AUGMENTED LAGRANGE METHOD APPLIED TO SURFACE GRID GENERATIONS IN TWO AND THREE DIMENSIONS

by

MANJULA BUDDHAPRIYA KOTTEGODA, B.S.

A THESIS

IN

MATHEMATICS

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

MASTER OF SCIENCE

Approved

May, 1996

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor Dr. Carrol J. Nunn for his guidance and encouragement throughout the preparation of this thesis. I would also like to thank Dr. Zhimin Zhang and Dr. Edward Allen for their help.

Last but not least, I wish to thank my parents. Dr. M. B. Kottegoda and Mrs. Ramya Kottegoda, for their consistent encouragement in the pursuit of my higher studies. To them I dedicate this work.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

The development of grid generation came about from the need to compute solutions to partial differential equations associated with computational fluid dynamics on complex physical regions. The need has been escalated by the use of Finite Element Method (FEM) in engineering, with the advances in computer technology. Hence it is increasingly apparent that as physical geometries become more and more complex, fast efficient algorithms are needed to produce high quality grids. This paper will discuss an algorithm for generating grids on the surface of a region.

Two common methods of grid generation are the Delaunay triangulation method and the mapped grid generation approach. We will discuss the mapped grid generation approach. In most studies, researchers have concentrated their efforts on generation of grids on the interior of the domain with less effort devoted to surface grid generation. In this thesis, we will concentrate on techniques of grid generation on the surface of a given domain.

The mapped grid generation approach consists of generating a reference grid on a simple (logical) region and moving the points of the reference grid to the target arbitrary-shaped (physical) region or domain. The physical region of the problem will be called the physical space and the logical region will be called the logical space. The set of points in the logical space, when transformed or mapped to the physical space then forms a grid. This process produces mapped grids, and hence is called mapped grid generation. By choosing the correct combination of reference grid and mapping technique an improved grid can be created in the the physical space or on the surface of the space.

The advantage of using mapped grid generation, as discussed by Thompson, Warsi and Mastin [1], is that it can be used to create quadrilateral elements in two dimensions, and tetrahedral or pentahedral elements in three dimensions. These types of elements are commonly used in FEM. The disadvantage of these techniques is that more user interaction is required in the mapped grid generation techniques, Nunn [2].

One method of mapped grid generation can be viewed as creating a grid by moving the points from the logical to the physical region. This is done by minimizing a function involving the points of the grid after the boundary points are moved to the boundary of the physical space by a given mapping function. This method is used by Kennon and Dulikrvich [3] and Castillo [4]. The method described in this paper is an extension of the ideas of Castillo, Steinberg [5], and Roache [6]. whose main concentration was finding a mapping in the interior of the particular physical region. They concentrated on an appropriate discretization of a Euler-Lagrange equations for the variational problem to get algorithms for generating grids. We will propose a method for surface grid generation based on these known methods. and discuss practical issues involved in implementing the method. The technique we will propose will require that we find a constrained minimum of a n-dimensional function. In our study we use the Augmented Lagrange method to solve our constrained optimization problem. This paper will also discuss the existence of a constrained minimum for the optimization problem. Limitations on the physical surface will be studied in the two-dimensional case and this will give us intuition about what to expect in the three-dimensional case.

Chapter II will consist of an introduction to our algorithm. The general idea of the method will be presented together with a brief introduction of the minimization technique. Examples of the method applied to sample surfaces will also be given. We will explore the issues related to the optimization using quadratic two-dimensional surfaces. The third chapter will describe the problem in the two-dimensional case with justifications for why the method should work. We will also discuss the conditions under which a strict local minimum can be expected. The next chapter will consist of some examples for sample surface shapes. Finally. Chapter V discusses possible extensions of the two-dimensional ideas of the technique and algorithm to three-dimensions along with a conclusion of the research done in the paper.

CHAPTER II
PRELIMINARIES

This section will start with a brief introduction of the problem and the proposed method of surface grid genetarion. This is followed by a general overview of the Augmented Lagrange method and how it is used in the proposed method.

The main goal of this paper is to create a reasonable grid on the surface of a given domain. Creating a grid on the surface can be seen as creating a collection of points on the surface connected by lines. There are many measures of how reasonable a grid is. One of those measures is grid smoothness. The smoothness can be described as how smoothly the spacing and the angles of the grid vary [8]. The reason being that when a grid is used for certain numerical computations the smallest error often occurs on the smoothest grid. In our algorithm we will achieve a smooth grid by minimizing a function which involves the line segments connecting the points of the grid. The simplest case is the minimization of the function which is the sum of the squares of the distance between connecting points.

In this section, we are going to talk about the Augmented Lagrange method and its relevance to our algorithm. As explained earlier the smoothness of the algorithm depends on minimizing a certain function. and we will constrain the surface points to stay on the surface. Hence the minimization is a constrained minimization problem. A general constrained minimization problem is as follows:

$$\begin{aligned} &\text{minimize } \mathbf{f(x)} \\ &\text{subject to } \mathbf{g(x) = 0} \end{aligned} \tag{2.1}$$

Where $\mathbf{f : x \subset R^n \rightarrow R}$ and $\mathbf{g : x \subset R^n \rightarrow R^m}$ are given functions.

We will assume that we know a method for solving the unconstrained minimization problem

$$\begin{aligned} &\text{minimize } \mathbf{f(x)} \\ &\text{Where } \mathbf{f : x \subset R^n \rightarrow R}. \end{aligned} \tag{2.2}$$

For example a conjugate gradient method.

A method frequently used to approximate solutions for (2.1) is the penalty method. A simple application of the method consists of defining a function,

$$L_p(\mathbf{x}) = \mathrm{f}(\mathbf{x}) + \mathrm{P}\|\mathbf{g}(\mathbf{x})\|^2 \tag{2.3}$$

where P is a positive large number. The basic algorithm consists of finding the $\mathbf{x}$ which minimizes $L_p(\mathbf{x})$ after which the penalty is increased and $L_p(\mathbf{x})$ minimized again. This is done iteratively until P is sufficiently large, then at a minimum $\mathbf{x}^*$ of (2.3) the constraint $\mathrm{g}(\mathbf{x}^*)$ becomes sufficiently close to 0. And $\mathbf{x}^*$ in turn gets sufficiently close to $\mathbf{x}_{min}$ which is the actual solution to (2.1). But as P increases minimizing (2.3) becomes increasingly difficult [9]. Therefore to get around this difficulty we introduce Lagrange multipliers to the minimizing equation. This takes us to the Augmented Lagrange method. In Luenberger[10] the following motivation for altering the penalty method is given in detail.

Let $\mathbf{x}^*$ be a local extremum of f subject to the constraints $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Under relatively straightforward conditions there exists a Lagrange multiplier $\lambda^* \in \mathbf{R^m}$ such that

$$\bigtriangledown \mathrm{f}(\mathbf{x}^*) + \lambda^{*T} \bigtriangledown \mathrm{g}(\mathbf{x}^*) = 0. \tag{2.4}$$

If λ^* is known then there exists a P^* where for any $P > \mathrm{P}^*$

$$\mathrm{L}(\mathbf{x}, P) = \mathrm{f}(\mathbf{x}) + \lambda^{*T}\mathrm{g}(\mathbf{x}) + P\|\mathbf{g}(\mathbf{x})\|^2 \tag{2.5}$$

has the same minimum as problem (2.1). Since λ^* is not always known, we modify (2.5) as following

$$\mathcal{L}(\mathbf{x},\lambda,\mathrm{P}) = \mathrm{f}(\mathbf{x}) + \lambda^T\, \mathrm{g}(\mathbf{x}) + \mathrm{P}\|\mathbf{g}(\mathbf{x})\|^2. \tag{2.6}$$

From (2.6) we get the following algorithm for the Augmented Lagrange method.

Algorithm : (2.7)

1. Choose initial values $\mathbf{x}_0$,λ_0.$\mathbf{P_0}$.
2. On iteration i we do the following:
   Let $\mathbf{x}_{i+1} = \min \mathcal{L}(\mathbf{x}_i,\lambda_i,\mathrm{P}_i)$
   Let $\lambda_{i+1} = \lambda_i + 2\mathrm{P}_i\mathrm{g}(\mathbf{x}_i)$
   $\mathrm{P}_{i+1} = 2\mathrm{P}_i$
   Iteration until $\|\mathbf{g}\|$ is small.

