *For any pair of normalized dual immersions f and f^*, whose orientation is the same or opposite in sign, the corresponding (halved) mean curvature and Hopf differential factor satisfy a duality relationship, through the following identities:*
$|Q| = \frac{|H^*|}{2}$ *and* $|Q^*| = \frac{|H|}{2}$.

Proof. The proposition is a direct consequence of the computations that precede it. For the dual immersions f and f^*, normalized as above, the orientations will be opposite in sign, in which case, $Q = -\frac{H^*}{2}$ and $Q^* = -\frac{H}{2}$. However, if one decides to change the orientation of the dual to the exact same orientation of the original surface, the minus sign will become plus. Hence, the absolute value sign. □

To our knowledge, this proposition/observation has never been stated in the literature before.

3.2 Darboux Transformations

It is well know that for CMC surfaces, the Hopf differential is a holomorphic quadratic differential. Thus, about every point away from the umbilical locus we can choose an isothermal coordinate system (x, y) in which the Hopf differential has the local representation dz^2 where $z = x + iy$. The coordinate lines of such a coordinate system are curvature lines This procedure will not work at umbilic points at which the local representation of the Hopf differential is $z^m dz^2$. The sphere is a special, exceptional example, since its Hopf differential is identically zero. Every umbilic-free constant mean curvature immersion is isothermic [7].

Darboux transformations are of special geometric and historical interest, with their origin in the nineteenth century. The latter arose in a study by Darboux in 1882, of Sturm-Lioiville problems. However, they are but a special case of transformations due to Moutard and introduced earlier in 1878 in connection with the sequential reduction of linear hyperbolic equations to canonical form. Iterated Darboux transformations were constructed by Crum in 1955 in connection with related Sturm-Liouville problems. In 1975, the Crum transformation was taken up by Wadati et al. and used to generate multi-soliton solutions of integrable equations. By 1976, it was clear that Darboux transformations, with their origins in the classical differential geometry of surfaces, have deep connections with soliton theory ([10]). A key tool in the study of of Darboux transformations of an isothermic surface in Euclidean space is a careful analysis of the Christoffel transformation (or dual isothermic surface) of the surface, which maybe considered as a certain limiting case of Darboux transformations ([9]).

Theorem 3.2.1. *The correctly scaled and positioned Christoffel transform f^d of an isothermic surface $f : D \to \mathbb{R}^3$ is also a Darboux transform $\hat{f}$ of f if and only if f is a surface of constant mean curvature $H \neq 0$. In this case $\hat{f}$ is parallel surface of CMC where $\hat{f} = f + \frac{1}{H}N$.[9]*

$g = f + \lambda N$ is called a parallel immersion to original f at distance λ from f. So, a natural question arises whether the Gauss map of the parallel immersion g is equal to the Gauss map of the parallel immersion f? The answer is no, in general. But, if $\lambda = \frac{1}{H}$, the Gauss map is the same (up to a change in orientation).

3.3 Constructing Dual Surfaces for Surfaces in Isothermic Coordinates

The last Chapter, 5, presents a method of constructing parameterizations in isothermic coordinates (β, γ) on surfaces that are known to admit them, starting from an arbitrary immersion, given in coordinates (x, y).

For details about notations used, please first refer to Chapter 5.

In practice, we integrate along γ and β and draw a virtual mesh to construct the surfaces.

$$f(\gamma, \beta) = f(0, \beta) + \int_0^{\gamma} f(\bar{\gamma}(x, y), \beta(x, y))\, d\bar{\gamma}. \tag{3.13}$$

We start constructing the surface in a neighborhood of $(0, 0)$ by solving a succession of ODE's.

First, we fix $\beta = 0$, and move along the γ.

Note that we are moving along (γ, β), but our functions are all in terms of old parameters (x, y), and virtual displacements in old versus new systems of coordinates are all kept track of.

The system will be solved with a combined procedure:

1. Solve the vector field witk Runge-Kutta method and find (x, y),
2. Construct the surface with solving the system of PDEs with Runke-Kutta method.

Next we fix γ, and move along the β coordinate line, in a similar way to the previous procedure.

3.4 Some CMC Surfaces and Duals

Unduloid is an example of a CMC-Delaunay surface which can be parametrized:

$$f(x, y) = \begin{pmatrix} X(x)\cos y \\ X(x)\sin y \\ Z(x) \end{pmatrix}$$

where

$$\begin{aligned} X(x) &= (1 + d^2 + 2d\sin x)^{1/2}, \\ Z(x) &= \int \frac{1 + d\sin x}{X(x)} dx, \\ &\quad 0 < d < 1. \end{aligned}$$

Remark that the immersion $f(x, y)$ is not conformal. Towards a pair of isothermic coordinates, we are performing some necessary changes of variables. The first change in variable is $\eta = \int \frac{1}{X(x)} dx$.

Ultimately, for finding f^*, we are using transformation (3.2), leading to a normalized dual frame. However, this equation provides just the dual frame (i.e., partial velocities of f^*); in order to obtain the dual immersion f^* explicitely, one needs to integrate, after selecting a base point.

Picard's Theorem guarantees that "two-planes fit together" if the compatibility condition holds. For a smooth solution to a system of equations, the compatibility condition is equivalent to the condition that the "mixed partials must commute", i.e., $(f_x)_y = (f_y)_x$.

Consider the system corresponding to f^*:

$$f^*_x = A(\eta(x), y) \tag{3.14}$$

$$f^*_y = B(\eta(x), y) \tag{3.15}$$

We solve this system of equations in a neighborhood of $(0, 0)$ in the usual way, by solving a succession of ODE's.

This procedure gives us the dual immersion, f^*, in some neighborhood of $(0, 0)$. The solution satisfies the original system, as a consequence of the compatibility condition. [11]

Figure 3.1: Delaunay-CMC Surface and Dual

(a) Delaunay-CMC

(b) Dual Surface

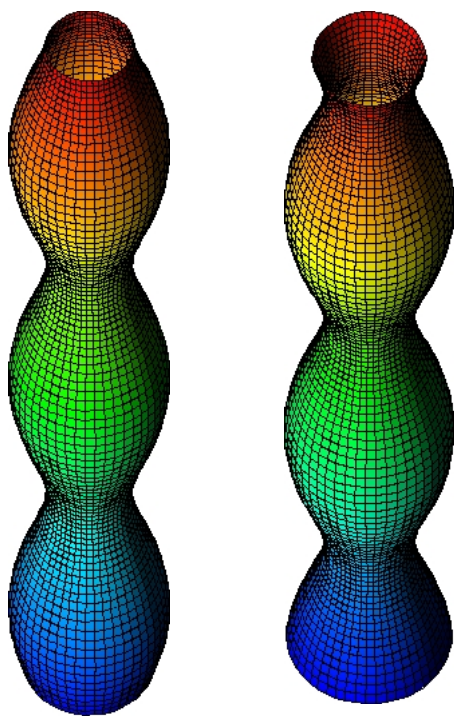
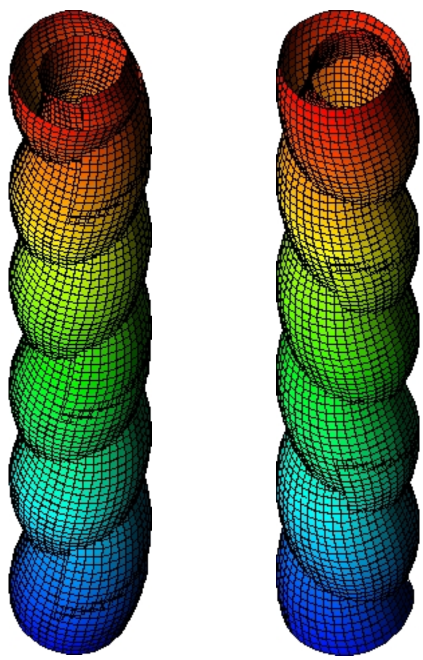


Figure 3.2: Helicoidal-CMC Surface and Dual

(a) Helicoidal CMC (b) Dual Surface



CHAPTER 4
SOLVING BONNET PROBLEMS WITH MOVING FRAMES AND CARTAN STRUCTURE EQUATIONS

In this section, Bonnet Problems are approached using Cartan moving frames and associated structure equations. The Cartan structural forms are written in terms of the first and second fundamental forms, and the Lax system is consequently reinterpreted; orthonomal moving frames are obtained solutions to this Bonnet-Lax system, via numerical integration. Certain classifications of families of surfaces are provided in terms of the second and second fundamental forms, given certain prescribed invariants. Numerical applications (an improved Runge Kutta method) applied to this theoretical framework produced a fast way to visualize families of surfaces under investigation. A few examples and visual models are provided.

Pierre Bonnet (1819-1892) had numerous and significant contributions to the field of Differential Geometry, including three famous results: the Gauss-Bonnet theorem (computing the path integral of the geodesic curvature along a closed curve); an upper bound result for the diameter of a complete Riemannian manifold when the Ricci curvature is bounded from below (what is known today as the Bonnet-Myers theorem proved in 1941); and ultimately what the geometric community called Bonnet's Theorem, a fundamental result proved by Bonnet around 1860, whose statement is as short as all the major philosophical statements, namely:

Theorem 4.0.1. *(Bonnet's Theorem) The first and second fundamental forms determine an immersed surface up to rigid motions*[1]*.*

Solving a Bonnet problem usually means obtaining an explicit formula for the immersion of a surface, starting from given first and second fundamental forms. Remark that this is impossible to achieve in general, even for the case of a regular immersion in the 3-d Euclidean space. However, Cartan's theory on structure

[1] Euclidean motions, roto-translations

equations, together with solving a **compatible** Lax system numerically, by an improved Runge-Kutta method, provides us with a consistent method of achieving this goal to its best extent.

The second part of this chapter focuses on families of surfaces which naturally arise in the study of Bonnet problems, by considering certain invariants (such as mean curvature H, or Gauss curvature K, or both). One of the families that we obtain as a bi-product of this procedure is an isometric family of Bonnet surfaces which preserve both mean curvature and Gauss curvature (i.e., preserve the principal curvatures). In particular, when the mean curvature is prescribed as constant, such a family resumes to the so-called associate surfaces.

Several classics (e.g., works of Cartan and Chern) brought contributions to the topic of studying Bonnet problems from specific view points.

In the past few decades, several authors brought new insights and significant progress to the general study of Bonnet problems (such as, George Kamberov in [14] and Alexander Bobenko in [12] and other related papers).

Our research meshes well with the preexisting work in this field, without significant overlaps and without any contradictions.

4.1 Bonnet's Theorem

Consider an immersed surface in Euclidean 3-space parametrized via the map

$$f : D \subset \mathbb{R}^2 \to \mathbb{R}^3,$$
$$f(u,v) = (f^1(u,v), f^2(u,v), f^3(u,v)).$$

where D represents an open simply connected domain in plane. The first and second fundamental forms corresponding to this immersion can be expressed as:

$$I = \langle df, df \rangle = Edu^2 + 2Fdudv + Gdv^2 \tag{4.1}$$
$$II = -\langle df, dN \rangle = ldu^2 + 2mdudv + ndv^2, \tag{4.2}$$

where $E = \langle f_u, f_u \rangle$, $F = \langle f_u, f_v \rangle$ and $G = \langle f_v, f_v \rangle$, $l = \langle f_{uu}, N \rangle$, $m = \langle f_{uv}, N \rangle$ and $n = \langle f_{vv}, N \rangle$ represent the coefficients of the first and second fundamental forms, respectively.

The shape operator can be expressed in terms of the first and second fundamental forms in matrix form as follows:

$$S = \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \begin{pmatrix} l & m \\ m & n \end{pmatrix}$$

The principal curvatures k_1, k_2 represent the eigenvalues of the shape operator. One can express the Gaussian and mean curvatures in terms of the coefficients of the first and second fundamental forms, as follows:

$$H = \frac{k_1 + k_2}{2} = \frac{1}{2}\frac{En - 2Fm + Gl}{EG - F^2} \tag{4.3}$$

$$K = k_1 k_2 = \frac{ln - m^2}{EG - F^2}. \tag{4.4}$$

Theorem 4.1.1. *(Bonnet Theorem) The first and second fundamental forms determine an immersed surface up to rigid motions*[2]*.*

The most natural context in which Bonnet's problem can be studied is that of Cartan's: Moving Frames, Equivalence Method and Structure Equations. We are recalling the following important result, which can be found in [2].

We apply the method of moving frames to the special case of surfaces in $\mathbb{R}^3$. Let $x : M^2 \to \mathbb{R}^3$ be an immersion of a two-dimensional differentiable manifold in $\mathbb{R}^3$. For each point $p \in M$, an inner product $\langle , \rangle_p$ is defined in T_pM by the rule:

$$\left\langle v^1, v^2 \right\rangle_p = \left\langle dx_p(v^1), dx_p(v^2) \right\rangle_{x(p)}, \tag{4.5}$$

[2]Euclidean motions, roto-translations

where the inner product in the right hand side is the canonical inner product of $\mathbb{R}^3$. It is straightforward to check that $\langle , \rangle_p$ is differentiable and defines a Riemannian metric in M^2, called the *metric induced by the immersion x.*

We will study the local geometry of M around a point $p \in M$. Let $U \subset M$ be a neighborhood of p such that the restriction $x|_U$ is an embedding. Let $V \subset \mathbb{R}^3$ be a neighborhood of p in $\mathbb{R}^3$ such that $V \cap x(M) = x(U)$ and such that we can choose in V an adapted moving frame e^1, e^2, e^3. Therefore, when restricted to $x(U)$, e^1 and e^2 span the tangent bundle to $x(U)$.

Each vector field e^i is a differentiable map into $\mathbb{R}^3$. The differential at $p \in D$, $(de^i)_p : \mathbb{R}^3 \to \mathbb{R}^3$, is a linear map. Thus, for each p and each $v \in \mathbb{R}^n$ we can write

$$(de^i)_p(v) = \sum_j (\omega^i_j)_p(v)e^j.$$

The expressions $(\omega^i_j)_p(v)$, above defined, depend linearly on v. Thus $(\omega^i_j)_p$ is a linear form in $\mathbb{R}^3$ and, since e^i is a differentiable vector field, ω^i_j is a differential 1-form. So, we write the above as

$$de^i = \sum_j \omega^i_j e^j. \tag{4.6}$$

The forms ω^i_j so defined are called *connection forms* corresponding to the moving frame e^i.

The map $x : U \to V$ induces forms $x^*(\omega_i),\ x^*(\omega^i_j)$ on U. Since x^* commutes with d and $\wedge$, such forms satisfy Cartan's equations. For all $q \in U$ and all $v \in T_qM$, it follows that

$$x^*(\omega_3)(v) = \omega_3(dx(v)) = 0.$$

Following [2], by a well-motivated abuse of notation, we identify $x^*(\omega_i) = \omega_i$, and $x^*(\omega^i_j) = \omega^i_j$. $x|_U$ represents an embedding and these restricted forms satisfy

Cartan's structure equations, where $\omega_3 = 0$.

In V we have, associated to the frame e^i, the coframe forms ω_i and the connection forms $\omega^i_j = -\omega^j_i$, $i, j = 1, 2, 3$ which satisfy the structure equations:

$$
\begin{aligned}
d\omega_1 &= \omega_2 \wedge \omega^2_1, \\
d\omega_2 &= \omega_1 \wedge \omega^1_2, \\
d\omega_3 &= \omega_1 \wedge \omega^1_3 + \omega_2 \wedge \omega^2_3, \\
d\omega^1_2 &= \omega^1_3 \wedge \omega^3_2, \\
d\omega^1_3 &= \omega^1_2 \wedge \omega^2_3, \\
d\omega^2_3 &= \omega^2_1 \wedge \omega^1_3.
\end{aligned}
$$

The 4th listed equation among these structure equations is called *Gauss equation.*

The 5th and 6th (last two) equations are called *Codazzi-Mainardi-Peterson equations*, or sometimes just Codazzi equations, in the literature.

Now, let $f : D \subset \mathbb{R}^2 \to \mathbb{R}^3$ be a parametrization of an immersed surface such that $(u, v) \in D$ are orthogonal, $E = \langle f_u, f_u \rangle$, $F = 0$ and $G = \langle f_v, f_v \rangle$. Then, we can choose an orthonormal frame $e^1 = \frac{f_u}{||f_u||}$, $e^2 = \frac{f_v}{||f_v||}$ in D and given the moving frame e^i we will define differential 1-forms ω_i which is the asociated coframe to e^i by the conditions $\omega_i(e^j) = \delta_{ij}$. The following expressions are an immediate consequence of the coframe conditions:

$$
\begin{aligned}
\omega_1 &= \sqrt{E}du \\
\omega_2 &= \sqrt{G}dv.
\end{aligned}
$$

Next,

$$
d\omega_3 = \omega_1 \wedge \omega^1_3 + \omega_2 \wedge \omega^2_3 = 0,
$$

hence, by Cartan's lemma,

$$\omega_3^1 = h_{11}\omega_1 + h_{12}\omega_2, \tag{4.7}$$

$$\omega_3^2 = h_{21}\omega_1 + h_{22}\omega_2, \tag{4.8}$$

where $h_{ij} = h_{ji}$ are differentiable functions in U. Remark that the matrix h can be reinterpreted as a shape operator, and it corresponds to the Weingarten operator $-dN(v) = S(v)$.

It is very important to remark (see [15]) that the shape operator S is related to both the first and second fundamental forms through the formula

$II(v, w) = \langle S(v), w\rangle_g$, where the inner product is considered with respect to the first fundamental form $I \equiv g$, and so

$S = I^{-1}II$.

As [15] remarks (see p. 177), *this matrix in general is not symmetric, even though the shape operator is symmetric with respect to the inner products on the tangent spaces.*

On the other hand, Cartan's Lemma yields a matrix h which is symmetric, and represents a symmetrization of the matrix S. This aspect, among others, shows how natural, and much more convenient, it is to work with Cartan forms instead of the classical approach to differential geometry.

In matrix form, the (similar) matrices S and h can be represented as (in the case when the parameterization is orthogonal, $F = 0$):

$$S = \begin{pmatrix} \frac{l}{E} & \frac{m}{E} \\ \frac{m}{G} & \frac{n}{G} \end{pmatrix}, \tag{4.9}$$

while

$$h = \begin{pmatrix} \frac{l}{E} & \frac{m}{\sqrt{EG}} \\ \frac{m}{\sqrt{EG}} & \frac{n}{G} \end{pmatrix}. \tag{4.10}$$

Remark that matrices S and h have the same eigenvalues, namely the principal curvatures k_1 and k_2. Consequently, H and K are invariants with respect to this symmetrization.

This allows us to rewrite equations (4.7), (4.8) for any parametrized surface where f_u and f_v are orthogonal:

$$\omega_3^1 = \frac{l}{\sqrt{E}}du + \frac{m}{\sqrt{E}}dv, \tag{4.11}$$

$$\omega_3^2 = \frac{m}{\sqrt{G}}du + \frac{n}{\sqrt{G}}dv \tag{4.12}$$

One more connection form needs to be determined, namely $\omega_2^1 = -\omega_1^2$, which can be obtained from the Theorem of Levi-Civita.

Lemma 1. *In isothermal coordinates, the Levi-Civita connection has the following expression:*

$$\omega_2^1 = -\frac{(\sqrt{E})_v}{\sqrt{G}}du + \frac{(\sqrt{G})_u}{\sqrt{E}}dv. \tag{4.13}$$

Proof of the Lemma. The Theorem of Levi-Civita states that on any 2-dimensional manifold M, for any open set U in M where a moving frame e^1, e^2 is defined, together with its associated frame ω^1, ω^2, there exists a unique one-form $\omega_1^2 = -\omega_2^1$ which satisfies the following conditions:

$$\omega_1^2 \wedge \omega_2 = d\omega_1,$$
$$\omega_2^1 \wedge \omega_1 = d\omega_2.$$

On the other hand, by definition, for any differential 1-form $\omega = \sum_i a_i dx_i$, we have:

$$d\omega = \sum_i da_i \wedge dx_i.$$

The previously-stated expressions of $d\omega_1$ and $d\omega_2$, together with the expressions of the coframe $\omega_1 = \sqrt{E}du$, and respectively, $\omega_2 = \sqrt{G}dv$, lead us to the following expression of the Levi-Civitta connection form ω_2^1:

$$\omega_2^1 = -\frac{(\sqrt{E})_v}{\sqrt{G}}du + \frac{(\sqrt{G})_u}{\sqrt{E}}dv.$$

This concludes the proof.

Hence, by the equation (4.6) we can construct a system of equations (classically called Lax System of equations):

$$d\mathcal{F} = \mathcal{F}\Omega, \tag{4.14}$$

where $\mathcal{F} =< e^1, e^2, e^3 >$ represents the orthonormal frame matrix whose vectors e^i are ordered as column vectors, and Ω represents the transposed Cartan matrix. The matrix Ω is written in standard matrix notation as: $\Omega = (\Omega_{ij}) = (\omega_i^j)$, $i, j = 1, 2, 3$, where i represents the row index and j represents the column index.

The Maurer-Cartan form Ω is valued in the Lie algebra $\mathrm{so}(3, \mathrm{R})$ and so $\omega_j^i = -\omega_i^j$. This form, corresponding to a smooth immersion of prescribed first fundamental form and second fundamental form, can be explicitely written $\Omega = \mathcal{F}^{-1} \cdot d\mathcal{F}$ writes as:

$$\Omega = \begin{pmatrix} 0 & \frac{(\sqrt{E})_v}{\sqrt{G}} & -\frac{l}{\sqrt{E}} \\ -\frac{(\sqrt{E})_v}{\sqrt{G}} & 0 & -\frac{m}{\sqrt{G}} \\ \frac{l}{\sqrt{E}} & \frac{m}{\sqrt{G}} & 0 \end{pmatrix} du + \begin{pmatrix} 0 & -\frac{(\sqrt{G})_u}{\sqrt{E}} & -\frac{m}{\sqrt{E}} \\ \frac{(\sqrt{G})_u}{\sqrt{E}} & 0 & -\frac{n}{\sqrt{G}} \\ \frac{m}{\sqrt{E}} & \frac{n}{\sqrt{G}} & 0 \end{pmatrix} dv.$$

and

$$\mathcal{F}_u = \mathcal{F} \begin{pmatrix} 0 & \frac{(\sqrt{E})_v}{\sqrt{G}} & -\frac{l}{\sqrt{E}} \\ -\frac{(\sqrt{E})_v}{\sqrt{G}} & 0 & -\frac{m}{\sqrt{G}} \\ \frac{l}{\sqrt{E}} & \frac{m}{\sqrt{G}} & 0 \end{pmatrix} \tag{4.15}$$

$$\mathcal{F}_v = \mathcal{F} \begin{pmatrix} 0 & -\frac{(\sqrt{G})_u}{\sqrt{E}} & -\frac{m}{\sqrt{E}} \\ \frac{(\sqrt{G})_u}{\sqrt{E}} & 0 & -\frac{n}{\sqrt{G}} \\ \frac{m}{\sqrt{E}} & \frac{n}{\sqrt{G}} & 0 \end{pmatrix} \tag{4.16}$$

where E, F, G and l, m, n represent the coefficient functions of the first and second fundamental forms, respectively.

Further, we can write the following explicit formulas:

$$\mathcal{F}_u = \left\langle \frac{f_{uu}\sqrt{E} - f_u(\sqrt{E})_u}{E}, \frac{f_{vu}\sqrt{G} - f_v(\sqrt{G})_u}{G}, (\frac{f_u}{\sqrt{E}} \times \frac{f_v}{\sqrt{G}})_u \right\rangle,$$
$$\mathcal{F}_v = \left\langle \frac{f_{uv}\sqrt{E} - f_u(\sqrt{E})_v}{E}, \frac{f_{vv}\sqrt{G} - f_v(\sqrt{G})_v}{G}, (\frac{f_u}{\sqrt{E}} \times \frac{f_v}{\sqrt{G}})_v \right\rangle.$$

On the Lax system of differential equations, one needs to ask: Are there any solutions, and if so, how many?

The answer is given by Picard in case of a single ordinary differential equation (ODE) which is classically known as Picard's Theorem.

Let's consider our system of PDE:

$$\begin{aligned} \mathcal{F}_u &= \mathcal{F}A(u, v) \\ \mathcal{F}_v &= \mathcal{F}B(u, v) \end{aligned} \tag{4.17}$$

where $A(u, v), B(u, v) \in so(3)$, $Adu + Bdv = \Omega$.

We can attempt to solve the Lax system in a neighborhood of $(0, 0)$ by solving a succession of ODE's. In each step, Picard's Theorem implies the existence and uniqueness of a solution to the corresponding ODE system, depending only on the initial value. At every step, the compatibility condition for the Lax system is verified, namely $F_{uv} = F_{vu}$; this condition reduces to:

$$BA + A_v = AB + B_u \tag{4.18}$$

which can be rewritten as $A_v - B_u - [A, B] = 0$.

In the following theorem, we will show that the compatibility condition for the Lax system is equivalent to the Gauss-Codazzi Mainardi equations.

Theorem 4.1.2. *Let $Edu^2 + Gdv^2$ represent a positive definite bilinear form, called metric tensor (I) , and $ldu^2 + 2mdudv + ndv^2$ represent a symmetric and positive* *definite bilinear form denoted as (II). Assume that all coefficients of these forms are at least class C^2. Assume that together, these coefficients satisfy the Gauss-Codazzi Mainardi equations in a simply connected open subset of $\mathbb{R}^2$. Then, there exists a unique solution of the system*

$$\begin{aligned} \mathcal{F}_u &= \mathcal{F}A(u,v) \\ \mathcal{F}_v &= \mathcal{F}B(u,v) \end{aligned} \tag{4.19}$$

where $A(u,v), B(u,v) \in so(3)$

$$A = \begin{pmatrix} 0 & \frac{(\sqrt{E})_v}{\sqrt{G}} & -\frac{l}{\sqrt{E}} \\ -\frac{(\sqrt{E})_v}{\sqrt{G}} & 0 & -\frac{m}{\sqrt{G}} \\ \frac{l}{\sqrt{E}} & \frac{m}{\sqrt{G}} & 0 \end{pmatrix};$$
$$B = \begin{pmatrix} 0 & -\frac{(\sqrt{G})_u}{\sqrt{E}} & -\frac{m}{\sqrt{E}} \\ \frac{(\sqrt{G})_u}{\sqrt{E}} & 0 & -\frac{n}{\sqrt{G}} \\ \frac{m}{\sqrt{E}} & \frac{n}{\sqrt{G}} & 0 \end{pmatrix}$$

depending only on the choice of the initial value. This solution represents the orthonormal moving frame of a surface immersion in $\mathbb{R}^3$ that admits (I) and (II) as first and second fundamental forms, respectively. This surface is unique up to a rigid motion in space.

Conversely, if this Lax system admits a solution $\mathcal{F}$, then it corresponds to a surface immersion whose Gauss-Codazzi-Mainardi equations represent the compatibility conditions of the Lax system.

Proof. We know that the existence and uniqueness of a solution to this system depends on the compatibility condition. Let's recall the compatibility condition

$$AB - BA = A_v - B_u \tag{4.20}$$

Here after some computations, we will have the right and left hand side of the equation (4.20) as

$$AB - BA = \begin{pmatrix} 0 & -\frac{ln-m^2}{\sqrt{EG}} & -\frac{(\sqrt{E})_v n}{G} - \frac{(\sqrt{G})_u m}{\sqrt{EG}} \\ \frac{ln-m^2}{\sqrt{EG}} & 0 & \frac{(\sqrt{E})_v m}{\sqrt{EG}} + \frac{(\sqrt{G})_u n}{E} \\ \frac{(\sqrt{E})_v n}{G} + \frac{(\sqrt{G})_u m}{\sqrt{EG}} & -\frac{(\sqrt{E})_v m}{\sqrt{EG}} - \frac{(\sqrt{G})_u n}{E} & 0 \end{pmatrix}$$

$$A_v - B_u = \begin{pmatrix} 0 & (\frac{(\sqrt{E})_v}{\sqrt{G}})_v + (\frac{(\sqrt{G})_u}{\sqrt{E}})_u & (-\frac{l}{\sqrt{E}})_v + (\frac{m}{\sqrt{E}})_u \\ -(\frac{(\sqrt{E})_v}{\sqrt{G}})_v - (\frac{(\sqrt{G})_u}{\sqrt{E}})_u & 0 & -(\frac{m}{\sqrt{G}})_v + (\frac{n}{\sqrt{G}})_u \\ (\frac{l}{\sqrt{E}})_v - (\frac{m}{\sqrt{E}})_u & (\frac{m}{\sqrt{G}})_v - (\frac{n}{\sqrt{G}})_u & 0 \end{pmatrix}$$

Since both matrices are skew-symmetric, then we will have the following equations in order to satisfy the compatibility condition:

$$ln - m^2 = \frac{-E_{vv} - G_{uu}}{2} + \frac{G_u E_u}{4E} + \frac{E_v^2}{4E} + \frac{G_u^2}{4G} + \frac{E_v G_v}{4G} \tag{4.21}$$

$$l_v - m_u = \frac{lE_v}{2E} + m\left(\frac{G_u}{2G} - \frac{E_u}{2E}\right) + \frac{nE_v}{2G} \tag{4.22}$$

$$m_v - n_u = -\frac{lG_u}{2E} + m\left(\frac{G_v}{2G} - \frac{E_v}{2G}\right) - \frac{nG_u}{2G} \tag{4.23}$$

which represent one of the many equivalent forms of the Gauss-Codazzi-Mainardi equations. In this group of three equations, the first equation represents the Gauss equation, while the second and third are the Codazzi-Mainardi-Peterson equations. Through the classical literature, all these equations are usually expressed in terms of Riemann-Christoffel symbols, thus making their expression extremely sophisticated and complicated. However, one can easily verify, after some heavy computation by hand or using mathematical software, that the expressions above are equivalent to the Gauss-Codazzi-Mainardi equations given in terms of Riemann-Christoffel symbols. □

Remark A. It follows from the above equation and Bonnet theorem that if we prescribe the smooth functions E, F, G, l, m, n which satisfy the required compatibility equations, as coefficients of the first and second fundamental forms of an immersion, then a solution to the above Lax system is found. It represents a moving orthonormal frame $\mathcal{F}$ corresponding to a smooth surface that is unique, up to roto-translations.

Remark B.

An important particular case is that of curvature line coordinates, which exist on all differentiable 2-manifolds, away from singularities. This means that the first and the second fundamental forms, (I) and (II), are diagonalizable simultaneously. In this case, the Gauss-Codazzi-Mainardi equations reduce to the following equations:

$$l_v = \frac{E_v}{2}\left(\frac{l}{E} + \frac{n}{G}\right) \tag{4.24}$$

$$n_u = \frac{G_u}{2}\left(\frac{l}{E} + \frac{n}{G}\right). \tag{4.25}$$

It is important to note that the expression $\left(\frac{l}{E} + \frac{n}{G}\right) = 2H$.

Remark C.

An important particular subcase of the curvature line coordinate parameterization is that of isothermic coordinates. As defined before, isothermic coordinates represent isothermal (conformal) coordinates which also represents curvature lines ($E = G$, $F = 0$ and $m = 0$). Known surfaces that admit isothermic coordinates include all constant mean curvature (CMC) surfaces, Bonnet surfaces (non-CMC), quadrics, and a few other special families.

4.2 Families of Isometric Surfaces in Curvature Line Coordinates

In the previous section, we formulated the Bonnet Problem in terms of coefficients of the first and second fundamental forms. An important geometric application consists in studying families of surfaces corresponding to certain prescribed invariants. We can characterize surfaces via the coefficients of their fundamental forms. We remark that these coefficients are not independent: namely, they need to satisfy the specific compatibility conditions.