Therefore for a sufficiently large P as λ approaches λ^* the minimum of $\mathbf{x}$ of (2.6) would approach $\mathbf{x}_{min}$. The Augmented Lagrange method is very effective in solving problems like (2.1), which is what we are faced with in generating the grid.

To conclude our preliminaries, we will give a brief explanation as to why we think that our method is reasonable to explore. This is done by providing a simple geometric illustration. Later on in the paper, more examples of this technique on simple surfaces will be given.

Take, for example, the physical surface to be a circle. and say there are n points on the surface of this circle. Then according to our previous notation we will have to minimize a function provided that the n points are on the surface. Which brings us to minimizing the function below.

$$\mathrm{f}(\vec{\mathbf{x}}) = (x_1 - x_2)^2 + (y_1 - y_2)^2 + \ldots + (x_{n-1} - x_n)^2 + (y_{n-1} - y_n)^2 + (x_n - x_1)^2 + (y_n - y_1)^2$$

$$\text{where } \vec{\mathbf{x}} = [(x_1, y_1) , \ldots , (x_n, y_n)]$$

$$\text{subject to } \mathbf{g}(\vec{\mathbf{x}})=0 \text{ where } \mathrm{g}_i(\vec{\mathbf{x}}) = x_i^2 + y_i^2 - 1 = 0 \quad \text{for } 1 \leq i \leq \mathrm{n}. \tag{2.8}$$

Figure 2.1 (below) shows the geometric setup.Where $N_i = (x_i, y_i)$

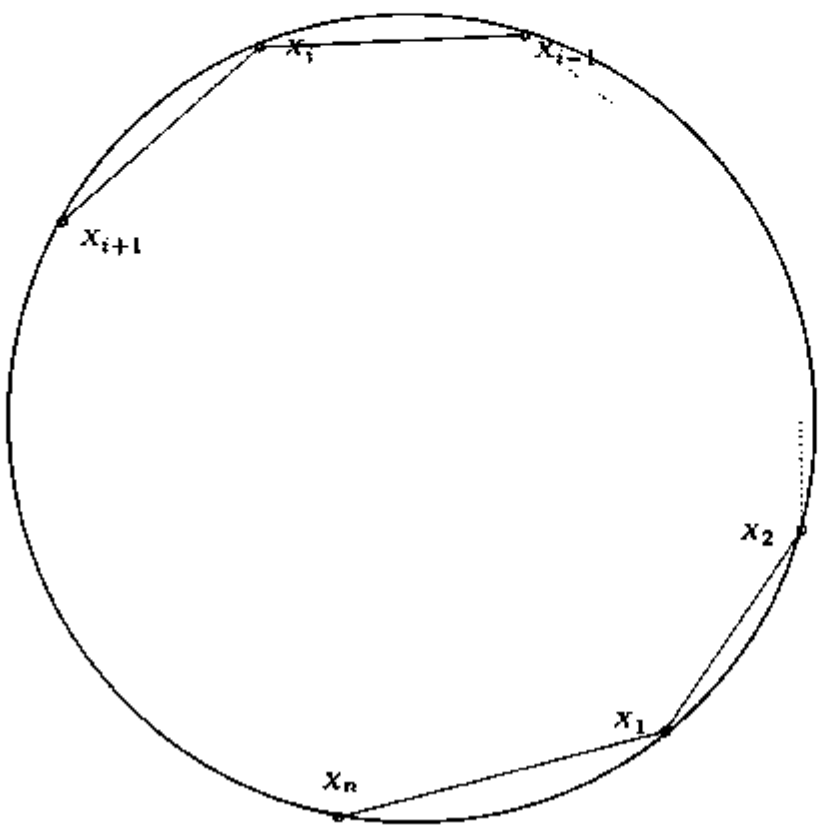


Figure 2.1. n points on a Circle.

This function f is a continuous function on a compact set, so a minimum exists. We will start the algorithm with the grid points on the surface and use algorithm (2.7) to find the constrained minimum. We will iterate until we are near the actual minimum. The end product of the Algorithm is a mapping which defines a smooth grid on the surface.

When trying to understand this algorithm we will look at surfaces that are quadratic. We feel that this is reasonable because all C^2 surfaces are locally quadratic. If the algorithm works for all quadratic surfaces then restrictions can be placed on the algorithm to make it work for non-quadratic surfaces.

CHAPTER III
PROBLEM 2-D CASE

As discussed earlier a primary objective of this paper is to create a reasonable grid on a given surface. By a reasonable grid we mean that the mesh be smooth, or the rate of change in the spacing is gradual and the angles between grid lines are not too small. The idea behind this being that, when grids are used in calculations, in most cases, the smallest error happens when the grids are smooth [Mastin;12]. The exception being that in some grids the solution of the hosted equations varies rapidly in some regions of the object: in such cases, it is reasonable to produce a finer grid on such regions, such a grid is called *solution adapted*. But our earlier generalization is more common in simple problems and is often true in more complex geometries.

Our method involves mapped surface grid generation. As we know, a mapping or transformation of a mesh from the logical space to a physical space produces the target grid on the physical space. An approach similar to Castillo's approach is to look at the parametric function.

$$f = t\sum_{i=1}^{n}\sum_{j=1}^{3}(l_{i,j})^2 + (1-t)\sum_{i=1}^{n}(\text{area}_i)^2 \qquad \text{for} \quad 0 \le t \le 1$$

where n is the number of triangles in the grid and $l_{i,j}$ is the length of the j^{th} grid segment of the i^{th} triangle; this approach gives acceptable grids in many cases where solution adaption is not required. Different grids are produced by different parametizations. We will try to minimize the same function except we will add the additional constraint that the boundary points be on the surface. For simplicity we will assume the value of t to be one, and we will look only at the surface of the two-dimensional object.

Our technique is similar to that described by Steinberg and Roach[6], and Castillo[4]. But unlike Steinberg and Roach who fixed the boundary points and optimized over the interior, we will be optimizing subject to the points being on the boundary. The minimization problem is solved by using the Augmented Lagrange method.

The mapping technique we use can best be described in the one-dimensional case. Take the simplest case for the logical space, a straight

line segment that consists of n points equally distributed on the space. Figure 3.1a. This simple structure is mapped on to a segment of the physical space. as in Figure 3.1b. so that the chosen functional is optimized for a smoother grid along the physical space.

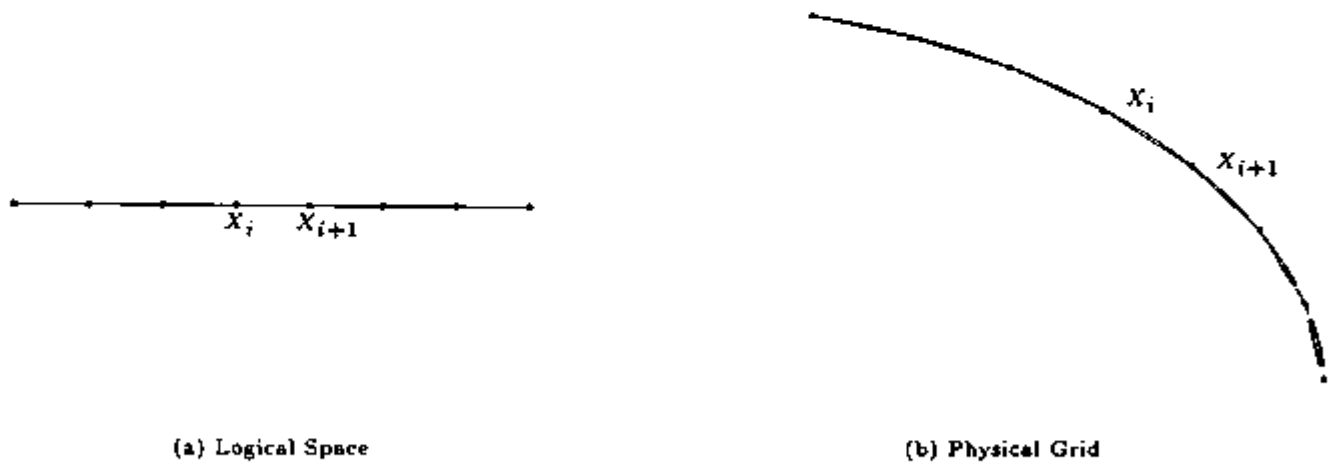


Figure 3.1. Initial mapping of the logical grid.