Let us first recall that an isometric family of smooth surfaces in conformal coordinates, immersed into the Euclidean 3-space is a family of surfaces which all have the same first fundamental form given by $I := E(du^2 + dv^2))$. The formula that expresses the Gauss curvature K is provided by Gauss' 'Theorema Egregium' (the Remarkable Theorem), which is actually the same with the *Gauss Equation.*

Of course, the expression $K = \frac{ln-m^2}{detI}$ is the most common expression of the Gauss curvature, and it was adopted due to its simplicity, but this formula makes us often forget that $ln - m^2$ (and hence K) can be exclusively written in terms of E, F and G, which is a rephrasing of *Gauss' Remarkable Theorem*, stating that the Gaussian curvature K is preserved by isometries.

Recall that the Hopf differential of a smooth immersion f is defined as the quadratic form Qdz^2, where $Q =< f_{zz}, N >$, where N represents the unit normal (or Gauss map) corresponding to the immersion.

The following result is easy to prove in conformal coordinates, but it is valid in any surface coordinates:

Lemma 2. *Consider an isometric family of immersions of Riemannian metric $E(u,v)(du^2 + dv^2)$, which all have the same mean curvature H function. Then, the modulus of the Hopf differential factor $|Q|$ will be invariant for the entire family.*

Proof. In writing down the expressions for H, K (4.3),(4.4), which are direct consequences of the Gauss-Codazzi-Mainardi equations, we will get

$$l = 2HE - n \tag{4.26}$$

$$m^2 = -n^2 + 2HEn - KE^2 \tag{4.27}$$

Considering the modulus of the Hopf coefficient, we have by definition $|Q| = (l - n)^2 + 4m^2$. Further, substituting (4.26) and (4.27) into $|Q|$, we obtain

$$|Q| = (l-n)^2 + 4m^2 = 4E^2(H^2 - K). \tag{4.28}$$

Since E, H, K are invariant, so is $|Q|$. □

Remark that k_1 and k_2, the principal curvatures, are also invariant as solutions of the equation

$$k^2 - 2Hk + K = 0 \tag{4.29}$$

Note the discriminant is $\Delta = H^2 - 4K \geq 0$, which appears in the expression of the Hopf coefficient $|Q|$.

Note that property that the Hopf differential factor $|Q|$ is invariant actually expresses a rotational transformation on the Hopf differential:

$$Q \to e^{it} Q. \tag{4.30}$$

If the parameter e^{it} takes all the values of S^1, such a family is sometimes called **1-parameter family of (associate) surfaces**.

Remark that here we assume that such a 1-parameter family of isometric surfaces with prescribed mean curvature function exists! However, we would like to make it clear that such a family does not always exist, and if it does, it is called a Bonnet family of surfaces and satisfies some very interesting property.

Definition 4.2.1. *An isometric immersion f (of an open, simply connected domain D of the plane) into $\mathbb{R}^3$ is called a Bonnet surface if there exists a non-trivial 1-parameter family of isometric immersions f_t which contains $f = f_0$, such that each f_t has the same mean curvature H as that of f.*

It is actually straightforward to prove (see [18]) that for a Bonnet surface, the Hopf differential not only remains invariant in modulus, but satisfies a specific PDE.

[18] gave a few characteristic equations for a Bonnet surface (all equivalent to the given definition), including the following PDE:

Theorem 4.2.1. *([18]) Let M be a surface in R^3. Then M is a Bonnet surface if and only if*

$$\left(\frac{Q_{\bar{z}}}{|Q|^2}\right)_z = 0$$

Remark that the domain D is sometimes replaced by an arbitrary Riemannian 2-manifold, in the literature.

Many papers assume that a Bonnet surface is of non-constant mean curvature and has no umbilic points. However, constant mean curvature surfaces represent a very particular case of Bonnet surfaces, which is easy to see from Chen's Theorem above: the forementioned PDE is satisfied by constant mean curvature surfaces, since one characteristic property of CMC surfaces is that the Hopf differential coefficient Q is a holomorphic function ($Q_{\bar{z}} = 0$).

A 1-parameter family of Bonnet surfaces is classically well-studied [13], but not from a constructional view point. Our constructional approach involves solving a Bonnet-type problem for the following invariants: given conformal metric, and mean curvature.

Construction algorithm for a Bonnet 1-family of surfaces on an open, simply connected domain D.

a). Choose a smooth Bonnet surface of Riemannian metric $E(du^2 + dv^2)$ and differentiable mean curvature function $H(u, v)$. Compute K determined by the metric, from the Gauss equation. H and K determine k_1 and k_2, the principal curvatures, at every point.

b). Choose an appropriate smooth function $n(u, v)$ within the range

$$[min(k_1E, k_2E), max(k_1E, k_2E)] \in \mathbb{R}$$

. (This condition is imposed by equation 4.27).

c). Based on equation $l = 2HE - n$, compute the function $l(u, v)$.

d). Compute $m(u, v)$ from equation 4.27, taking into consideration both possible solutions.

A straightforward computation shows that the obtained functions E, l, m, n verify the Gauss-Codazzi-Mainardi equations.

e). Plug the functions $E = G, l, m, n$ into the Lax matrices from 4.17, and solve the Lax system numerically, with appropriate initial conditions.

f). Use the solution obtained above (moving frame) in order to obtain the explicit immersion formula, either using Sym's formula for associated surfaces, or with direct numerical integration using Picard's theorem.

Remark 1 Observe that for the case of isothermic coordinates (for examples, for the case of CMC surfaces), one can obtain the functions l, n, m at the same time, by solving the Gauss-Codazzi-Mainardi equations simultaneously, as they are easy to solve compared to the general ones:

$$ln = \frac{-E_{vv} - E_{uu}}{2} + \frac{(E_u)^2 + (E_v)^2}{2E} \tag{4.31}$$

$$l_v = E_v \cdot H \tag{4.32}$$

$$n_u = E_u \cdot H. \tag{4.33}$$

In particular for CMC surfsces (H non-zero constant), one immediately obtains $l = HE + \alpha(u)$ and $n = HE + \beta(v)$, and the rest of the conditions subsequently follow.

4.3 Visual Examples

We are concluding this report with a few numerical/visual examples of such families, namely, families of associate constant mean curvature surfaces (which are a particular and very important case of Bonnet surfaces).

Figure 4.1: Associated Family of Cylinder

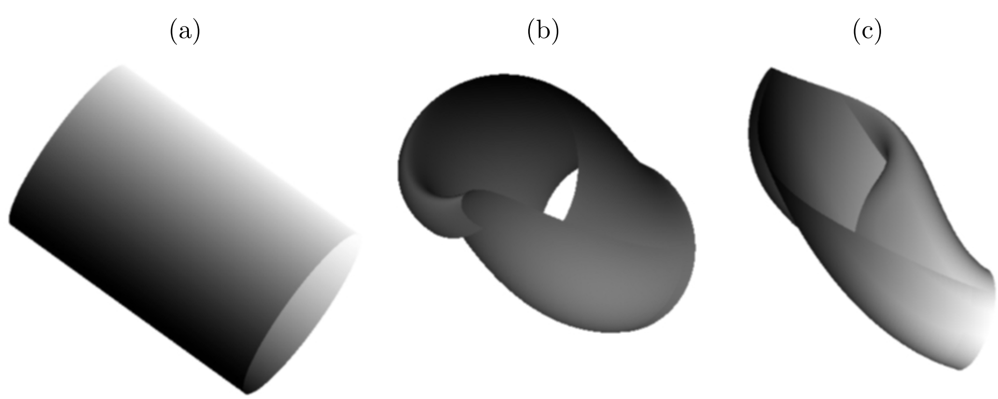
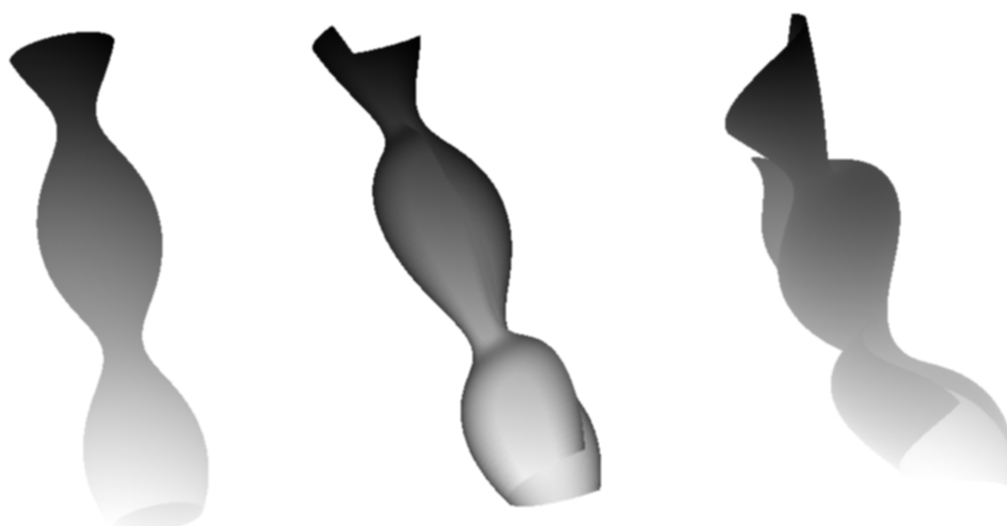


Figure 4.2: Associated Family of Unduloid

(a) (b) (c)



CHAPTER 5
CONSTRUCTING ISOTHERMIC COORDINATES (CURVATURE LINE ISOTHERMAL CHARTS)

As stated in the previous sections, trajectories of the principal curvature directions, where they exist, constitute the network of curvature lines of a surface, and a parametrization $(\gamma, \beta) \mapsto f(\gamma, \beta)$ along curvature lines is called a **curvature line parametrization** [17]

Principal curvatures at each point represent the minimum, respectively, maximum values of the normal curvature; they also represent the eigenvalues of the shape operator, as a linear operator on the tangent plane. Clearly, if the principal curvatures are distinct, then the eigenvectors (called principal directions) are orthogonal. At umbilic points, where the normal curvature is the same in every direction, the actual choice of an orthogonal system of eigenvectors becomes problematic. Therefore, we usually view all the umbilics as singularities. Clearly, this point of view makes the sphere a very special example of isothermic surface.

Assume that we are dealing with a regular surfaces in $\mathbb{R}^3$ which admit isothermic coordinates (away from a discrete set of singularities). We will call such a surface an **isothermic surface**.

Isothermic surfaces were extensively studied by the paper [18], where the main interest was to study Bonnet surface as a particular case of isothermic surfaces.

It is important to note that [18] mentioned the following important characteristic property of isothermic surfaces:

M is an isothermic surface in $\mathbb{R}^3$ if and only if locally there exists a conformal parameter $z = x + iy$ such that the Hopf differential coefficient, $Q = Q(z, \bar{z})$ is a real-valued function.

This characteristic property (in terms of Q being real)is obvious from the definition of an isothermic surface, together with the formula of Q in terms of l, m and n.

[18] presented a few reformulations of this important condition.

We arrived at the following question, which was important to answer in this work, from both a theoretical and a constructive (applicative) point of view:

Q: "Starting from a arbitrarily given immersion $(x, y) \mapsto f(x, y)$ of an isothermic surface in the Euclidean 3-space, is there a simple method of obtaining an isothermic parameterization $(\gamma, \beta) \mapsto f(\gamma, \beta)$ corresponding to it?"'

We implemented a key idea for changing the old coordinates to the isothermic ones, that can be summarized in two steps:

a). applying the Gram-Schmidt orthogonalization;

and

b). multiplying the velocity vectors by a 'scaling function' K, namely a class C^2 non-constant function specific to this chapter (not to be confused with the Gaussian curvature).

Corresponding to this method, we obtained a condition that this function, K, must satisfy, and we constructed isothermic coordinates from an arbitrary immersion, under the above-stated condition.

Solving a Bonnet problem gives rise to a generic immersion, which is rarely isothermic; constructing an isothermic parameterization for the same physical surface, whenever possible, presents a great deal of simplification. Such an isothermic parameterization creates a special mesh that is desirable in some applications, such as:

- applying geometric methods to mesh discretization which are both homogeneous and orthogonal;
- in structural mechanics, deriving the equation of shells along the principal directions;
- in architecture/industrial design, working with orthogonal meshes is often preferrable, in order to minimize the production cost;
- finding dual surfaces (Christoffel transforms).

Theorem 5.0.1. *Let $f(x,y) = (f^1(x,y), f^2(x,y), f^3(x,y))$ be a generic parametrization with velocities f_x and f_y, for a regular surface $M = f(D)$ that is known to admit an isothermic parameterization everywhere, where D represents an open and simply connected domain. Let us define the vector distributions:*

$$f_\gamma := K\left(\frac{f_x}{\sqrt{E}}cos(\alpha) + \frac{f_y - \frac{F}{E}f_x}{\sqrt{G - \frac{F^2}{E}}}sin(\alpha)\right) \tag{5.1}$$

$$f_\beta := K\left(-\frac{f_x}{\sqrt{E}}sin(\alpha) + \frac{f_y - \frac{F}{E}f_x}{\sqrt{G - \frac{F^2}{E}}}cos(\alpha)\right)$$

where the angle α satisfies the following equation (in terms of coefficients of the first and second fundamental forms corresponding to f):

$$tan(2\alpha) = -2\left(\frac{\frac{-\frac{F}{E}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}l + \frac{1}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}m}{\frac{2\frac{F^2}{E}-G}{E(G-\frac{F^2}{E})}l - 2\frac{\frac{F}{E}}{G-\frac{F^2}{E}}m + \frac{1}{G-\frac{F^2}{E}}n}\right) \tag{5.2}$$

Then, the named vectors f_γ and f_β represent the velocities on an isothermic parametrization in terms of (γ, β) **if** *there exists a smooth function K which satisfies the following equations:*

$$K_\gamma = K^2\left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G - \frac{F^2}{E})\sqrt{E}}cos(\alpha) - \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G - \frac{F^2}{E}}}sin(\alpha)\right)$$
$$+K^2\left(-\frac{\alpha_x}{\sqrt{E}}sin(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}cos(\alpha)\right)$$
$$K_\beta = K^2\left(-\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G - \frac{F^2}{E})\sqrt{E}}sin(\alpha) - \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G - \frac{F^2}{E}}}cos(\alpha)\right)$$
$$+K^2\left(-\frac{\alpha_x}{\sqrt{E}}cos(\alpha) - \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}sin(\alpha)\right) \tag{5.3}$$

Remark from the fractional expression of $\tan(2\alpha)$ that both the numerator and the denominator would become zero at umbilic points. For umbilics, the expression would be undetermined, and not undefined: at such points, frame choices are still possible. However, remark that the hypothesis of regular isothermic surface, as stated, leaves aside any types of singularities, including umbilics.

The existence of such a function K represents just a sufficient (not necessary and sufficient) condition for the reparameterization of $f(x, y)$ in terms of isothermic coordinates (β, γ). This represents just a simple method to obtain this reparameterization. We are not claiming that this is the only method.

In this theorem, equations (5.1) and (5.3) assure the compatibility condition $(f_\gamma)_\beta = (f_\beta)_\gamma$, which must be satisfied in order for the isothermic immersion to exist.

Proof. By the definition of **isothermic**, if the vectors f_γ and f_β represent the velocities on the isothermic parametrization (γ, β) then, they need to satisfy the following conditions:

1. $||f_\gamma|| = ||f_\beta||$,
2. $< f_\gamma, f_\beta >= 0$,
3. $< f_{\gamma\beta}, N >= 0$.
4. $(f_\gamma)_\beta = (f_\beta)_\gamma$

1. It is easy to see that f_γ and f_β satisfy the first condition. Specifically, we obtain:

$$||f_\gamma||^2 =< f_\gamma, f_\gamma >= K^2 \left(\frac{||f_x||^2}{E} cos^2(\alpha) + 2 \frac{< f_x, f_y > - \frac{F}{E}||f_x||^2}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} cos(\alpha) sin(\alpha) \right)$$
$$+K^2 \left(\frac{||f_y||^2 + \frac{F^2}{E^2}||f_x||^2}{G - \frac{F^2}{E}} sin^2(\alpha) \right)$$
$$= K^2$$

and

$$||f_\beta||^2 =< f_\beta, f_\beta >= K^2 \left(\frac{||f_x||^2}{E} sin^2(\alpha) - 2 \frac{< f_x, f_y > - \frac{F}{E}||f_x||^2}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} cos(\alpha) sin(\alpha) \right)$$
$$+K^2 \left(\frac{||f_y||^2 + \frac{F^2}{E^2}||f_x||^2}{G - \frac{F^2}{E}} cos^2(\alpha) \right)$$
$$= K^2.$$

It is now proved that $||f_\gamma|| = ||f_\beta|| = K$.

2. The second condition is also straight-forward to verify:

$$< f_\gamma, f_\beta >= K^2 \left(-\frac{||f_x||^2}{E} cos(\alpha) sin(\alpha) + \frac{< f_x, f_y > - \frac{F}{E}||f_x||^2}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} cos^2(\alpha) \right)$$
$$+K^2 \left(-\frac{< f_x, f_y > - \frac{F}{E}||f_x||^2}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} sin^2(\alpha) \right)$$
$$+K^2 \left(\frac{||f_y||^2 - 2\frac{F}{E} < f_y, f_x > + \frac{F^2}{E^2}||f_x||^2}{G - \frac{F^2}{E}} sin(\alpha) cos(\alpha) \right)$$
$$= 0.$$

3. And, to prove the third condition let $D_1(x, y) = \frac{f_x}{\sqrt{E}}$ and $D_2(x, y) = \frac{f_y - \frac{F}{E} f_x}{\sqrt{G - \frac{F^2}{E}}}$ to make computations easy to understand, so f_γ, f_β will be

$$f_\gamma := K \left(D_1 cos(\alpha) + D_2 sin(\alpha) \right)$$
$$f_\beta := K \left(-D_1 sin(\alpha) + D_2 cos(\alpha) \right), \tag{5.4}$$

and

$$< (f_\gamma)_\beta, N >=< K_\beta (D_1 cos(\alpha) + D_2 sin(\alpha)) + K^2 (- \frac{\frac{\partial}{\partial x} D_1}{\sqrt{E}} sin(\alpha) cos(\alpha)$$
$$- \frac{\frac{\partial}{\partial x} D_2}{\sqrt{E}} sin^2(\alpha) + \frac{\frac{\partial}{\partial y} D_1}{\sqrt{G - \frac{F^2}{E}}} cos^2(\alpha) + \frac{\frac{\partial}{\partial y} D_2}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha) cos(\alpha)$$
$$- \frac{F}{E} \frac{\frac{\partial}{\partial x} D_1}{\sqrt{G - \frac{F^2}{E}}} cos^2(\alpha) + \frac{D_1 \alpha_x}{\sqrt{E}} sin^2(\alpha) - \frac{F}{E} \frac{\frac{\partial}{\partial x} D_2}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha) cos(\alpha)$$

$$-\frac{D_2\alpha_x}{\sqrt{E}}sin(\alpha)cos(\alpha)-\frac{D_1\alpha_y}{\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)+\frac{D_2\alpha_y}{\sqrt{G-\frac{F^2}{E}}}cos^2(\alpha)$$
$$+\frac{F}{E}\frac{D_1\alpha_x}{\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)-\frac{F}{E}\frac{D_2\alpha_x}{\sqrt{G-\frac{F^2}{E}}}cos^2(\alpha)), N> \quad (5.5)$$

Since $<f_x, N>=0$ and $<f_y, N>=0$, then all terms with $<D_1, N>$ and $<D_2, N>$ will be zero; replacing the partial dervatives of D_1 and D_2 into the equation (5.5) we obtain:

$$<(f_\gamma)_\beta, N>=\left(\frac{-\frac{F}{E}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}(cos^2(\alpha)-sin^2(\alpha))\right)<f_{xx}, N>$$
$$+\left(\frac{2\frac{F^2}{E}-G}{E\left(G-\frac{F^2}{E}\right)}sin(\alpha)cos(\alpha)\right)<f_{xx}, N>$$
$$+\left(\frac{1}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}(cos^2(\alpha)-sin^2(\alpha))-2\frac{\frac{F}{E}}{G-\frac{F^2}{E}}sin(\alpha)cos(\alpha)\right)<f_{xy}, N>$$
$$+\left(\frac{1}{G-\frac{F^2}{E}}cos(\alpha)sin(\alpha)\right)<f_{yy}, N>$$
$$=\left(\frac{-\frac{F}{E}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}l+\frac{1}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}m\right)cos(2\alpha)$$
$$+\left(\frac{2\frac{F^2}{E}-G}{E\left(G-\frac{F^2}{E}\right)}l-2\frac{\frac{F}{E}}{G-\frac{F^2}{E}}m+\frac{1}{G-\frac{F^2}{E}}n\right)\frac{sin(2\alpha)}{2} \quad (5.6)$$

Now, if we rewrite the equation (5.2):

$$sin(2\alpha)=-2\left(\frac{\frac{-\frac{F}{E}}{\sqrt{(G-\frac{F^2}{E})}\sqrt{E}}l+\frac{1}{\sqrt{\left(G-\frac{F^2}{E}\right)}\sqrt{E}}m}{\frac{2\frac{F^2}{E}-G}{E\left(G-\frac{F^2}{E}\right)}l-2\frac{\frac{F}{E}}{G-\frac{F^2}{E}}m+\frac{1}{G-\frac{F^2}{E}}n}\right)cos(2\alpha) \quad (5.7)$$

and plug it into the equation (5.6), then we obtain:

$$< (f_\gamma)_\beta, N >= \left(\frac{-\frac{F}{E}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}} l + \frac{1}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}} m \right) cos(2\alpha)$$
$$- \left(\frac{2\frac{F^2}{E} - G}{E\left(G-\frac{F^2}{E}\right)} l - 2\frac{\frac{F}{E}}{G-\frac{F^2}{E}} m + \frac{1}{G-\frac{F^2}{E}} n \right)$$
$$\left(\frac{\frac{-\frac{F}{E}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}} l + \frac{1}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}} m}{\frac{2\frac{F^2}{E} - G}{E\left(G-\frac{F^2}{E}\right)} l - 2\frac{\frac{F}{E}}{G-\frac{F^2}{E}} m + \frac{1}{G-\frac{F^2}{E}} n} \right) cos(2\alpha) = 0.$$

4. Last, we prove the compatibility condition. Again, consider the equation (5.4):

$$(f_\gamma)_\beta = K_\beta \left(D_1 cos(\alpha) + D_2 sin(\alpha)\right) + K \frac{\partial}{\partial \beta} \left(D_1 cos(\alpha) + D_2 sin(\alpha)\right)$$
$$= K_\beta (D_1 cos(\alpha) + D_2 sin(\alpha)) + K(-\frac{\frac{\partial}{\partial x}(D_1 cos(\alpha) + D_2 sin(\alpha))}{\sqrt{E}} sin(\alpha)$$
$$+ \frac{\frac{\partial}{\partial y}(D_1 cos(\alpha) + D_2 sin(\alpha)) - \frac{F}{E}\frac{\partial}{\partial x}(D_1 cos(\alpha) + D_2 sin(\alpha))}{\sqrt{G-\frac{F^2}{E}}} cos(\alpha))$$
$$= K_\beta (D_1 cos(\alpha) + D_2 sin(\alpha)) + K^2(-\frac{\frac{\partial}{\partial x} D_1}{\sqrt{E}} sin(\alpha)cos(\alpha) - \frac{\frac{\partial}{\partial x} D_2}{\sqrt{E}} sin^2(\alpha)$$
$$+ \frac{\frac{\partial}{\partial y} D_1}{\sqrt{G-\frac{F^2}{E}}} cos^2(\alpha) + \frac{\frac{\partial}{\partial y} D_2}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) - \frac{F}{E}\frac{\frac{\partial}{\partial x} D_1}{\sqrt{G-\frac{F^2}{E}}} cos^2(\alpha)$$
$$+ \frac{D_1 \alpha_x}{\sqrt{E}} sin^2(\alpha) - \frac{F}{E}\frac{\frac{\partial}{\partial x} D_2}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) - \frac{D_2 \alpha_x}{\sqrt{E}} sin(\alpha)cos(\alpha)$$
$$- \frac{D_1 \alpha_y}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) + \frac{D_2 \alpha_y}{\sqrt{G-\frac{F^2}{E}}} cos^2(\alpha)$$
$$+ \frac{F}{E}\frac{D_1 \alpha_x}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) - \frac{F}{E}\frac{D_2 \alpha_x}{\sqrt{G-\frac{F^2}{E}}} cos^2(\alpha))$$

$$(f_\beta)_\gamma = K_\beta(-D_1 sin(\alpha) + D_2 cos(\alpha)) + K\frac{\partial}{\partial\beta}(-D_1 sin(\alpha) + D_2 cos(\alpha))$$
$$= K_\beta(-D_1 sin(\alpha) + D_2 cos(\alpha)) + K(\frac{\frac{\partial}{\partial x}(-D_1 sin(\alpha) + D_2 cos(\alpha))}{\sqrt{E}} cos(\alpha)$$
$$+\frac{\frac{\partial}{\partial y}(-D_1 sin(\alpha) + D_2 cos(\alpha)) - \frac{F}{E}\frac{\partial}{\partial x}(-D_1 sin(\alpha) + D_2 cos(\alpha))}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha))$$

$$= K_\gamma(-D_1 sin(\alpha) + D_2 cos(\alpha)) + K^2(-\frac{\frac{\partial}{\partial x}D_1}{\sqrt{E}} sin(\alpha)cos(\alpha) + \frac{\frac{\partial}{\partial x}D_2}{\sqrt{E}} cos^2(\alpha)$$
$$-\frac{\frac{\partial}{\partial y}D_1}{\sqrt{G-\frac{F^2}{E}}} sin^2(\alpha) + \frac{\frac{\partial}{\partial y}D_2}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) + \frac{F}{E}\frac{\frac{\partial}{\partial x}D_1}{\sqrt{G-\frac{F^2}{E}}} sin^2(\alpha)$$
$$+\frac{D_1\alpha_x}{\sqrt{E}} sin^2(\alpha) - \frac{F}{E}\frac{\frac{\partial}{\partial x}D_2}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) - \frac{D_2\alpha_x}{\sqrt{E}} sin(\alpha)cos(\alpha)$$
$$-\frac{D_1\alpha_y}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) - \frac{D_2\alpha_y}{\sqrt{G-\frac{F^2}{E}}} sin^2(\alpha)$$
$$+\frac{F}{E}\frac{D_1\alpha_x}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) + \frac{F}{E}\frac{D_2\alpha_x}{\sqrt{G-\frac{F^2}{E}}} sin^2(\alpha))$$

Therefore:

$$(f_\gamma)_\beta - (f_\beta)_\gamma = K_\beta(D_1 cos(\alpha) + D_2 sin(\alpha)) - K_\gamma(-D_1 sin(\alpha) + D_2 cos(\alpha))$$
$$+K^2(-\frac{\frac{\partial}{\partial x}D_2}{\sqrt{E}} + \frac{\frac{\partial}{\partial y}D_1}{\sqrt{G-\frac{F^2}{E}}} - \frac{F}{E}\frac{\frac{\partial}{\partial x}D_1}{\sqrt{G-\frac{F^2}{E}}} + \frac{D_1\alpha_x}{\sqrt{E}} + \frac{D_2\alpha_y}{\sqrt{G-\frac{F^2}{E}}} - \frac{F}{E}\frac{D_2\alpha_x}{\sqrt{G-\frac{F^2}{E}}})$$
$$= K_\beta(D_1 cos(\alpha) + D_2 sin(\alpha)) - K_\gamma(-D_1 sin(\alpha) + D_2 cos(\alpha))$$
$$+K^2(-\frac{f_{yx} - \frac{F}{E}f_{xx} - \frac{\partial}{\partial x}(\frac{F}{E})f_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}} + \frac{(f_y - \frac{F}{E}f_x)\frac{\partial}{\partial x}(\sqrt{G-\frac{F^2}{E}})}{(G-\frac{F^2}{E})\sqrt{E}}$$
$$+\frac{f_{xy}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}} - \frac{\frac{\partial}{\partial y}(\sqrt{E})f_x}{\sqrt{G-\frac{F^2}{E}}E} - \frac{F}{E}\frac{f_{xx}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}$$
$$+\frac{D_1\alpha_x}{\sqrt{E}} + \frac{D_2\alpha_y}{\sqrt{G-\frac{F^2}{E}}} - \frac{F}{E}\frac{D_2\alpha_x}{\sqrt{G-\frac{F^2}{E}}}) \tag{5.8}$$

After some simplified terms, we factorize everything with D_1 and D_2 as

$$
\begin{aligned}
&(f_\gamma)_\beta - (f_\beta)_\gamma = (K_\beta cos(\alpha) + K_\gamma sin(\alpha))\, D_1 \\
&+K^2 \left(\frac{\frac{\partial}{\partial x}(\frac{F}{E})}{\sqrt{G - \frac{F^2}{E}}} - \frac{\frac{\partial}{\partial y}(\sqrt{E})}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} + \frac{F}{E}\frac{\frac{\partial}{\partial x}(\sqrt{E})}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}} \right) D_1 \\
&+ \left(K_\beta sin(\alpha) - K_\gamma cos(\alpha) + K^2 \left(\frac{\frac{\partial}{\partial x}(\sqrt{G - \frac{F^2}{E}})}{\sqrt{G - \frac{F^2}{E}}\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} \right) \right) D_2 \\
&= \left(K_\beta cos(\alpha) + K_\gamma sin(\alpha) + K^2 \left(\frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G - \frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}} \right) \right) D_1 \\
&+ \left(K_\beta sin(\alpha) - K_\gamma cos(\alpha) + K^2 \left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G - \frac{F^2}{E})\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} \right) \right) D_2
\end{aligned} \tag{5.9}
$$

Next,

$$K_\beta cos(\alpha) + K_\gamma sin(\alpha) = K^2 \left(-\frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G - \frac{F^2}{E}}} - \frac{\alpha_x}{\sqrt{E}} \right) \tag{5.10}$$

$$K_\beta sin(\alpha) - K_\gamma cos(\alpha) = K^2 \left(-\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G - \frac{F^2}{E})\sqrt{E}} - \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} \right) \tag{5.11}$$

Finally, if we replace (5.10) and (5.11) into the equation (5.9), then we obtain:

$$(f_\gamma)_\beta - (f_\beta)_\gamma = 0 \tag{5.12}$$

All the conditions are satisfied, in order for the isothermic immersion $f(\beta, \gamma)$ to exist.