Once all the points of the logical grid are mapped onto or sufficiently close to the physical space then the points are moved along the surface of the physical space to smoothen the mapped grid. This task is done by minimizing a function which involves the spacing of the points. The function in this case being the sum of the squares of the grid line segments. Since this function is minimized by moving the points on the surface of the physical space a constrained minimization technique needs to be used. For this we use the Augmented Lagrange algorithm (2.7).

Now we will set up the general constrained minimization problem for two dimensions as follows. Minimize,

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 + (y_i - y_{i+1})^2$$

$$\text{subject to } \mathbf{g(x) = 0.} \tag{3.2}$$

where $\mathrm{g}_i(\mathbf{x}) = g(\mathbf{x}_i)$ where g is a function that implicitly describes the surface and $\mathbf{x} = [(x_1, y_1) \ldots (x_n, y_n)]$, $\mathrm{f} : \mathbf{x} \subset \mathbf{R^n} \rightarrow \mathbf{R}$ and $\mathbf{g} : \mathbf{x} \subset \mathbf{R^n} \rightarrow \mathbf{R^m}$.

Figure 3.1b shows the resulting grid after the mapping the points on to a segment of an ellipse where the two end points are fixed. The function $\mathbf{g(x)} = \mathbf{0}$ is an implicit description of our physical surface. Hence the problem in our algorithm consists of trying to minimize $\mathrm{f(x)}$ subject to $\mathbf{g(x)} = \mathbf{0}$.

Before going on any further in the above minimization case, we return to a general study of the constrained minimization problem. In this study we will come to understand what problems can occur in using the algorithm.

In general a constrained minimization problem is written as

$$\begin{gathered} \text{minimize } \mathrm{f(x)} \\ \text{subject to } \mathbf{g(x)} = \mathbf{0} \text{ and } \mathbf{x} \in \mathbf{\Omega} \\ \text{where } \mathbf{g} = (g_1, .., g_n). \end{gathered} \tag{3.3}$$

The first constraint $\mathbf{g(x)}$ is a functional constraint and $\mathbf{x} \in \mathbf{\Omega}$ is called a set constraint. We work under the assumption that all points of interest are in the respective region $\mathbf{\Omega}$ or $\mathbf{R^n}$ as defined later in the paper. All the points $\mathbf{x} \in \mathbf{\Omega}$ satisfying the constraints are said to be *feasible*. We now define a regular point as given in Luenberger[10].

Definition

A point $\mathbf{x^*}$ satisfying the constraint $\mathbf{g(x^*)=0}$ is said to be a *regular point* of the constraint if the gradient vectors $\nabla g_1(\mathbf{x^*})$, .. , $\nabla g_n(\mathbf{x^*})$ are linearly independent.

We also define the tangent plane to the surface defined by the constraints at a feasible point $\mathbf{x'}$ as $M_{\mathbf{x'}}$ where

$$M_{\mathbf{x'}} = \{\mathbf{y} : \nabla\, \mathrm{g(x')y} = 0\}.$$

Throughout this paper it is also assumed that f, $\mathbf{g} \in \mathrm{C}^2$. The following conditions appear as theorems in Luenberger[10].

First–Order Necessary Conditions (3.5)

Let $\mathbf{x}^*$ be a local extremum point of f subject to the constraints $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Assume further that $\mathbf{x}^*$ is a regular point of these constraints. Then there is a $\lambda \in \mathbf{R^m}$ such that

$$\nabla \mathrm{f}(\mathbf{x}^*) + \lambda^T \nabla \mathbf{g}(\mathbf{x}^*) = 0.$$

The above condition states that $\nabla \mathrm{f}(\mathbf{x}^*)$ is orthogonal to the tangent plane of the surface defined by the constraints at $\mathbf{x}^*$. We wish to define $\lambda^T \mathbf{G} = \sum_{i=1}^{m} \lambda_i \mathbf{G}_i(\mathbf{x})$.

Second – Order Necessary Conditions (3.6)

Suppose that $\mathbf{x}^*$ is a local minimum subject to $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ and that $\mathbf{x}^*$ is a regular point of these constraints. Then there is a $\lambda \in \mathbf{R^m}$ such that

$$\nabla \mathbf{f}(\mathbf{x}^*) + \lambda^T \nabla \mathbf{g}(\mathbf{x}^*) = 0$$

Then the Hessian matrix

$$\mathbf{L}(\mathbf{x}^*) = \mathbf{F}(\mathbf{x}^*) + \lambda^T \mathbf{G}(\mathbf{x}^*)$$

is positive semi-definite on $M_{\mathbf{x}^*}$, that is $\mathbf{y}^T \mathbf{L}(\mathbf{x}^*)\mathbf{y} \geq \mathbf{0}$ for all $\mathbf{y} \in M_{\mathbf{x}^*}$.

The corresponding set of sufficient conditions follows.

Second – Order Sufficiency Conditions (3.7)

Suppose that there exists a point $\mathbf{x}^*$ satisfying $\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$ and $\lambda \in \mathbf{R^m}$ such that

$$\nabla \mathbf{f}(\mathbf{x}^*) + \lambda^T \nabla \mathbf{g}(\mathbf{x}^*) = 0$$

suppose also that the matrix

$$\mathbf{L}(\mathbf{x}^*) = \mathbf{F}(\mathbf{x}^*) + \lambda^T \mathbf{G}(\mathbf{x}^*) \tag{3.8}$$

is positive definite on M. That is for $\mathbf{y} \neq 0$ $\mathbf{y}^T \mathbf{L}(\mathbf{x}^*)\mathbf{y} > \mathbf{0}$. Then $\mathbf{x}^*$ is a strict local minimum of f subject to $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.

We wish to note that, we will in our case studies. refer to the matrix (3.8) as H_L. This is the matrix of second partial derivatives with respect to $\mathbf{x}$ of the Lagrangian $l(\mathbf{x}, \lambda) = \mathbf{f}(\mathbf{x}) + \lambda^{\mathbf{T}}\mathbf{g}(\mathbf{x})$. Also the Hessian of $\mathcal{L}(\mathbf{x}.\lambda.\mathbf{P})$ (2.6) will often be referred to as $H_{\mathcal{L}}$ (3.9).

We will now look at the relevance of the above conditions to our algorithm. The minimization techniques are less likely to work well if the problem does not have a strict local minimum. So we are interested in the conditions under which problems similar to problem (2.8) have strict local minima.

Therefore showing the Hessian matrix $H_{\mathcal{L}}$ of $\mathcal{L}$ is positive definite at a minimum will prove that the minimum is a strict local minimum for f. Therefore it is essential for us to understand the matrix

$$H_{\mathcal{L}} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial x_1} & \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial x_n} & \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial y_1} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial y_n} \\ \frac{\partial^2 \mathcal{L}}{\partial x_2 \partial x_1} & & & & & & \frac{\partial^2 \mathcal{L}}{\partial x_2 \partial y_n} \\ \vdots & \ddots & & & & & \vdots \\ \vdots & & & & & & \vdots \\ \frac{\partial^2 \mathcal{L}}{\partial x_n \partial x_1} & & & \frac{\partial^2 \mathcal{L}}{\partial x_n \partial x_n} & & & \vdots \\ \frac{\partial^2 \mathcal{L}}{\partial y_1 \partial x_1} & & & & & & \\ \vdots & & & & \ddots & & \vdots \\ \vdots & \cdots & & & & & \vdots \\ \frac{\partial^2 \mathcal{L}}{\partial y_n \partial x_1} & \frac{\partial^2 \mathcal{L}}{\partial y_n \partial x_2} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial y_n \partial x_n} & \frac{\partial^2 \mathcal{L}}{\partial y_n \partial y_1} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial y_n \partial y_n} \end{bmatrix}. \qquad (3.9)$$

In what follows we examine H_L in those cases where the function that describes the surface is quadratic. The second-order sufficiency condition requires that $H_{\mathcal{L}}$ be positive definite in the tangent plane, but we instead check to see if $H_{\mathcal{L}}$ is positive definite in all of $\mathbf{R^n}$.