□

In the above theorem, we proved that, the existence of a certain scaling function $K(x, y)$ of arbitrarily given coordinates guarantees to provide some new, isothermic coordinates. K represents the solution of a system of differential equations, (5.13). Moreover, we can provide a condition for such a scaling function K to exist and be unique:

Theorem 5.0.2. *As in the previous theorem, let E,F,G and l,m,n represent the coefficients of first and second fundamental forms of the immersion $f(x,y)$, respectively, and*

$$\begin{aligned}
K_\gamma &= K^2\left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}}cos(\alpha) - \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}}sin(\alpha)\right)\\
&+K^2\left(-\frac{\alpha_x}{\sqrt{E}}sin(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}cos(\alpha)\right)\\
K_\beta &= K^2\left(-\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}}sin(\alpha) - \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}}cos(\alpha)\right)\\
&+K^2\left(-\frac{\alpha_x}{\sqrt{E}}cos(\alpha) - \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}sin(\alpha)\right) \qquad (5.13)
\end{aligned}$$

be a system of differential equations, where

$$tan(2\alpha) = -2\left(\frac{\frac{-\frac{F}{E}}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}l + \frac{1}{\sqrt{G-\frac{F^2}{E}}\sqrt{E}}m}{\frac{2\frac{F^2}{E}-G}{E(G-\frac{F^2}{E})}l - 2\frac{\frac{F}{E}}{G-\frac{F^2}{E}}m + \frac{1}{G-\frac{F^2}{E}}n}\right). \qquad (5.14)$$

The following condition:

$$\begin{aligned}
&\left(\frac{\frac{\partial}{\partial y}\left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}}\right) - \frac{F}{E}\frac{\partial}{\partial x}\left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}}\right)}{\sqrt{G-\frac{F^2}{E}}} + \frac{\frac{\partial}{\partial x}\left(\frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}}\right)}{\sqrt{E}}\right)\\
&+\left(\left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}}\right)\left(\frac{\alpha_x}{\sqrt{E}}\right) - \left(\frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}}\right)\left(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}\right)\right) \qquad (5.15)\\
&-\left(\frac{\frac{\partial}{\partial x}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{E}} + \left(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}\right)\frac{\alpha_x}{\sqrt{E}} + \frac{\frac{\partial}{\partial y}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}})}{\sqrt{G-\frac{F^2}{E}}} - \frac{F}{E}\frac{(\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}) + (\frac{\alpha_x}{\sqrt{E}})\alpha_x)}{\sqrt{G-\frac{F^2}{E}}}\right) = 0
\end{aligned}$$

assures the existence of a function $K(x,y)$ that satisfies the system of differential *equations. Moreover, if the solution K exists, then it will be unique, given some appropriate initial value condition.*

Proof.

$$K_\gamma = -K^2 \left(\frac{\frac{\partial}{\partial x}(\frac{F}{E})}{\sqrt{G - \frac{F^2}{E}}} - \frac{\frac{\partial}{\partial y}(\sqrt{E})}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} + \frac{F}{E}\frac{\frac{\partial}{\partial x}(\sqrt{E})}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}} \right) sin(\alpha)$$
$$+K^2 \left(\frac{\frac{\partial}{\partial x}(\sqrt{G - \frac{F^2}{E}})}{\sqrt{G - \frac{F^2}{E}}\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} \right) cos(\alpha) \tag{5.16}$$

$$K_\beta = -K^2 \left(\frac{\frac{\partial}{\partial x}(\frac{F}{E})}{\sqrt{G - \frac{F^2}{E}}} - \frac{\frac{\partial}{\partial y}(\sqrt{E})}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} + \frac{F}{E}\frac{\frac{\partial}{\partial x}(\sqrt{E})}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}} \right) cos(\alpha)$$
$$-K^2 \left(\frac{\frac{\partial}{\partial x}(\sqrt{G - \frac{F^2}{E}})}{\sqrt{G - \frac{F^2}{E}}\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} \right) sin(\alpha). \tag{5.17}$$

After some simplifications, we will have:

$$K_\gamma = K^2 \left(\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G - \frac{F^2}{E})\sqrt{E}} cos(\alpha) - \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G - \frac{F^2}{E}}} sin(\alpha) \right) \tag{5.18}$$
$$+K^2 \left(-\frac{\alpha_x}{\sqrt{E}} sin(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} cos(\alpha) \right)$$
$$K_\beta = K^2 \left(-\frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G - \frac{F^2}{E})\sqrt{E}} sin(\alpha) - \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G - \frac{F^2}{E}}} cos(\alpha) \right)$$
$$+K^2 \left(-\frac{\alpha_x}{\sqrt{E}} cos(\alpha) - \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha) \right).$$

Let $P_1(x,y) = \frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}$ and $P_2(x,y) = \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}}$ to make computations easy to understand:

$$K_\gamma = K^2\left(P_1 cos(\alpha) - P_2 sin(\alpha)\right) \tag{5.19}$$

$$K_\beta = K^2\left(-P_1 sin(\alpha) - P_2 cos(\alpha)\right). \tag{5.20}$$

Further, we are writing the mixed partial derivatives of K as given by the system of equations:

$$\begin{aligned}
K_{\gamma\beta} &= 2KK_\beta\left(P_1 cos(\alpha) - P_2 sin(\alpha)\right)\\
&+K^3\left(\left(-\frac{\frac{\partial}{\partial x}P_1}{\sqrt{E}} - \frac{\frac{\partial}{\partial y}P_2 - \frac{F}{E}\frac{\partial}{\partial x}P_2}{\sqrt{G-\frac{F^2}{E}}}\right)sin(\alpha)cos(\alpha) + \frac{\frac{\partial}{\partial y}P_1 - \frac{F}{E}\frac{\partial}{\partial x}P_1}{\sqrt{G-\frac{F^2}{E}}}cos^2(\alpha)\right)\\
&+K^3\left(\frac{\frac{\partial}{\partial x}P_2}{\sqrt{E}}sin^2(\alpha)\right) + W_1(\alpha) \qquad (5.21)\\
&= 2KK_\beta\frac{K_\gamma}{K} + K^3\left(-\frac{\frac{\partial}{\partial x}P_1}{\sqrt{E}} - \frac{\frac{\partial}{\partial y}P_2 - \frac{F}{E}\frac{\partial}{\partial x}P_2}{\sqrt{G-\frac{F^2}{E}}}\right)sin(\alpha)cos(\alpha)\\
&+K^3\left(\frac{\frac{\partial}{\partial y}P_1 - \frac{F}{E}\frac{\partial}{\partial x}P_1}{\sqrt{G-\frac{F^2}{E}}}cos^2(\alpha) + \frac{\frac{\partial}{\partial x}P_2}{\sqrt{E}}sin^2(\alpha)\right) + W_1(\alpha)\\
K_{\beta\gamma} &= 2KK_\gamma\left(-P_1 sin(\alpha) - P_2 cos(\alpha)\right) + K^3\left(-\frac{\frac{\partial}{\partial x}P_1}{\sqrt{E}} - \frac{\frac{\partial}{\partial y}P_2 - \frac{F}{E}\frac{\partial}{\partial x}P_2}{\sqrt{G-\frac{F^2}{E}}}\right)sin(\alpha)cos(\alpha)\\
&+K^3\left(-\frac{\frac{\partial}{\partial y}P_1 - \frac{F}{E}\frac{\partial}{\partial x}P_1}{\sqrt{G-\frac{F^2}{E}}}sin^2(\alpha) - \frac{\frac{\partial}{\partial x}P_2}{\sqrt{E}}cos^2(\alpha)\right) + W_2(\alpha)\\
&= 2KK_\gamma\frac{K_\beta}{K} + K^3\left(-\frac{\frac{\partial}{\partial x}P_1}{\sqrt{E}} - \frac{\frac{\partial}{\partial y}P_2 - \frac{F}{E}\frac{\partial}{\partial x}P_2}{\sqrt{G-\frac{F^2}{E}}}\right)sin(\alpha)cos(\alpha)\\
&+K^3\left(-\frac{\frac{\partial}{\partial y}P_1 - \frac{F}{E}\frac{\partial}{\partial x}P_1}{\sqrt{G-\frac{F^2}{E}}}sin^2(\alpha) - \frac{\frac{\partial}{\partial x}P_2}{\sqrt{E}}cos^2(\alpha)\right) + W_2(\alpha) \qquad (5.22)\\
K_{\gamma\beta} - K_{\beta\gamma} &= K^3\left(\frac{\frac{\partial}{\partial y}P_1 - \frac{F}{E}\frac{\partial}{\partial x}P_1}{\sqrt{G-\frac{F^2}{E}}} + \frac{\frac{\partial}{\partial x}P_2}{\sqrt{E}}\right) + W_1(\alpha) - W_2(\alpha) = 0.
\end{aligned}$$

In the above computations, we introduced $W_1(\alpha)$ and $W_2(\alpha)$ for simplification. Now, we need to evaluate the expression $W_1(\alpha) - W_2(\alpha)$:

$$W_1(\alpha) - W_2(\alpha) = K^3 \left(-P_1 \left(-\frac{\alpha_x}{\sqrt{E}} sin^2(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha)cos(\alpha) \right) \right)$$
$$-K^3 \left(P_2 \left(-\frac{\alpha_x}{\sqrt{E}} sin(\alpha)cos(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} cos^2(\alpha) \right) \right)$$
$$+K^3 \left(P_1 \left(\frac{\alpha_x}{\sqrt{E}} cos^2(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha)cos(\alpha) \right) \right)$$
$$-K^3 \left(P_2 \left(\frac{\alpha_x}{\sqrt{E}} sin(\alpha)cos(\alpha) + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}} sin^2(\alpha) \right) \right)$$
$$-K^3 \left(\frac{\frac{\partial}{\partial x}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{E}} cos^2(\alpha) + \frac{\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}})}{\sqrt{E}} sin(\alpha)cos(\alpha) - \frac{\alpha_x^2}{E} sin(\alpha)cos(\alpha) \right)$$
$$-K^3 \left(\frac{\frac{\partial}{\partial y}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha)cos(\alpha) + \frac{\frac{\partial}{\partial y}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}})}{\sqrt{G - \frac{F^2}{E}}} sin^2(\alpha) - \frac{\alpha_x^2}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} sin(\alpha)cos(\alpha) \right)$$
$$-K^3 \left(-\frac{F}{E} \frac{(\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}) + (\frac{\alpha_x}{\sqrt{E}})\alpha_x)}{\sqrt{G - \frac{F^2}{E}}} sin^2(\alpha) + (\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}) \frac{\alpha_x}{\sqrt{E}} cos^2(\alpha) \right)$$
$$-K^3 \left(\frac{\frac{\partial}{\partial x}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{E}} sin^2(\alpha) - \frac{\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}})}{\sqrt{E}} sin(\alpha)cos(\alpha) + \frac{\alpha_x^2}{E} sin(\alpha)cos(\alpha) \right)$$
$$-K^3 \left(-\frac{\frac{\partial}{\partial y}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{G - \frac{F^2}{E}}} sin(\alpha)cos(\alpha) + \frac{\frac{\partial}{\partial y}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}})}{\sqrt{G - \frac{F^2}{E}}} cos^2(\alpha) + \frac{\alpha_x^2}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} sin(\alpha)cos(\alpha) \right)$$
$$-K^3 \left(-\frac{F}{E} \frac{(\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}) + (\frac{\alpha_x}{\sqrt{E}})\alpha_x)}{\sqrt{G - \frac{F^2}{E}}} cos^2(\alpha) + (\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}) \frac{\alpha_x}{\sqrt{E}} sin^2(\alpha) \right)$$

$$W_1(\alpha) - W_2(\alpha) = K^3 \left(P_1(\frac{\alpha_x}{\sqrt{E}}) - P_2(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}) \right)$$
$$-K^3 \left(\frac{\frac{\partial}{\partial x}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{E}} + (\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}})\frac{\alpha_x}{\sqrt{E}} + \frac{\frac{\partial}{\partial y}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}})}{\sqrt{G - \frac{F^2}{E}}} \right)$$
$$-K^3 \left(-\frac{F}{E}\frac{(\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}) + (\frac{\alpha_x}{\sqrt{E}})\alpha_x)}{\sqrt{G - \frac{F^2}{E}}} \right).$$

Remark that the compatibility condition of the system of equations writes $K_{\gamma\beta} - K_{\beta\gamma}$, which is equivalent to the condition provided in the hypothesis.

$$K_{\gamma\beta} - K_{\beta\gamma} = K^3 \left(\frac{\frac{\partial}{\partial y}P_1 - \frac{F}{E}\frac{\partial}{\partial x}P_1}{\sqrt{G - \frac{F^2}{E}}} + \frac{\frac{\partial}{\partial x}P_2}{\sqrt{E}} + P_1(\frac{\alpha_x}{\sqrt{E}}) - P_2(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}) \right)$$
$$-K^3 \left(\frac{\frac{\partial}{\partial x}(\frac{\alpha_x}{\sqrt{E}})}{\sqrt{E}} + (\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}})\frac{\alpha_x}{\sqrt{E}} + \frac{\frac{\partial}{\partial y}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}})}{\sqrt{G - \frac{F^2}{E}}} - \frac{F}{E}\frac{(\frac{\partial}{\partial x}(\frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G - \frac{F^2}{E}}}) + (\frac{\alpha_x}{\sqrt{E}})\alpha_x)}{\sqrt{G - \frac{F^2}{E}}} \right) = 0. \quad (5.23)$$

□

Theorem 5.0.3. *Let*

$$x_\gamma = K \left(\frac{cos(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{sin(\alpha)}{\sqrt{G - \frac{F^2}{E}}} \right) \quad (5.24)$$
$$x_\beta = K \left(\frac{-sin(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{cos(\alpha)}{\sqrt{G - \frac{F^2}{E}}} \right)$$

and

$$y_\gamma = K\left(\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right) \tag{5.25}$$
$$y_\beta = K\left(\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)$$

be a system of differential equations. Then, there exist (x, y), solutions to the system of equations (5.24) and (5.25), which are unique up to choice of initial values (x_0, y_0).

Proof. Again, here it is enough to show that the compatibility conditions for each system are satisfied. Let us start with the equation (5.24) and recall

$P_1(x,y) = \frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}$ and $P_2(x,y) = \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}}$.

Then, we obtain the mixed partial derivatives as following

$$x_{\gamma\beta} = K_\beta\left(\frac{cos(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right) + K\frac{\partial}{\partial\beta}\left(\frac{cos(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)$$
$$= K^2\left(-P_1 sin(\alpha) - P_2 cos(\alpha)\right)\left(\frac{cos(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)$$
$$+K^2\left(-\frac{sin(\alpha)\alpha_\beta}{\sqrt{E}} - \frac{cos(\alpha)E_\beta}{2E\sqrt{E}} - \frac{\partial}{\partial\beta}\left(\frac{F}{E}\right)\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}} - \frac{F}{E}\frac{cos(\alpha)\alpha_\beta}{\sqrt{G-\frac{F^2}{E}}}\right)$$
$$+K^2\left(\frac{F}{E}\frac{\frac{\partial}{\partial\beta}\left(\sqrt{G-\frac{F^2}{E}}\right)sin(\alpha)}{G-\frac{F^2}{E}}\right)$$
$$= K^2\left(-\frac{P_1}{\sqrt{E}}sin(\alpha)cos(\alpha) - \frac{P_2}{\sqrt{E}}cos^2(\alpha) + \frac{F}{E}\frac{P_1}{\sqrt{G-\frac{F^2}{E}}}sin^2(\alpha)\right)$$
$$+K^2\left(\frac{\alpha_x}{E}sin^2(\alpha) - \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha) + \frac{E_x}{2E^2}sin(\alpha)cos(\alpha)\right)$$

$$+K^2\left(\frac{F}{E}\frac{P_2}{\sqrt{G-\frac{F^2}{E}}}cos(\alpha)sin(\alpha)-\frac{E_y-\frac{F}{E}E_x}{2E\sqrt{E}\sqrt{G-\frac{F^2}{E}}}cos^2(\alpha)\right)$$

$$+K^2\left(\frac{\partial}{\partial x}\left(\frac{F}{E}\right)\frac{sin^2(\alpha)}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}-\left(\frac{\partial}{\partial y}\left(\frac{F}{E}\right)-\frac{F}{E}\frac{\partial}{\partial x}\left(\frac{F}{E}\right)\right)\frac{sin(\alpha)cos(\alpha)}{G-\frac{F^2}{E}}\right)$$

$$+K^2\left(\frac{F}{E}\frac{\alpha_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)-\frac{F}{E}\frac{\alpha_y-\frac{F}{E}\alpha_x}{G-\frac{F^2}{E}}cos^2(\alpha)\right)$$

$$+K^2\left(-\frac{F}{E}\frac{\frac{\partial}{\partial x}\left(\sqrt{G-\frac{F^2}{E}}\right)}{\sqrt{E}\left(G-\frac{F^2}{E}\right)}sin^2(\alpha)+\frac{F}{E}\frac{\frac{\partial}{\partial y}\left(\sqrt{G-\frac{F^2}{E}}\right)-\frac{F}{E}\frac{\partial}{\partial x}\left(\sqrt{G-\frac{F^2}{E}}\right)}{\sqrt{G-\frac{F^2}{E}}\left(G-\frac{F^2}{E}\right)}sin(\alpha)cos(\alpha)\right)$$

$$x_{\beta\gamma}=K_\gamma\left(\frac{-sin(\alpha)}{\sqrt{E}}-\frac{F}{E}\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)+K\frac{\partial}{\partial\gamma}\left(\frac{-sin(\alpha)}{\sqrt{E}}-\frac{F}{E}\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)$$

$$=K^2\left(P_1cos(\alpha)-P_2sin(\alpha)\right)\left(-\frac{sin(\alpha)}{\sqrt{E}}-\frac{F}{E}\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)$$

$$+K^2\left(-\frac{cos(\alpha)\alpha_\gamma}{\sqrt{E}}+\frac{sin(\alpha)E_\gamma}{2E\sqrt{E}}-\frac{\partial}{\partial\gamma}\left(\frac{F}{E}\right)\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}+\frac{F}{E}\frac{sin(\alpha)\alpha_\gamma}{\sqrt{G-\frac{F^2}{E}}}\right)$$

$$+K^2\left(\frac{F}{E}\frac{\frac{\partial}{\partial\gamma}\left(\sqrt{G-\frac{F^2}{E}}\right)cos(\alpha)}{G-\frac{F^2}{E}}\right)$$

$$=K^2\left(-\frac{P_1}{\sqrt{E}}cos(\alpha)sin(\alpha)+\frac{P_2}{\sqrt{E}}sin^2(\alpha)-\frac{F}{E}\frac{P_1}{\sqrt{G-\frac{F^2}{E}}}cos^2(\alpha)\right)$$

$$+K^2\left(-\frac{\alpha_x}{E}cos^2(\alpha)-\frac{\alpha_y-\frac{F}{E}\alpha_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)+\frac{E_x}{2E^2}sin(\alpha)cos(\alpha)\right)$$

$$+K^2\left(\frac{F}{E}\frac{P_2}{\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)+\frac{E_y-\frac{F}{E}E_x}{2E\sqrt{E}\sqrt{G-\frac{F^2}{E}}}sin^2(\alpha)\right)$$

$$+K^2\left(\frac{F}{E}\frac{\alpha_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)+\frac{F}{E}\frac{\alpha_y-\frac{F}{E}\alpha_x}{G-\frac{F^2}{E}}sin^2(\alpha)\right)$$

$$+K^2\left(+\frac{F}{E}\frac{\frac{\partial}{\partial x}\left(\sqrt{G-\frac{F^2}{E}}\right)}{\sqrt{E}\left(G-\frac{F^2}{E}\right)}cos^2(\alpha)+\frac{F}{E}\frac{\frac{\partial}{\partial y}\left(\sqrt{G-\frac{F^2}{E}}\right)-\frac{F}{E}\frac{\partial}{\partial x}\left(\sqrt{G-\frac{F^2}{E}}\right)}{\sqrt{G-\frac{F^2}{E}}\left(G-\frac{F^2}{E}\right)}sin(\alpha)cos(\alpha)\right).$$

After a heavy computation with repetitive simplifications, we obtain $x_{\gamma\beta}-x_{\beta\gamma}=0$ which implies the compatibility condition $x_{\gamma\beta}=x_{\beta\gamma}$ for x.

Next, we study the compatibility condition for y. Consider the equation (5.25) and mixed partial derivatives will be the following:

$$y_{\gamma\beta}=K_\beta\left(\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)+K\left(\frac{\alpha_\beta}{\sqrt{G-\frac{F^2}{E}}}cos(\alpha)-\frac{\frac{\partial}{\partial\beta}\sqrt{G-\frac{F^2}{E}}}{G-\frac{F^2}{E}}sin(\alpha)\right)$$

$$=K^2\left(-P_1sin(\alpha)-P_2cos(\alpha)\right)\left(\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right)$$

$$+K\left(\frac{\alpha_\beta}{\sqrt{G-\frac{F^2}{E}}}cos(\alpha)-\frac{\frac{\partial}{\partial\beta}\sqrt{G-\frac{F^2}{E}}}{G-\frac{F^2}{E}}sin(\alpha)\right)$$

$$=K^2\left(-\frac{P_1}{\sqrt{G-\frac{F^2}{E}}}sin^2(\alpha)-\frac{P_2}{\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)-\frac{\alpha_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}sin(\alpha)cos(\alpha)\right)$$

$$+K^2\left(+\frac{\alpha_y-\frac{F}{E}\alpha_x}{G-\frac{F^2}{E}}cos^2(\alpha)+\frac{\frac{\partial}{\partial x}\sqrt{G-\frac{F^2}{E}}}{\sqrt{E}\left(G-\frac{F^2}{E}\right)}sin^2(\alpha)\right)$$

$$+K^2\left(-\frac{\frac{\partial}{\partial y}\sqrt{G-\frac{F^2}{E}}-\frac{F}{E}\frac{\partial}{\partial x}\sqrt{G-\frac{F^2}{E}}}{\sqrt{G-\frac{F^2}{E}}\left(G-\frac{F^2}{E}\right)}sin(\alpha)cos\alpha\right)$$

$$y_{\beta\gamma} = K_\gamma \left(\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}} \right) + K \left(-\frac{\alpha_\gamma}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha) - \frac{\frac{\partial}{\partial\gamma}\sqrt{G-\frac{F^2}{E}}}{G-\frac{F^2}{E}} cos(\alpha) \right)$$

$$= K^2 \left(P_1 cos(\alpha) - P_2 sin(\alpha)\right) \left(\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}} \right)$$

$$+K \left(-\frac{\alpha_\gamma}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha) - \frac{\frac{\partial}{\partial\gamma}\sqrt{G-\frac{F^2}{E}}}{G-\frac{F^2}{E}} cos(\alpha) \right)$$

$$= K^2 \left(\frac{P_1}{\sqrt{G-\frac{F^2}{E}}} cos^2(\alpha) - \frac{P_2}{\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) - \frac{\alpha_x}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}} sin(\alpha)cos(\alpha) \right)$$

$$+K^2 \left(-\frac{\alpha_y - \frac{F}{E}\alpha_x}{G-\frac{F^2}{E}} sin^2(\alpha) - \frac{\frac{\partial}{\partial x}\sqrt{G-\frac{F^2}{E}}}{\sqrt{E}\left(G-\frac{F^2}{E}\right)} cos^2(\alpha) \right)$$

$$+K^2 \left(-\frac{\frac{\partial}{\partial y}\sqrt{G-\frac{F^2}{E}} - \frac{F}{E}\frac{\partial}{\partial x}\sqrt{G-\frac{F^2}{E}}}{\sqrt{G-\frac{F^2}{E}}\left(G-\frac{F^2}{E}\right)} sin(\alpha)cos\alpha \right).$$

The previous computations lead us to $y_{\gamma\beta} - y_{\beta\gamma} = 0$. □

Theorem 5.0.4. *Assume that there exists an isothermic surface which satisfies the system of equation*

$$X_\gamma = \begin{bmatrix} x \\ y \\ K \\ \mathcal{F} \\ f \end{bmatrix}_\gamma = \begin{bmatrix} K\left(\frac{cos(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right) \\ K\left(\frac{sin(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right) \\ K^2\left(P_1 cos(\alpha) - P_2 sin(\alpha)\right) \\ \mathcal{F}KA(\gamma,\beta) \\ Ke^1 \end{bmatrix} \tag{5.26}$$

$$X_\beta = \begin{bmatrix} x \\ y \\ K \\ \mathcal{F} \\ f \end{bmatrix}_\beta = \begin{bmatrix} K\left(\frac{-sin(\alpha)}{\sqrt{E}} - \frac{F}{E}\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right) \\ K\left(\frac{cos(\alpha)}{\sqrt{G-\frac{F^2}{E}}}\right) \\ K^2\left(-P_1 sin(\alpha) - P_2 cos(\alpha)\right) \\ \mathcal{F}KB(\gamma,\beta) \\ Ke^2 \end{bmatrix} \tag{5.27}$$

where $\mathcal{F} =< e^1, e^2, e^3 >, P_1 = \frac{\frac{F^2}{E^2}E_x - 2\frac{F}{E}F_x + G_x}{2(G-\frac{F^2}{E})\sqrt{E}} + \frac{\alpha_y - \frac{F}{E}\alpha_x}{\sqrt{G-\frac{F^2}{E}}}$ *and* $P_2 = \frac{-E_y - \frac{F}{E}E_x + 2F_x}{2E\sqrt{G-\frac{F^2}{E}}} + \frac{\alpha_x}{\sqrt{E}}$,

$$KA = \begin{pmatrix} 0 & \frac{(\sqrt{\tilde{E}})_\beta}{\sqrt{\tilde{G}}} & -\frac{\tilde{l}}{\sqrt{\tilde{E}}} \\ -\frac{(\sqrt{\tilde{E}})_\beta}{\sqrt{\tilde{G}}} & 0 & -\frac{\tilde{m}}{\sqrt{\tilde{G}}} \\ \frac{\tilde{l}}{\sqrt{\tilde{E}}} & \frac{\tilde{m}}{\sqrt{\tilde{G}}} & 0 \end{pmatrix}$$

$$KB = \begin{pmatrix} 0 & -\frac{(\sqrt{\tilde{G}})_\gamma}{\sqrt{\tilde{E}}} & -\frac{\tilde{m}}{\sqrt{\tilde{E}}} \\ \frac{(\sqrt{\tilde{G}})_\gamma}{\sqrt{\tilde{E}}} & 0 & -\frac{\tilde{n}}{\sqrt{\tilde{G}}} \\ \frac{\tilde{m}}{\sqrt{\tilde{E}}} & \frac{\tilde{n}}{\sqrt{\tilde{G}}} & 0 \end{pmatrix}$$

$\tilde{E} = \tilde{G} = K^2$, $\tilde{m} = 0$,

$$\tilde{n} = K^2\left(\frac{Gsin^2(\alpha) + \frac{F^2}{E}cos(2\alpha)}{E\left(G - \frac{F^2}{E}\right)} + \frac{\frac{F}{E}sin(2\alpha)}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}}\right)$$
$$+K^2\left(\left(\frac{cos^2(\alpha)}{G-\frac{F^2}{E}}\right)n - \left(\frac{sin(2\alpha)}{\sqrt{E}\sqrt{G-\frac{F^2}{E}}} + 2\frac{\frac{F}{E}cos^2(\alpha)}{G-\frac{F^2}{E}}\right)m\right)$$

$$\tilde{l} = K^2 \left(\frac{Gcos^2(\alpha) - \frac{F^2}{E}cos(2\alpha)}{E\left(G - \frac{F^2}{E}\right)} - \frac{\frac{F}{E}sin(2\alpha)}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} \right)$$
$$+K^2 \left(\left(\frac{sin^2(\alpha)}{G - \frac{F^2}{E}} \right) n + \left(\frac{sin(2\alpha)}{\sqrt{E}\sqrt{G - \frac{F^2}{E}}} - 2\frac{\frac{F}{E}sin^2(\alpha)}{G - \frac{F^2}{E}} \right) m \right)$$

with initial condition

$$X_0 = \begin{bmatrix} x(0,0) = x_0 \\ y(0,0) = y_0 \\ K(0,0) = 1 \\ \mathcal{F}(0,0) = \mathcal{F}_0 \\ f(0,0) = f_0 \end{bmatrix} \tag{5.28}$$

. *Then, there exists a* Y *such that*

$$Y = \begin{bmatrix} x(a\gamma, a\beta) \\ y(a\gamma, a\beta) \\ aK(a\gamma, a\beta) \\ \mathcal{F}(a\gamma, a\beta) \\ f(a\gamma, a\beta) \end{bmatrix} \tag{5.29}$$

satisfies the same IBVP with the initial condition

$$Y_0 = \begin{bmatrix} x_0 \\ y_0 \\ a \\ \mathcal{F}_0 \\ f_0 \end{bmatrix} \tag{5.30}$$

.

Proof. We split the IBVP into two parts such that:

$$X = \begin{bmatrix} W \\ K \end{bmatrix} \tag{5.31}$$

and

$$W_\gamma(\gamma,\beta) = K(\gamma,\beta) f_1(W(\gamma,\beta)) \tag{5.32}$$
$$K_\gamma(\gamma,\beta) = K^2(\gamma,\beta) g_1(W(\gamma,\beta)) \tag{5.33}$$
$$W_\beta(\gamma,\beta) = K(\gamma,\beta) f_2(W(\gamma,\beta)) \tag{5.34}$$
$$K_\beta(\gamma,\beta) = K^2(\gamma,\beta) g_2(W(\gamma,\beta)) \tag{5.35}$$

and, next, we split for Y, into the corresponding two parts such that:

$$Y = \begin{bmatrix} T \\ U \end{bmatrix} \tag{5.36}$$

and

$$T(\gamma,\beta) = W(a\gamma, a\beta) \tag{5.37}$$
$$U(\gamma,\beta) = aK(a\gamma, a\beta). \tag{5.38}$$
$$\tag{5.39}$$

Clearly, we need to show that T and U solve the IBVP (5.32-5.35) with the given initial condition (5.30).

The partial derivatives become:

$$\begin{aligned} T_\gamma(\gamma,\beta) &= aW_\gamma(a\gamma,a\beta) \\ &= aK(a\gamma,a\beta)f_1(W(a\gamma,a\beta)) \\ &= U(\gamma,\beta)f_1(T(\gamma,\beta)) \end{aligned} \tag{5.40}$$

$$\begin{aligned} T_\beta(\gamma,\beta) &= aW_\beta(a\gamma,a\beta) \\ &= aK(a\gamma,a\beta)f_2(W(a\gamma,a\beta)) \\ &= U(\gamma,\beta)f_2(T(\gamma,\beta)) \end{aligned} \tag{5.41}$$

$$\begin{aligned} U_\gamma(\gamma,\beta) &= a^2K_\gamma(a\gamma,a\beta) \\ &= a^2K^2(a\gamma,a\beta)g_1(W(a\gamma,a\beta)) \\ &= U^2(\gamma,\beta)g_1(T(\gamma,\beta)) \end{aligned} \tag{5.42}$$

$$\begin{aligned} U_\beta(\gamma,\beta) &= a^2K_\beta(a\gamma,a\beta) \\ &= a^2K^2(a\gamma,a\beta)g_2(W(a\gamma,a\beta)) \\ &= U^2(\gamma,\beta)g_2(T(\gamma,\beta)). \end{aligned} \tag{5.43}$$

Therefore,

$$\begin{aligned} T_\gamma(\gamma,\beta) &= U(\gamma,\beta)f_1(T(\gamma,\beta)) \\ U_\gamma(\gamma,\beta) &= U^2(\gamma,\beta)g_1(T(\gamma,\beta)) \end{aligned} \tag{5.44}$$

$$T_\beta(\gamma,\beta) = U(\gamma,\beta) f_2(T(\gamma,\beta))$$
$$U_\beta(\gamma,\beta) = U^2(\gamma,\beta) g_2(T(\gamma,\beta)) \tag{5.45}$$

and it is straight-forward that the initial condition is satisfied:

$$Y \begin{bmatrix} x(0,0) \\ y(0,0) \\ aK(0,0) \\ \mathcal{F}(0,0) \\ f(0,0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ a \\ \mathcal{F}_0 \\ f_0 \end{bmatrix} = Y_0$$

which proves that T and U solve the IBVPs with the corresponding initial condition (5.30).

□

BIBLIOGRAPHY

[1] E.Aulisa, A. Ibragimov, M. Toda, *Geometric Framework for Modeling Nonlinear Flows in Porous Media, and Its Applications in Engineering*, Journal of Nonlinear Analysis, vol.10, (2009).

[2] M. P. do Carmo, *Differential Forms and Applications*

[3] W. Kuhnel, *Differential Geometry, Curves - Surfaces - Manifolds*, A.M.S., 2006.

[4] D. J. Struik, *Lectures on Classical Differential Geometry*, Addison-Wesley Pub. Co., 1961.

[5] E. Kreyzig, *Differential Geometry*, Mathematical Expositions vol. 11, 1959.

[6] V. A. Toponogov, *Differential Geometry of Curves and Surfaces*, Birkhauser, 2006.

[7] G. Kamberov, P. Norman, F. Pedit, U. Pinkall, *Quaternions, Spinors, and Surfaces*, American Mathematical Society, 2002.

[8] K. Tayoma, *Self-Parallel Constant Mean Curvature Surfaces*, Electronic Geometry Model, 2002.03.