We begin our examples with the simple case of three points. on the boundary of the first quadrant, of a unit circle. First as in Figure 3.2(below) we define f and g as follows,

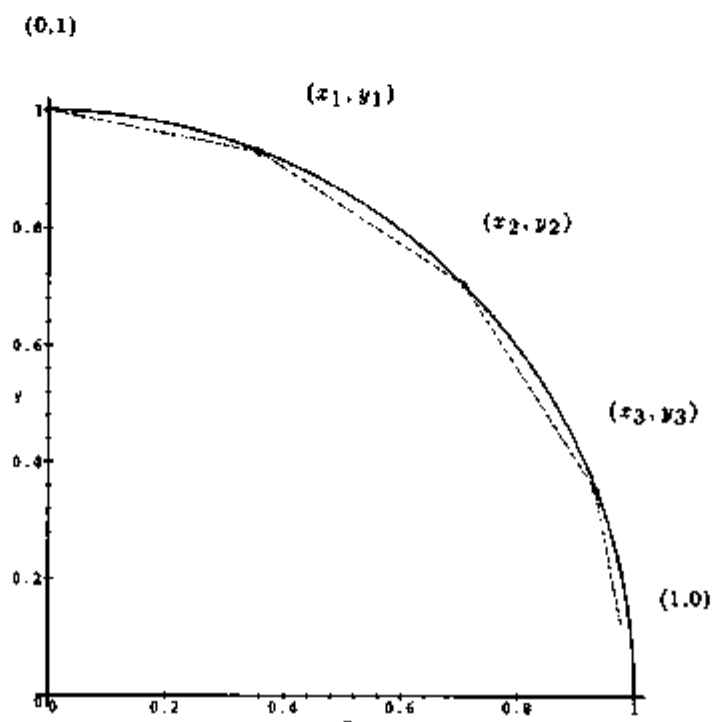


Figure 3.2. Three points on Unit Circle.

$$f = (0-x_1)^2 + (1-y_1)^2 + (x_1-x_2)^2 + (y_1-y_2)^2+(x_2-x_3)^2 + (y_2-y_3)^2+(x_3-1)^2 + (y_3-0)^2$$

and

$$g_i = x_i^2 + y_i^2 - 1$$

In this example the first point is attached to the point (0,1) and the last point is attached to the point (1,0). The Lagrange function L is.

$$L = (0-x_1)^2 + (1-y_1)^2 + (x_1-x_2)^2 + (y_1-y_2)^2+(x_2-x_3)^2 + (y_2-y_3)^2+(x_3-1)^2 + (y_3-0)^2+\lambda_1(x_1^2+y_1^2-1) + \lambda_2(x_2^2+y_2^2-1) + \lambda_3(x_3^2+y_3^2-1) \tag{3.10.1}$$

When $\nabla L = 0$.

$$
\begin{aligned}
\frac{\partial L}{\partial x_1} &= 4x_1 - 2x_2 + \lambda_1 x_1 = 0 \\
\frac{\partial L}{\partial x_2} &= -2x_1 + 4x_2 - 2x_3 + 2\lambda_2 x_2 = 0 \\
\frac{\partial L}{\partial x_3} &= -2x_2 + 4x_3 - 2 + 2\lambda_3 x_3 = 0
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial y_1} &= -2 + 4y_1 - 2y_2 + 2\lambda_1 y_1 = 0 \\
\frac{\partial L}{\partial y_2} &= -2y_1 + 4y_2 - 2y_3 + 2\lambda_2 y_2 = 0 \\
\frac{\partial L}{\partial y_3} &= -2y_2 + 4y_3 + 2\lambda_3 y_3 = 0
\end{aligned}
$$

Since the points are on the boundary $x_i^2 + y_i^2 - 1 = 0$ for all three points. Solving which gives us

$$(x_1, y_1) = \left(\frac{\sqrt{2-\sqrt{2}}}{2}, \frac{\sqrt{2+\sqrt{2}}}{2} \right)$$

$$(x_2, y_2) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)$$

$$(x_3, y_3) = \left(\frac{\sqrt{2+\sqrt{2}}}{2}, \frac{\sqrt{2-\sqrt{2}}}{2} \right).$$

Since we are on the circle $\lambda = \lambda_1 = \lambda_2 = \lambda_3$.

Then

$$\lambda = \left(\sqrt{\frac{2}{(2-\sqrt{2})}} - 2 \right).$$

We get the Hessian as

$$
H_L = \begin{bmatrix}
4+2\lambda & -2 & 0 & 0 & 0 & 0 \\
-2 & 4+2\lambda & -2 & 0 & 0 & 0 \\
0 & -2 & 4+2\lambda & 0 & 0 & 0 \\
0 & 0 & 0 & 4+2\lambda & -2 & 0 \\
0 & 0 & 0 & -2 & 4+2\lambda & -2 \\
0 & 0 & 0 & 0 & -2 & 4+2\lambda
\end{bmatrix}.
$$

The Hessian is a sparse matrix of size 2n×2n where n is the number of free points.

For this case, we get three different eigenvalues for H_L which are

$$4 + 2\lambda \quad , \quad 4 + 2\lambda + 2\sqrt{2} \quad , \quad 4 + 2\lambda - 2\sqrt{2}.$$

When we plug in λ and find the eigenvalues the smallest is

$$\left(4 + 2\left(\sqrt{\frac{2}{(2-\sqrt{2})}} - 2\right) - 2\sqrt{2}\right) > 0.$$

Therefore the Hessian is Positive definite.

Before moving on to other cases let us examine the way each case will be analyzed. First, we are working under the assumption of the existence of a relative minimum $\mathbf{x}^*$ as defined in the previous section. Then from the first-order necessary conditions (3.5) we know about the existence of λ^* as in (2.4). Therefore from the second-order sufficiency conditions (3.7), if we show that the Hessian matrix (3.9) for the case is positive definite for the given points; we will have reason to believe that the algorithm will work reasonably well. Therefore in most of our case studies we will stop the analysis after showing the relevant Hessian is positive definite.

Now we move on to the case where there n points on disk of radius K. In this case we can guess that the solution is such that each point is equally spaced on the surface, and that each Lagrange multiplier is the same.
Figure 3.3 shows a partial view of the grid structure when g= $x^2 + y^2 - K^2$.

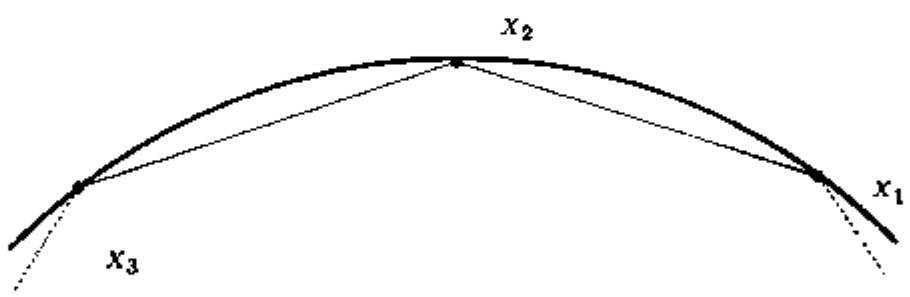


Figure 3.3. A Circle with n points.