[9] U. H. Jeromin, F. Pedit, *Remarks on the Darboux Transform of Isothermic Surfaces*, Doc. Math.J. DMV,

[10] C. Rogers, W. K. Schief, *Backlaund and Darboux Transformations*, Cambridge University Press, 2002.

[11] T. A. Ivey, J. M. Landsberg, *Cartan for Beginners: Differential Geometry via Moving Frames and Exterior Differential Systems*, American Mathematical Society, Graduate Studies in Mathematices,Vol. 61,2003.

[12] A. I. Bobenko, *Exploring Surfaces through Methods from the Theory of Integrable Systems. Lectures on the Bonnet Problem*, 1999.

[13] J.H. Eschenburg, *The Associated Family*, Mathematica Contemporanea, Vol. 31,1-12, 2006.

[14] G. Kamberov, *Prescribing Mean Curvature Existence and Uniqueness Problems*, Electronic research announcements of the AMS, Vol. 4, 4-11, 1998.

[15] J. M. Lee, *Manifolds and Differential Geometry*, AMS, Graduate Studies in Mathematics, 107, 2009.

[16] A. Pressley, *Elementary Differential Geometry*, Springer, 2009.

[17] E. Huhnen-Venedy, *Curvature Line Parametrized Surfaces and Orthogonal Coordinate Systems*, Barret Memorial Lecture Series, 2010.

[18] W. Chen, H. Li, Bonnet Surfaces and Isothermic Surfaces, Results in Mathematics, Birkhauser Verlag, Vol.31, 40-52, 1997.