For this situation of n points with coordinates of $X_i = (x_i, y_i)$.

$$f = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (x_2 - x_3)^2 + (y_2 - y_3)^2 + \Sigma(\text{other segments})^2 \tag{3.11}$$

Let us pay attention to point X_2. (Note $\lambda = \lambda_2$).

$$L = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (x_2 - x_3)^2 + (y_2 - y_3)^2 + \lambda(x_2^2 + y_2^2 - K^2)$$

$$+ \Sigma_{i \neq 2} \lambda_i g(X_i).$$

Now at the minimum the gradient $\frac{\partial L}{\partial x_2} = 0$, so

$$-2x_1 - 2x_3 + 4x_2 + 2\lambda x_2 = 0.$$

Then

$$\lambda = \frac{2x_1 - 4x_2 + 2x_3}{2x_2}. \tag{3.12}$$

Similarly we could find λ by using the fact that $\frac{\partial L}{\partial y_2} = 0$. If

$$x_2 = K \quad \cos(\hat{\theta})$$

then

$$x_1 = K \quad \cos(\hat{\theta} - \theta)$$

and

$$x_3 = K \quad \cos(\hat{\theta} + \theta).$$

Where θ is the angular distance between the points. It is clear that the function does not have a unique minimum if all of the points are free to move. So we *fix* some of the points, and concentrate only on a segment between two fixed points. Using this information we see that $\lambda = 2\cos(\theta) - 2$. On a circle $\lambda = \lambda_i$ for $1 \leq i \leq$ n. In this case the Hessian matrix is as follows.

$$H_L = \begin{bmatrix}
4+2\lambda & -2 & 0 & 0 & 0 & \cdots & & & 0 \\
-2 & 4+2\lambda & -2 & 0 & 0 & \cdots & & & 0 \\
\vdots & & & & & & & & \vdots \\
\vdots & & \ddots & -2 & 0 & \cdots & & & \vdots \\
0 & 0 & -2 & 4+2\lambda & 0 & \cdots & & & 0 \\
\vdots & & & 0 & 4+2\lambda & -2 & 0 & \cdots & 0 \\
\vdots & & & 0 & -2 & 4+2\lambda & & & \vdots \\
\vdots & & & & & & & & \vdots \\
\vdots & & & & & & \ddots & & \vdots \\
0 & & & \cdots & & 0 & -2 & 4+2\lambda & -2 \\
0 & & & \cdots & & 0 & 0 & -2 & 4+2\lambda
\end{bmatrix}$$

It is known that the minimum eigenvalues of a matrix of this form is ([14]).

$$\text{eigenvalues} \geq 4+2\lambda - 4\cos(\theta_j) \qquad \text{where } \theta_j = \tfrac{j\pi}{n+1}.$$

Using the function $\lambda = 2\cos(\theta) - 2$ we get that the minimum eigenvalue is

$$\begin{aligned} \text{eigenvalue} &= 4 + 2(2\cos(\theta) - 2) - 4\cos(\theta_1) \\ &= 4\cos(\theta) - 4\cos(\theta_1) \end{aligned}$$

Where θ depends on the angular distance between the two fixed points. For example if the distance between the two fixed points is $\frac{\pi}{2}$ then. $\theta = \frac{\pi}{2n}$ and $\theta_1 = \frac{1\pi}{n+1}$. Showing that

$$\cos\left(\frac{\pi}{2n}\right) > \cos\left(\frac{j\pi}{n+1}\right)$$

will complete the analysis. But 2n > n+1 for all n>2. This makes the above inequality true, because cosine is a decreasing function near 0. Note that the angle of the surface segment can be defined as large as $0 < \theta < \pi$. and still have the minimum eigenvalue positive. The smaller the θ the larger the minimum eigenvalue.

Next we consider the case where the physical surface is an ellipse. In this case the implicit function that describes the surface would look like,

$$g = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0. \tag{3.13}$$

or parametrically for $a > 0$, $b > 0$

$$x_i = a\cos(\theta_i),\ y_i = b\sin(\theta_i) \qquad \text{where } \theta_i = \left(\pi - \frac{2i}{n}\right) \text{ and } 1 \le i \le n.$$

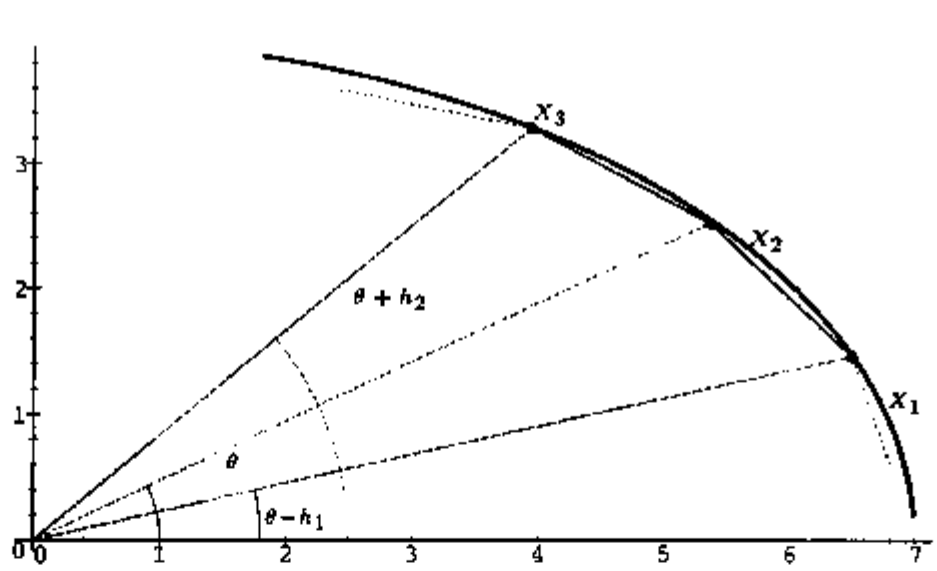


Figure 3.4. n points on the Ellipse.

Then as in Figure 3.4 we isolate three points and move the middle point, subject to constraints of the surface, to minimize the function f as in (3.11). Since the physical surface here is an ellipse we get L as.

$$L = f + \lambda_2\left(\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1\right) + \sum_{i \neq 2}\lambda_i g(X_i)$$

Considering the gradient $\frac{\partial L}{\partial x_2} = 0$, we get ($\lambda = \lambda_2$.)

$$\lambda = \frac{a^2(2x_1 - 4x_2 + 2x_3)}{2x_2}. \tag{3.15}$$

For $\frac{\partial L}{\partial y_2} = 0$, we get

$$\lambda = \frac{b^2(2y_1 - 4y_2 + 2y_3)}{2y_2}. \tag{3.16}$$

Then looking at the three points on the surface as in Figure (above) named X_1, X_2 and X_3. We parameterize X_1 and X_3 in terms of X_2 as follows.

$$\begin{aligned}
X_2 = (x_2, y_2) &= \Big(\big(\text{a } \cos(\pi - \text{t}) \big) , \big(\text{b } \sin(\pi - \text{t}) \big) \Big) \\
X_1 = (x_1, y_1) &= \Big(\big(\text{a } \cos(\pi - [\text{t} - h_1]) \big) , \big(\text{b } \sin(\pi - [\text{t} - h_1]) \big) \Big) \\
X_3 = (x_3, y_3) &= \Big(\big(\text{a } \cos(\pi - [\text{t} + h_2]) \big) , \big(\text{b } \sin(\pi - [\text{t} + h_2]) \big) \Big)
\end{aligned}$$

where $h_1, h_2, t \in [0, 2\pi]$. Substituting each case to (3.15) and (3.16) respectively

$$\lambda = a^2 \frac{\cos(t - h_1) + \cos(t + h_2)}{\cos(t)} - 2a^2. \tag{3.17}$$

Equation (3.17) gives λ in terms of X-coordinates, given below is λ in terms of Y-coordinates.

$$\lambda = b^2 \frac{\sin(t - h_1) + \sin(t + h_2)}{\sin(t)} - 2a^2 \tag{3.18}$$

The same analysis applies for each λ_i. Our Hessian matrix for the case looks as below,

$$H_L = \begin{bmatrix}
4+\frac{2\lambda_1}{a^2} & -2 & 0 & 0 & \cdots & & & 0 \\
-2 & 4+\frac{2\lambda_2}{a^2} & -2 & 0 & \cdots & & & 0 \\
0 & & & & \cdots & & & 0 \\
\vdots & & & & & & & \vdots \\
\vdots & & \ddots & -2 & & & & \vdots \\
\vdots & & -2 & 4+\frac{2\lambda_n}{a^2} & 0 & & & \vdots \\
\vdots & & & 0 & 4+\frac{2\lambda_1}{b^2} & -2 & & \\
\vdots & & & & -2 & \ddots & & 0 \\
\vdots & & & & & & & \vdots \\
\vdots & & & & & & & 0 \\
0 & \cdots & & 0 & 0 & -2 & 4+\frac{2\lambda_{n-1}}{b^2} & -2 \\
0 & \cdots & & 0 & 0 & 0 & -2 & 4+\frac{2\lambda_n}{b^2}
\end{bmatrix}$$

Then using the equations (3.17) and (3.18) the smallest eigenvalue is greater that or equal to either

$$4 + \frac{2\lambda'}{a^2} - 4\cos(\theta_1) \tag{3.19}$$

or

$$4 + \frac{2\lambda'}{b^2} - 4\cos(\theta_1). \tag{3.20}$$

Where λ' is the smallest of the Lagrange multipliers. Substituting (3.17) for the first equation gives the smallest eigenvalue as

$$\text{eigenvalue} \geq 4 + \frac{2a^2}{a^2}\frac{\cos(t-h_1)+\cos(t+h_2)}{\cos(t)} - 4 - 4\cos(\theta_1)$$

$$= 2\,\frac{(\cos(t-h_1)+\cos(t+h_2))}{\cos(t)-4\cos(\theta_1).}$$

The number θ_1 depends on the number of free points between two fixed points. Also h_1 and h_2 both get smaller as the angular distance between the fixed

points is reduced. Therefore we can ensure that the minimum eigenvalue is positive by having the angular distance between the fixed points small enough.

Next we move on to the case when our physical surface is a parabola as in Figure 3.5.

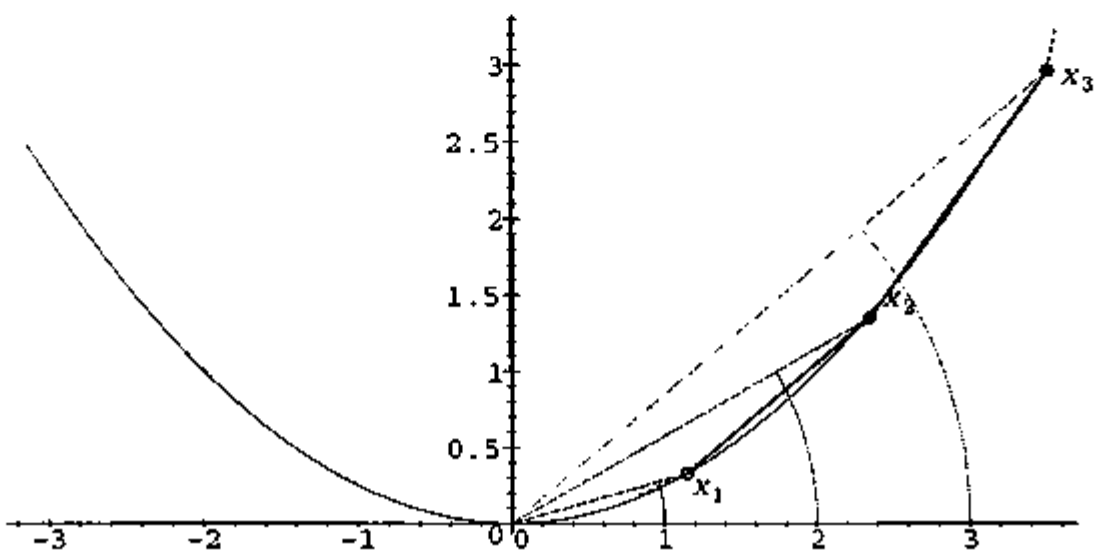


Figure 3.5. n points on the Parabola.

Then

$$g = y - ax^2.$$

Since the notations are similar to the previous cases the function f can be defined as in (3.11) as

$$f = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (x_2 - x_3)^2 + (y_2 - y_3)^2 + \Sigma(\text{other segments})^2$$

then (when $\lambda = \lambda_2$),

$$L = f + \lambda(y_2 - ax_2^2) + \sum_{i \neq 2} \lambda_i g(X_i).$$

Considering the gradient $\frac{\partial L}{\partial x_2} = 0$, we get

$$\lambda = \frac{2x_2 - x_1 - x_3}{ax_2} = \frac{2}{a} - \frac{(x_1 + x_3)}{ax_2}$$

The same analysis applies for each λ_i. The Hessian for the case is.

$$H_L = \begin{bmatrix} 4-2a\lambda_1 & -2 & 0 & 0 & \cdots & & 0 \\ -2 & 4-2a\lambda_2 & -2 & 0 & \cdots & & 0 \\ 0 & & & & \cdots & & 0 \\ \vdots & \ddots & & & & & \vdots \\ \vdots & -2 & 4-2a\lambda_n & 0 & & & \vdots \\ \vdots & & 0 & 4 & -2 & & \\ \vdots & & & & \ddots & & 0 \\ \vdots & & & & & & 0 \\ 0 & \cdots & & 0 & -2 & 4 & -2 \\ 0 & \cdots & & 0 & 0 & -2 & 4 \end{bmatrix}.$$

With this H_L the lower right hand block is a positive definite matrix. If the minimum of H_L is to be negative, then the minimum eigenvalue of the upper left hand block must be negative. The minimum eigenvalue of this block is

$$\frac{2(x_1+x_3)}{x_2} - 4\cos(\theta_1).$$

Again θ_1 depends on the number of points and the points X_1 and X_3 approach X_2 as the surface distance between the fixed points is reduced. For this reason we believe that we can make H_L to be positive definite by reducing the distance between fixed points.

Finally we take the physical space to be a hyperbola as in Figure 3.6.

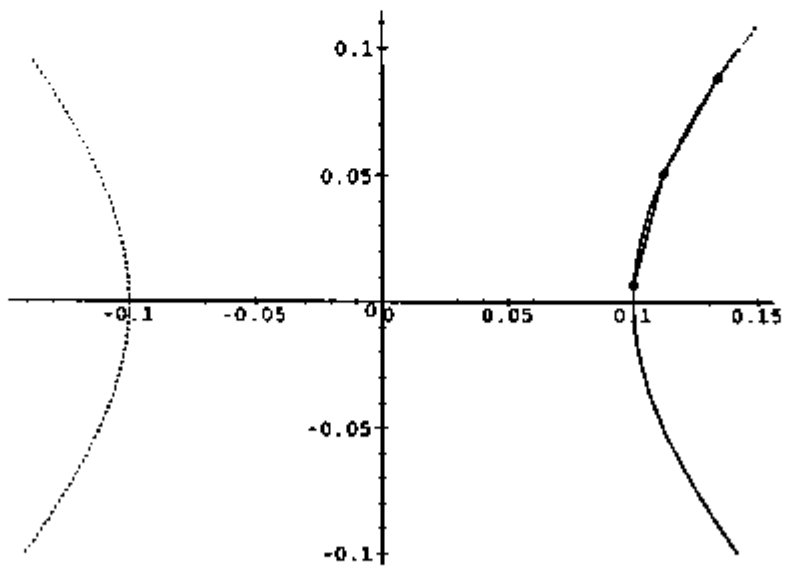


Figure 3.6. n points on the Hyperbola.