APPENDIX: COMPUTER CODE

```
//The following computer code solves Bonnet Problem on
curvature line coordinates:

//Bonnet.hpp


using std::cout;
using std::endl;

class bonnet{
  friend bonnet operator*( const double a,const bonnet &other);
  friend bonnet operator/( const double a,const bonnet &other);

public:
  bonnet();
  bonnet(const bonnet &other_bonnet);
  bonnet(const double other[15]);
  double get(const char name[]) const;
  void set(const char name[],const double &other);

  bonnet operator+( const bonnet &other ) const;
  bonnet operator-( const bonnet &other ) const;

  bonnet operator*( const double a) const;
  bonnet operator/( const double a) const;

private:
  double x;
  double y;
  double k;
  double e[3][3];
  double f[3];
};
```

```
bonnet operator*( const double a,const bonnet &other);
bonnet operator/( const double a,const bonnet &other);
std::ostream& operator<<(std::ostream& os, const bonnet&z);


//Bonnet.cpp


#include "main.hpp"
#include "bonnet.hpp"

bonnet::bonnet(){
};

bonnet::bonnet(const bonnet &other){
  x=other.x;
  y=other.y;
  k=other.k;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      e[i][j]=other.e[i][j];
    }
    f[i]=other.f[i];
  }
}

bonnet::bonnet(const double other[15]){
  x=other[0];
  y=other[1];
  k=other[2];
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      e[i][j]=other[3+i*3+j];
    }
    f[i]=other[12+i];
  }
}

double bonnet::get(const char name[]) const{
  double z=0.;
```

```
if(!strcmp(name,"x")){
  z=x;
}
else if(!strcmp(name,"y")){
  z=y;
}
else if(!strcmp(name,"k")){
  z=k;
}
else if(!strcmp(name,"e11")){
  z=e[0][0];
}
else if(!strcmp(name,"e12")){
  z=e[0][1];
}
else if(!strcmp(name,"e13")){
  z=e[0][2];
}
else if(!strcmp(name,"e21")){
  z=e[1][0];
}
else if(!strcmp(name,"e22")){
  z=e[1][1];
}
else if(!strcmp(name,"e23")){
  z=e[1][2];
}
else if(!strcmp(name,"e31")){
  z=e[2][0];
}
else if(!strcmp(name,"e32")){
  z=e[2][1];
}
else if(!strcmp(name,"e33")){
  z=e[2][2];
}
else if(!strcmp(name,"f1")){
  z=f[0];
}
else if(!strcmp(name,"f2")){
```

```
    z=f[1];
  }
  else if(!strcmp(name,"f3")){
    z=f[2];
  }
  else {
    cout<<"Error in Get components!"<<endl;
  }
  return z;
}



void bonnet::set(const char name[],const double &other){
  double z=other;
  if(!strcmp(name,"x")){
    x=z;
  }
  else if(!strcmp(name,"y")){
    y=z;
  }
  else if(!strcmp(name,"k")){
    k=z;
  }
  else if(!strcmp(name,"e11")){
    e[0][0]=z;
  }
  else if(!strcmp(name,"e12")){
    e[0][1]=z;
  }
  else if(!strcmp(name,"e13")){
    e[0][2]=z;
  }
  else if(!strcmp(name,"e21")){
    e[1][0]=z;
  }
  else if(!strcmp(name,"e22")){
    e[1][1]=z;
  }
  else if(!strcmp(name,"e23")){
```

```
    e[1][2]=z;
  }
  else if(!strcmp(name,"e31")){
    e[2][0]=z;
  }
  else if(!strcmp(name,"e32")){
    e[2][1]=z;
  }
  else if(!strcmp(name,"e33")){
    e[2][2]=z;
  }
  else if(!strcmp(name,"f1")){
    f[0]=z;
  }
  else if(!strcmp(name,"f2")){
    f[1]=z;
  }
  else if(!strcmp(name,"f3")){
    f[2]=z;
  }
  else {
    cout<<"Error in Set components!"<<endl;
  }
}


bonnet bonnet::operator+( const bonnet &other ) const{
  bonnet T;
  T.x=x+other.x;
  T.y=y+other.y;
  T.k=k+other.k;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      T.e[i][j]=e[i][j]+other.e[i][j];
    }
    T.f[i]=f[i]+other.f[i];
  }


  return T;
```

```
}


bonnet bonnet::operator-( const bonnet &other ) const{
  bonnet T;
  T.x=x-other.x;
  T.y=y-other.y;
  T.k=k-other.k;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      T.e[i][j]=e[i][j]-other.e[i][j];
    }
    T.f[i]=f[i]-other.f[i];
  }
  return T;

}


bonnet bonnet::operator*( const double a) const{
  bonnet T;
  T.x=x*a;
  T.y=y*a;
  T.k=k*a;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      T.e[i][j]=e[i][j]*a;
    }
    T.f[i]=f[i]*a;
  }


  return T;

}
bonnet bonnet::operator/( const double a) const{
  bonnet T;
  T.x=x/a;
  T.y=y/a;
```

```
  T.k=k/a;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      T.e[i][j]=e[i][j]/a;
    }
    T.f[i]=f[i]/a;
  }


  return T;

}


bonnet operator*( const double a,const bonnet &other){
  bonnet T;
  T.x=other.x*a;
  T.y=other.y*a;
  T.k=other.k*a;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      T.e[i][j]=other.e[i][j]*a;
    }
    T.f[i]=other.f[i]*a;
  }

  return T;
}

bonnet operator/( const double a,const bonnet &other){
  bonnet T;
  T.x=other.x/a;
  T.y=other.y/a;
  T.k=other.k/a;
  for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
      T.e[i][j]=other.e[i][j]/a;
    }
    T.f[i]=other.f[i]/a;
  }
```

```
  return T;
}
std::ostream& operator<<(std::ostream& os, const bonnet&z){
  os<<"x="<<z.get("x")<<endl;
  os<<"y="<<z.get("y")<<endl;
  os<<"k="<<z.get("k")<<endl;
  os<<"e1={"<<z.get("e11")
    <<" , " <<z.get("e12")
    <<" , " <<z.get("e13")
    <<"}"<<endl;
  os<<"e2={"<<z.get("e21")
    <<" , " <<z.get("e22")
    <<" , " <<z.get("e23")
    <<"}"<<endl;
  os<<"e3={"<<z.get("e31")
    <<" , " <<z.get("e32")
    <<" , " <<z.get("e33")
    <<"}"<<endl;
  os<<"f={"<<z.get("f1")
    <<" , " <<z.get("f2")
    <<" , " <<z.get("f3")
    <<"}"<<endl;
  return os;
}
```

```
//FunForm.hpp

class bonnet;

class FunForm{
public:
  FunForm(double (*E_other)(double x,double y),
  double (*F_other)(double x,double y),
  double (*G_other)(double x,double y),
```

```
  double (*l_other)(double x,double y),
  double (*m_other)(double x,double y),
  double (*n_other)(double x,double y),
  double (*Ex_other)(double x,double y),
  double (*Fx_other)(double x,double y),
  double (*Gx_other)(double x,double y),
  double (*Ey_other)(double x,double y),
  double (*Fy_other)(double x,double y),
  double (*Gy_other)(double x,double y),
  double (*lx_other)(double x,double y),
  double (*mx_other)(double x,double y),
  double (*nx_other)(double x,double y),
  double (*ly_other)(double x,double y),
  double (*my_other)(double x,double y),
  double (*ny_other)(double x,double y));
  bonnet getKg(const bonnet X,const double alpha0, double &alpha);
  bonnet getKb(const bonnet X,const double alpha0, double &alpha);

private:

  double getAlpha(double x,double y);


  double (*E)(double x,double y);
  double (*F)(double x,double y);
  double (*G)(double x,double y);
  double (*l)(double x,double y);
  double (*m)(double x,double y);
  double (*n)(double x,double y);

  double (*Ex)(double x,double y);
  double (*Fx)(double x,double y);
  double (*Gx)(double x,double y);

  double (*Ey)(double x,double y);
  double (*Fy)(double x,double y);
  double (*Gy)(double x,double y);

  double (*lx)(double x,double y);
  double (*mx)(double x,double y);
```

```
  double (*nx)(double x,double y);
  double (*ly)(double x,double y);
  double (*my)(double x,double y);
  double (*ny)(double x,double y);

};
```

```
// FunForm.cpp

#include "main.hpp"
#include "FunForm.hpp"
#include "bonnet.hpp"

FunForm::FunForm(double (*E_other)(double x,double y),
 double (*F_other)(double x,double y),
 double (*G_other)(double x,double y),
 double (*l_other)(double x,double y),
 double (*m_other)(double x,double y),
 double (*n_other)(double x,double y),
 double (*Ex_other)(double x,double y),
 double (*Fx_other)(double x,double y),
 double (*Gx_other)(double x,double y),
 double (*Ey_other)(double x,double y),
 double (*Fy_other)(double x,double y),
 double (*Gy_other)(double x,double y),
 double (*lx_other)(double x,double y),
 double (*mx_other)(double x,double y),
 double (*nx_other)(double x,double y),
 double (*ly_other)(double x,double y),
 double (*my_other)(double x,double y),
 double (*ny_other)(double x,double y)){
  E=E_other;
  F=F_other;
  G=G_other;
  l=l_other;
  m=m_other;
  n=n_other;
```

```
  Ex=Ex_other;
  Fx=Fx_other;
  Gx=Gx_other;
  Ey=Ey_other;
  Fy=Fy_other;
  Gy=Gy_other;
  lx=lx_other;
  mx=mx_other;
  nx=nx_other;
  ly=ly_other;
  my=my_other;
  ny=ny_other;
}


bonnet FunForm::getKg(const bonnet X,const double alpha0, double &alpha){
  double a[15];



  double x=X.get("x");
  double y=X.get("y");
  double k=X.get("k");
  double e[3][3];
  double f[3];
  e[0][0]=X.get("e11");
  e[0][1]=X.get("e12");
  e[0][2]=X.get("e13");
  e[1][0]=X.get("e21");
  e[1][1]=X.get("e22");
  e[1][2]=X.get("e23");
  e[2][0]=X.get("e31");
  e[2][1]=X.get("e32");
  e[2][2]=X.get("e33");
  f[0]=X.get("f1");
  f[1]=X.get("f2");
  f[2]=X.get("f3");


  double EE=E(x,y);
```

```
  double FF=F(x,y);
  double GG=G(x,y);

  double ll=l(x,y);
  double mm=m(x,y);
  double nn=n(x,y);

  double EE2=EE*EE;
  double FF2=FF*FF;
  double GF2E=GG-FF2/EE;

  double AA=FF/EE*ll-mm;
  double BB=( (2.*FF2/EE-GG)*ll - 2.*FF*mm+EE*nn);
  double CC=sqrt(EE*GF2E);

  alpha=0.5*atan(2.*AA*CC/BB);

  int imin;
  if(alpha>=0){
    double dalpha=fabs(alpha-alpha0);
    imin=0;
    for(int i=1;i<=3;i++){
      if(fabs(alpha-alpha0+i*0.5*acos(-1.))<dalpha ){
imin=i;
dalpha=fabs(alpha-alpha0+i*0.5*acos(-1.));
      }
    }
  }
  else{
    double dalpha=fabs(alpha-alpha0+0.5*acos(-1.));
    imin=1;
    for(int i=2;i<=4;i++){
      if(fabs(alpha-alpha0+i*0.5*acos(-1.))<dalpha ){
imin=i;
dalpha=fabs(alpha-alpha0+i*0.5*acos(-1.));
      }
    }
  }

  alpha+=imin*0.5*acos(-1.);
```

```
  double EEx=Ex(x,y);
  double FFx=Fx(x,y);
  double GGx=Gx(x,y);

  double EEy=Ey(x,y);
  double FFy=Fy(x,y);
  double GGy=Gy(x,y);

  double llx=lx(x,y);
  double mmx=mx(x,y);
  double nnx=nx(x,y);

  double lly=ly(x,y);
  double mmy=my(x,y);
  double nny=ny(x,y);

  double AAx=FFx/EE*ll+FF/EE*llx-FF/EE2*EEx*ll-mmx;
  double AAy=FFy/EE*ll+FF/EE*lly-FF/EE2*EEy*ll-mmy;

  double BBx=( (4.*FF*FFx/EE-2.*FF2*EEx/EE2-GGx)*ll
+ (2.*FF2/EE-GG)*llx - 2.*(FFx*mm+FF*mmx) + (EEx*nn+EE*nnx) );
  double BBy=( (4.*FF*FFy/EE-2.*FF2*EEy/EE2-GGy)*ll
+ (2.*FF2/EE-GG)*lly - 2.*(FFy*mm+FF*mmy) + (EEy*nn+EE*nny) );

  double CCx=0.5/sqrt(EE*GF2E)*(EEx*GF2E+EE*(GGx
- 2.*FF*FFx/EE + FF2*EEx/(EE*EE)) );
  double CCy=0.5/sqrt(EE*GF2E)*(EEy*GF2E+EE*(GGy
 - 2.*FF*FFy/EE + FF2*EEy/(EE*EE)) );

  double alphax=0.5/(1+4.*AA*AA*CC*CC/(BB*BB))*
2.*(AAx*CC/BB+AA*CCx/BB-AA*CC*BBx/(BB*BB));
  double alphay=0.5/(1+4.*AA*AA*CC*CC/(BB*BB))*
2.*(AAy*CC/BB+AA*CCy/BB-AA*CC*BBy/(BB*BB));

  double Es=k*k;
  double Gs=k*k;
```

```
  double cosa=cos(alpha);
  double cosa2=cosa*cosa;
  double sina=sin(alpha);
  double sina2=sina*sina;
  double sin2a=sin(2.*alpha);
  double cos2a=cos(2.*alpha);

  double ls=k*k*( ( (GG*cosa2-FF2/EE*cos2a)/(EE*GF2E)
                      - (FF/EE*sin2a)/sqrt(EE*GF2E) ) * ll+
  sina2/GF2E * nn +
  ( sin2a/sqrt(EE*GF2E)-2.*(FF/EE*sina2)/ GF2E ) * mm );

  double ms=0.;


  double P1=(FF2/EE2*EEx-2.*FF/EE*FFx+GGx)
/(2.*GF2E*sqrt(EE))+(alphay-FF/EE*alphax)/sqrt(GF2E);
  double P2=(-EEy-FF/EE*EEx+2.*FFx)/(2.*sqrt(GF2E)*EE)+alphax/sqrt(EE);

  double kg=k*k*( cosa*P1-sina*P2);
  double kb=k*k*(-sina*P1-cosa*P2);

  double Esb=kb;      //sqrt(Es)_beta=k_beta

  a[0]=k*( cosa/sqrt(EE)-FF/EE*sina/sqrt(GF2E) );
  a[1]=k*( sina/sqrt(GF2E) );
  a[2]=kg;
  a[3]=-(Esb/sqrt(Gs))*e[1][0]+(ls/sqrt(Es))*e[2][0];      //e11gamma
  a[4]=-(Esb/sqrt(Gs))*e[1][1]+(ls/sqrt(Es))*e[2][1];      //e12gamma
  a[5]=-(Esb/sqrt(Gs))*e[1][2]+(ls/sqrt(Es))*e[2][2];      //e13gamma
  a[6]=+(Esb/sqrt(Gs))*e[0][0]+(ms/sqrt(Gs))*e[2][0];      //e21gamma
  a[7]=+(Esb/sqrt(Gs))*e[0][1]+(ms/sqrt(Gs))*e[2][1];      //e22gamma
  a[8]=+(Esb/sqrt(Gs))*e[0][2]+(ms/sqrt(Gs))*e[2][2];      //e23gamma
  a[9]=-(ls/sqrt(Es))*e[0][0]-(ms/sqrt(Gs))*e[1][0];       //e31gamma
  a[10]=-(ls/sqrt(Es))*e[0][1]-(ms/sqrt(Gs))*e[1][1];      //e32gamma
  a[11]=-(ls/sqrt(Es))*e[0][2]-(ms/sqrt(Gs))*e[1][2];      //e33gamma
  a[12]=sqrt(Es)*e[0][0];                                  //f1gamma
  a[13]=sqrt(Es)*e[0][1];                                  //f2gamma
  a[14]=sqrt(Es)*e[0][2];                                  //f3gamma
```

```
  bonnet KK(a);
  return KK;
}

bonnet FunForm::getKb(const bonnet X,const double alpha0, double &alpha)
{


  double b[15];


  double x=X.get("x");
  double y=X.get("y");
  double k=X.get("k");
  double e[3][3];
  double f[3];
  e[0][0]=X.get("e11");
  e[0][1]=X.get("e12");
  e[0][2]=X.get("e13");
  e[1][0]=X.get("e21");
  e[1][1]=X.get("e22");
  e[1][2]=X.get("e23");
  e[2][0]=X.get("e31");
  e[2][1]=X.get("e32");
  e[2][2]=X.get("e33");
  f[0]=X.get("f1");
  f[1]=X.get("f2");
  f[2]=X.get("f3");


  double EE=E(x,y);
  double FF=F(x,y);
  double GG=G(x,y);

  double ll=l(x,y);
  double mm=m(x,y);
  double nn=n(x,y);
```

```
  double EE2=EE*EE;
  double FF2=FF*FF;
  double GF2E=GG-FF2/EE;

  double AA=FF/EE*ll-mm;
  double BB=( (2.*FF2/EE-GG)*ll - 2.*FF*mm+EE*nn);
  double CC=sqrt(EE*GF2E);

  alpha=0.5*atan(2.*AA*CC/BB);

  int imin;
  if(alpha>=0){
    double dalpha=fabs(alpha-alpha0);
    imin=0;
    for(int i=1;i<=3;i++){
      if(fabs(alpha-alpha0+i*0.5*acos(-1.))<dalpha ){
imin=i;
dalpha=fabs(alpha-alpha0+i*0.5*acos(-1.));
      }
    }
  }
  else{
    double dalpha=fabs(alpha-alpha0+0.5*acos(-1.));
    imin=1;
    for(int i=2;i<=4;i++){
      if(fabs(alpha-alpha0+i*0.5*acos(-1.))<dalpha ){
imin=i;
dalpha=fabs(alpha-alpha0+i*0.5*acos(-1.));
      }
    }
  }

  alpha+=imin*0.5*acos(-1.);




  double EEx=Ex(x,y);
  double FFx=Fx(x,y);
  double GGx=Gx(x,y);
```

```
  double EEy=Ey(x,y);
  double FFy=Fy(x,y);
  double GGy=Gy(x,y);

  double llx=lx(x,y);
  double mmx=mx(x,y);
  double nnx=nx(x,y);

  double lly=ly(x,y);
  double mmy=my(x,y);
  double nny=ny(x,y);

  double AAx=FFx/EE*ll+FF/EE*llx-FF/EE2*EEx*ll-mmx;
  double AAy=FFy/EE*ll+FF/EE*lly-FF/EE2*EEy*ll-mmy;

  double BBx=( (4.*FF*FFx/EE-2.*FF2*EEx/EE2-GGx)*ll
+ (2.*FF2/EE-GG)*llx - 2.*(FFx*mm+FF*mmx) + (EEx*nn+EE*nnx) );
  double BBy=( (4.*FF*FFy/EE-2.*FF2*EEy/EE2-GGy)*ll
+ (2.*FF2/EE-GG)*lly - 2.*(FFy*mm+FF*mmy) + (EEy*nn+EE*nny) );

  double CCx=0.5/sqrt(EE*GF2E)*(EEx*GF2E+EE*(GGx
- 2.*FF*FFx/EE + FF2*EEx/(EE*EE)) );
  double CCy=0.5/sqrt(EE*GF2E)*(EEy*GF2E+EE*(GGy
- 2.*FF*FFy/EE + FF2*EEy/(EE*EE)) );

  double alphax=0.5/(1+4.*AA*AA*CC*CC/(BB*BB))*
2.*(AAx*CC/BB+AA*CCx/BB-AA*CC*BBx/(BB*BB));
  double alphay=0.5/(1+4.*AA*AA*CC*CC/(BB*BB))*
2.*(AAy*CC/BB+AA*CCy/BB-AA*CC*BBy/(BB*BB));

  //  cout<<alphax<<" "<<alphay<<endl;

  double Es=k*k;
  double Gs=k*k;

  double cosa=cos(alpha);
  double cosa2=cosa*cosa;
  double sina=sin(alpha);
  double sina2=sina*sina;
```

```
  double sin2a=sin(2.*alpha);
  double cos2a=cos(2.*alpha);

  double ms=0.;

  double ns=k*k*( ( (GG*sina2+FF2/EE*cos2a)/(EE*GF2E)
                  + (FF/EE*sin2a)/sqrt(EE*GF2E) ) *ll+
  cosa2/GF2E * nn -
  ( sin2a/sqrt(EE*GF2E)+2.*(FF/EE*cosa2)/ GF2E ) * mm );


  double P1=(FF2/EE2*EEx-2.*FF/EE*FFx+GGx)
          /(2.*GF2E*sqrt(EE))+(alphay-FF/EE*alphax)/sqrt(GF2E);
  double P2=(-EEy-FF/EE*EEx+2.*FFx)/(2.*sqrt(GF2E)*EE)+alphax/sqrt(EE);

  double kg=k*k*( cosa*P1-sina*P2);
  double kb=k*k*(-sina*P1-cosa*P2);

  double Gsg=kg;

  b[0]=k*( -sina/sqrt(EE)-FF/EE*cosa/sqrt(GF2E) );
  b[1]=k*( cosa/sqrt(GF2E) );
  b[2]=kb;
  b[3]=(Gsg/sqrt(Es))*e[1][0]+(ms/sqrt(Es))*e[2][0];       //e11beta
  b[4]=(Gsg/sqrt(Es))*e[1][1]+(ms/sqrt(Es))*e[2][1];       //e12beta
  b[5]=(Gsg/sqrt(Es))*e[1][2]+(ms/sqrt(Es))*e[2][2];       //e13beta
  b[6]=-(Gsg/sqrt(Es))*e[0][0]+(ns/sqrt(Gs))*e[2][0];      //e21beta
  b[7]=-(Gsg/sqrt(Es))*e[0][1]+(ns/sqrt(Gs))*e[2][1];      //e22beta
  b[8]=-(Gsg/sqrt(Es))*e[0][2]+(ns/sqrt(Gs))*e[2][2];      //e23beta
  b[9]=-(ms/sqrt(Es))*e[0][0]-(ns/sqrt(Gs))*e[1][0];       //e31beta
  b[10]=-(ms/sqrt(Es))*e[0][1]-(ns/sqrt(Gs))*e[1][1];      //e32beta
  b[11]=-(ms/sqrt(Es))*e[0][2]-(ns/sqrt(Gs))*e[1][2];      //e33beta
  b[12]=sqrt(Gs)*e[1][0];                                  //f1beta
  b[13]=sqrt(Gs)*e[1][1];                                  //f2beta
  b[14]=sqrt(Gs)*e[1][2];                                  //f3beta


bonnet KK(b);
  return KK;
}
```

```
double FunForm::getAlpha(double x,double y){
  double alpha;

  alpha=0.5*atan(2.*(F(x,y)*l(x,y)/E(x,y)-m(x,y))
*sqrt(G(x,y)*E(x,y)-F(x,y)*F(x,y))
       /((2*F(x,y)*F(x,y)/E(x,y))*l(x,y)-2*F(x,y)*m(x,y)+E(x,y)*n(x,y)) );
  return alpha;
}


// main.hpp

#include<iostream>
#include<fstream>
#include<cstdlib>
#include<cmath>
#include<cstring>



// main.cpp

#include "main.hpp"
#include "bonnet.hpp"
#include "FunForm.hpp"

double E(double x,double y),Ex(double x,double y),Ey(double x,double y);
double F(double x,double y),Fx(double x,double y),Fy(double x,double y);
double G(double x,double y),Gx(double x,double y),Gy(double x,double y);
double l(double x,double y),lx(double x,double y),ly(double x,double y);
double m(double x,double y),mx(double x,double y),my(double x,double y);
double n(double x,double y),nx(double x,double y),ny(double x,double y);

int main(){
```

```
  FunForm H(E,F,G,l,m,n,Ex,Fx,Gx,Ey,Fy,Gy,lx,mx,nx,ly,my,ny);

  double w[3]={0.5,0.5,1.};
  double a[15]={0,0,1,0,1,0,0,0,1,1,0,0,0,0,0};


  bonnet X0(a);


  double B=sqrt(1.)*2.*acos(-1.);
  double G=3.;
  int nx=20;
  int ny=100;
  double hg=G/nx;
  double hb=B/ny;


  bonnet *XX_mem=new bonnet [(nx+1)*(ny+1)];
  bonnet **XX=new bonnet * [nx+1];

  double *alpha_mem=new double [(nx+1)*(ny+1)];
  double **alpha=new double *[nx+1];

  for(int i=0;i<=nx;i++){
    XX[i]=&XX_mem[(ny+1)*i];
    alpha[i]=&alpha_mem[(ny+1)*i];
  }

  XX[0][0]=X0;
  H.getKg(X0,1.*acos(-1.)/2,alpha[0][0]);
  cout<<XX[0][0].get("f1")<<" "<<XX[0][0].get("f2")
<<" "<<XX[0][0].get("f3")<<" "<<XX[0][0].get("k")
<<" "<<alpha[0][0]<<endl;

  for(int i=1;i<=nx;i++){
    bonnet X[4];
    bonnet Kg[4];
    X[0]=XX[i-1][0];
    for(int k=0;k<3;k++){
      Kg[k]=H.getKg(X[k],alpha[i-1][0],alpha[i][0]);
```

```
      X[k+1]=X[0]+w[k]*hg*Kg[k];
    }
    Kg[3]=H.getKg(X[3],alpha[i-1][0],alpha[i][0]);
    XX[i][0]=XX[i-1][0]+hg*(Kg[0]+2.*Kg[1]+2.*Kg[2]+Kg[3])/6.;
    cout<<XX[i][0].get("f1")<<" "<<XX[i][0].get("f2")<<" "
<<XX[i][0].get("f3")<<" "<<XX[i][0].get("k")<<" "<<alpha[i][0]
<<endl;
  }
  cout<<endl;


  for(int i=0;i<=nx;i++){
    cout<<XX[i][0].get("f1")<<" "<<XX[i][0].get("f2")
<<" "<<XX[i][0].get("f3")<<" "
<<XX[i][0].get("k")<<" "<<alpha[i][0]<<endl;
    for(int j=1;j<=ny;j++){
      bonnet X[4];
      bonnet Kb[4];
      X[0]=XX[i][j-1];
      for(int k=0;k<3;k++){
Kb[k]=H.getKb(X[k],alpha[i][j-1],alpha[i][j]);
X[k+1]=X[0]+w[k]*hb*Kb[k];
      }
      Kb[3]=H.getKb(X[3],alpha[i][j-1],alpha[i][j]);
      XX[i][j]=XX[i][j-1]+hb*(Kb[0]+2.*Kb[1]+2.*Kb[2]+Kb[3])/6.;
      cout<<XX[i][j].get("f1")<<" "<<XX[i][j].get("f2")
<<" "<<XX[i][j].get("f3")
<<" "<<XX[i][j].get("k")<<" "<<alpha[i][j]<<endl;
    }
    cout<<endl;
  }


  //open output file;
  FILE* script = 0;
  script = fopen("./output/dual.gmv", "w");
  if (script == NULL){
    printf("I could not create a script file to write to.
 Exiting...\n");
    return 0;
```

```
}


fprintf(script, "gmvinput ascii\nnodes
-2 %d %d 1\n",ny+1,nx+1);

for(int i=0;i<=nx;i++){
  for(int j=0;j<=ny;j++){
    //  cout<<f[0][i][j];
    fprintf(script,"%f ",XX[i][j].get("f1"));
  }
}

fprintf(script,"\n");

for(int i=0;i<=nx;i++){
  for(int j=0;j<=ny;j++){
    fprintf(script,"%f ",XX[i][j].get("f2"));
  }
}
fprintf(script,"\n");

for(int i=0;i<=nx;i++){
  for(int j=0;j<=ny;j++){
    fprintf(script,"%f ",XX[i][j].get("f3"));
  }
}

fprintf(script,"\n");

fprintf(script, "cells 0 \nendgmv\n");
fclose(script);

// **********
// GMV OUTPUT
// **********


delete [] XX;
delete [] XX_mem;
```

```
  delete [] alpha;
  delete [] alpha_mem;

  return 0;
}


double E(double x,double y){
  return 1.;
}
double F(double x,double y){
  return -cos(x)*sin(y);
}
double G(double x,double y){

  return 1.;
}


double Ex(double x,double y){
  return 0.;
}
double Ey(double x,double y){
  return 0.;
}
double Fx(double x,double y){
  return sin(x)*sin(y);
}
double Fy(double x,double y){
  return -cos(x)*cos(y);
}
double Gx(double x,double y){
  return 0.;
}
double Gy(double x,double y){
  return 0.;
}
```

```
double l(double x,double y){
  double den=sqrt(cos(x)*cos(x)*cos(y)*cos(y)+sin(x)*sin(x));
  return -cos(y)/den;
}
double m(double x,double y){
  return 0.;
}
double n(double x,double y){
  double den=sqrt(cos(x)*cos(x)*cos(y)*cos(y)+sin(x)*sin(x));
  return -sin(x)/den;
}


double lx(double x,double y){
 double den=sqrt(cos(x)*cos(x)*cos(y)*cos(y)+sin(x)*sin(x));
 return cos(y)*(sin(2*x)-sin(2*x)*cos(y)*cos(y))/(2*den*den*den);
}
double ly(double x,double y){
  double den=sqrt(cos(x)*cos(x)*cos(y)*cos(y)+sin(x)*sin(x));
  return -(cos(x)*cos(x)*cos(y)*cos(y)*sin(x))
/(den*den*den)+sin(y)/den;
}
double mx(double x,double y){
  return 0.;
}
double my(double x,double y){
  return 0.;
}
double nx(double x,double y){
  double den=sqrt(cos(x)*cos(x)*cos(y)*cos(y)+sin(x)*sin(x));
  return  sin(x)*(sin(2*x)-sin(2*x)*cos(y)*cos(y))
/(2*den*den*den)-cos(x)/den;
}
double ny(double x,double y){
  double den=sqrt(cos(x)*cos(x)*cos(y)*cos(y)+sin(x)*sin(x));
  return  -(cos(x)*cos(x)*cos(y)*sin(x)*sin(y))/(den*den*den);
}
```