Now $$g = \frac{x^2}{a^2} - \frac{y^2}{b^2} - K = 0.$$

Then as in the previous cases

$$L = f + \lambda\left(\frac{x_2^2}{a^2} - \frac{y_2^2}{b^2} - K\right) + \Sigma_{i\neq 2}\lambda_i g(N_i).$$

Considering the gradient $\frac{\partial L}{\partial x_2} = 0$, we get

$$\lambda = \frac{a^2(x_1 - 2x_2 + x_3)}{x_2}. \tag{3.21}$$

For $\frac{\partial L}{\partial y_2} = 0$, we get

$$\lambda = \frac{-b^2(y_1 - 2y_2 + y_3)}{y_2}. \tag{3.22}$$

The same analysis applies to each λ_i. The Hessian matrix is.

$$H_L = \begin{bmatrix} 4+\frac{2\lambda_1}{a^2} & -2 & 0 & 0 & \cdots & & 0 \\ -2 & 4+\frac{2\lambda_2}{a^2} & -2 & 0 & \cdots & & 0 \\ 0 & & & & \cdots & & 0 \\ \vdots & & \ddots & & & & \vdots \\ \vdots & & -2 & 4+\frac{2\lambda_n}{a^2} & & & \vdots \\ \vdots & & & & 4-\frac{2\lambda_1}{b^2} & -2 & \\ \vdots & & & & \ddots & & 0 \\ \vdots & & & & & & 0 \\ 0 & \cdots & & 0 & -2 & 4-\frac{2\lambda_{n-1}}{b^2} & -2 \\ 0 & \cdots & & 0 & 0 & -2 & 4-\frac{2\lambda_n}{b^2} \end{bmatrix} \quad (3.22.1)$$

and as in the previous cases we see that the minimum eigenvalue is

$$4+2\frac{x_1-2x_2+x_3}{x_2}-4\cos(\theta_1) \quad (3.23)$$

or, in terms of y we get

$$4+2\frac{y_1-2y_2+y_3}{y_2}-4\cos(\theta_1). \quad (3.24)$$

Using (3.23) we would get that the minimum eigenvalue is

$$\frac{2x_1+2x_3}{x_2}-4\cos(\theta_j). \quad (3.23.1)$$

Similarly, in terms of y, we would get the minimum eigenvalue as

$$\frac{2y_1+2y_3}{y_2}-4\cos(\theta_j). \quad (3.24.1)$$

Again the surface distance between fixed points can be chosen to insure that the minimum eigenvalue is not too small.

Before we go on to the numerical examples, we wish to note that contrary to what our intuition might say the grid segments are not necessarily equidistant at the minimum. Take for example the case below,

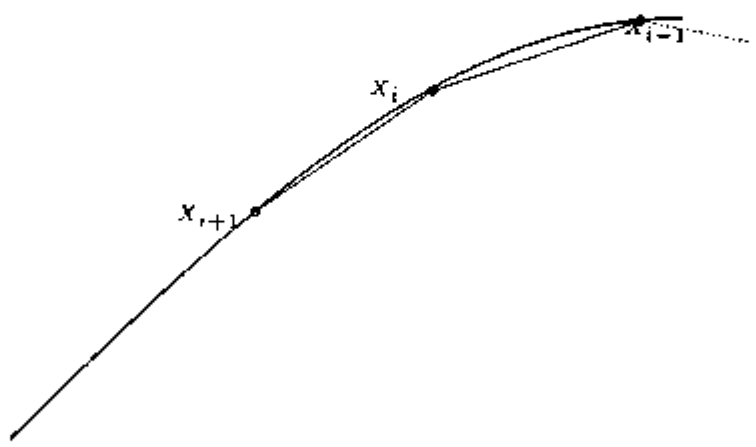


Figure 3.7. Three points on a known surface y=h(x).

Here $X_i = (x_i, y_i)$. If the distances are equal then,

$$(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2.$$

After simplification we get

$$-2x_i + (x_{i+1} + x_i) = \left(\frac{y_{i-1} - y_{i+1}}{x_{i-1} - x_{i+1}}\right)(2y_i - (y_{i-1} + y_{i+1})) \tag{3.25}$$

If we minimize the function

$$f = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$$

at the minimum $\frac{\partial f}{\partial x_i} = 0$. Then since $h(x_i) = y_i$

$$-2x_i + (x_{i+1} + x_i) = h'(x_2)\,(2y_i - (y_{i-1} + y_{i+1})) \tag{3.26}$$

We can see the similarity between (3.25) and (3.26). Here $\left(\frac{y_{i-1} - y_{i+1}}{x_{i-1} - x_{i+1}}\right)$ approaches $h'(x_i)$ as the distances become smaller between the three points. However at the minimum of our algorithm, the distances are not always equal. The first example (next chapter) contains a table showing the distances that we obtained at the minimum.

CHAPTER IV
NUMERICAL EXAMPLES

In this section we will test our algorithm on similar surfaces using computer code that was implemented in Maple V on a SUN 670 workstation.

First we begin by defining our physical surface as

$$g = x_i^2 + \frac{y_i^2}{4} - 1 = 0.$$

Given below are our initial points as in (2.7).(Note that we have fixed $(.1, \frac{\sqrt{99}}{5}), (.9, \frac{\sqrt{19}}{5})$).

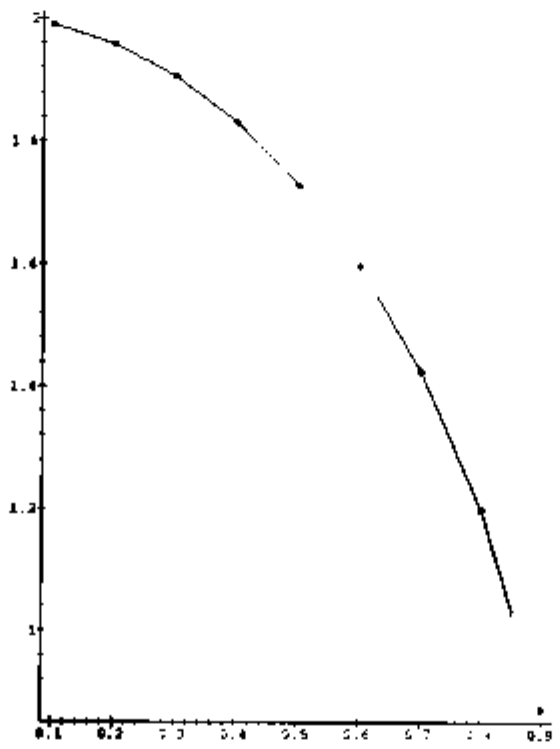


Figure 4.1. The initial grid.

The table below shows the behavior of the gradient $\| \nabla L \|^2$ as we iterate in minimizing the Lagrangian function [step 2a(2.7)].

Table 4.1. Minimizing L for fixed penalty.

Iteration	$\| \nabla L \|^2$	Penalty	Iteration	$\| \nabla L \|^2$	Penalty
1	1.142284957	25	1	0.9346654762	125
5	0.1184123433		5	0.09239369265	
10	0.7389013714		10	0.07691514344	
15	0.008343233481		15	0.001897722418	

The following table shows the average distance between the points and the

surface on the i^{th} iteration.

Table 4.2. Average displacement of the points.

Iteration	$\sum g(x) / n$	Penalty
1	-.01805089307	1
2	-0.004215467477	5
3	-0.0006243653222	25
4	-0.0001987211555	125
5	-.0000249971888	625

The Figure 4.2 shows the placement of the points obtained by our algorithm.

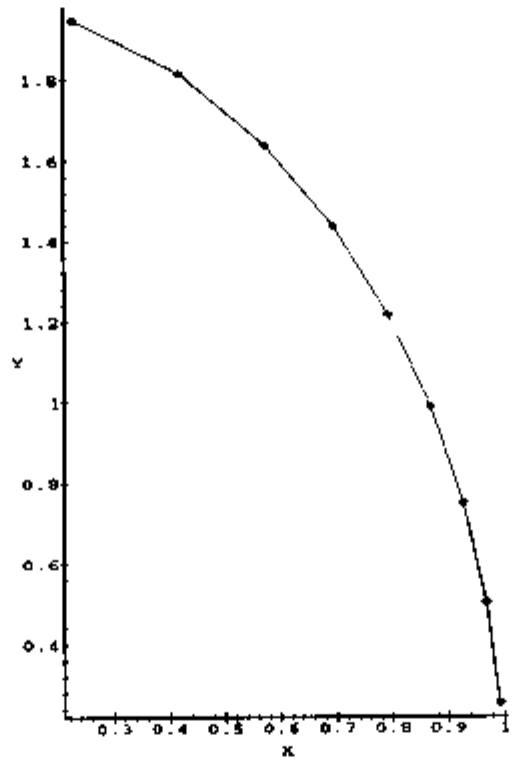


Figure 4.2. Grid after 6 iterations.

Table 4.3. The lengths of the i^{th} segment.

Segment	Length
1	0.1811
2	0.1804
3	0.1800
4	0.1799
5	0.1798
6	0.1797
7	0.1797
8	0.1797

In the next example the surface is the sine curve through 0 to 2π. Then we define the physical surface as

$$g = y_i - \sin(x_i) = 0.$$

Given below is the initial grid. (Note we fixed the points (0,0) $(0,2\pi)$).

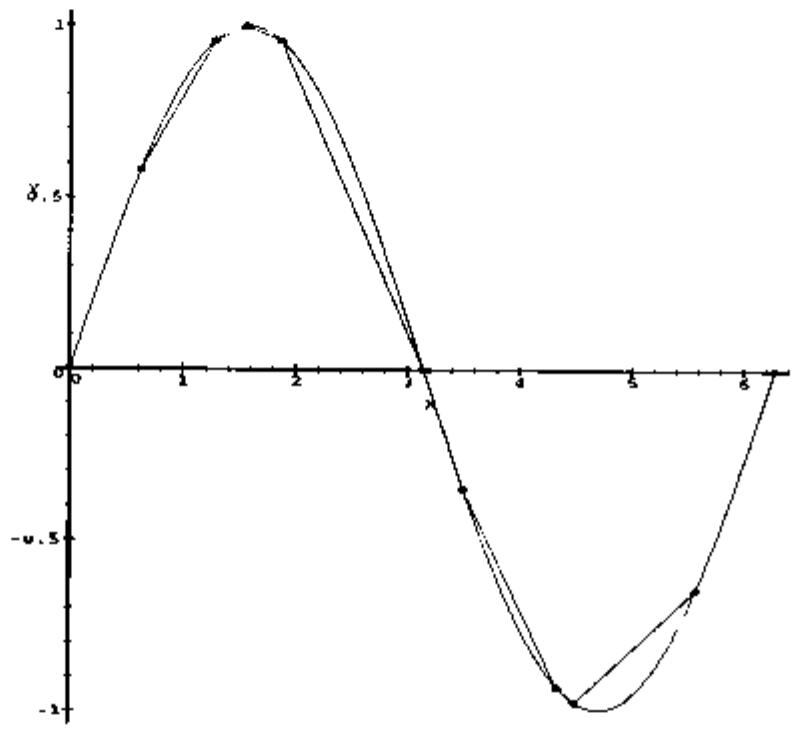


Figure 4.3. The initial grid.

Table given below shows the behavior of the gradient $\| \nabla L \|^2$ as we iterate in minimizing the Lagrangian function [step 2a(2.7)].

Table 4.4. Minimizing L for fixed penalty.

Iteration	$\| \nabla L \|^2$	Penalty	Iteration	$\| \nabla L \|^2$	Penalty
1	6.646376398	25	1	0.5570856204	125
5	0.004618772499		5	0.005495474750	
10	0.003429821033		10	0.002166818604	
15	0.002429035331		15	0.002111973939	

The following table shows the average distance between the points and the surface on the i^{th} iteration.

Table 4.5. Average displacement of the points.

Iteration	$\sum g(\mathbf{x}) / n$	Penalty
1	0.0365520230	1
2	0.00106757873	5
3	-0.00034793848	25
4	-0.00003674037125	125
5	-.00001644358	625

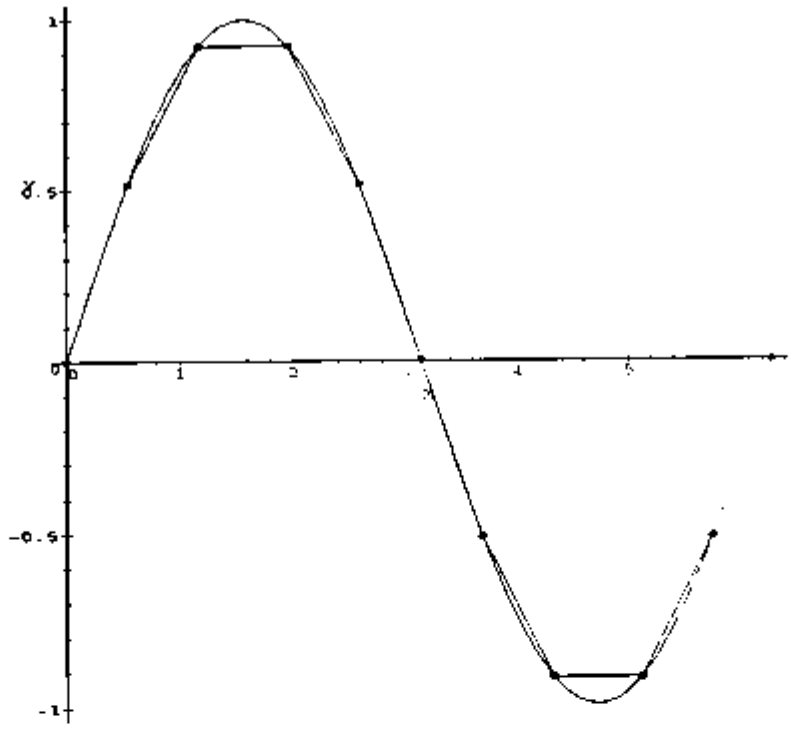


Figure 4.4. Grid after 6 iterations.

In the next example the surface is the Parabola . Then we define the physical surface as

$$g = y_i - x_i^2 = 0.$$

Given below is the initial grid. (Note we fixed the points (-2.4) (0,0)).

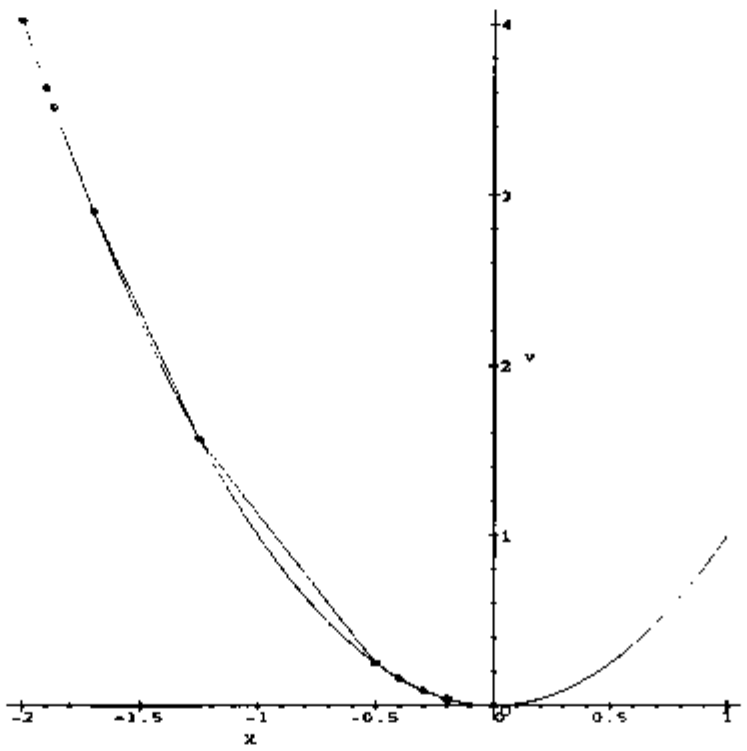


Figure 4.5. The initial grid.

Table given below shows the behavior of the gradient $\| \nabla L \|^2$ as we iterate in minimizing the Lagrangian function [step 2a(2.7)].

Table 4.6. Minimizing L for fixed penalty.

Iteration	$\| \nabla L \|^2$	Penalty	Iteration	$\| \nabla L \|^2$	Penalty
1	0.1655468302	25	1	0.3904269454	125
5	0.0014537212		5	0.0009001418	
10	0.0002657323		10	0.0004290574	

The following table shows the average distance between the points and the surface on the i^{th} iteration.

Table 4.7. Average displacement of the points.

Iteration	$\sum g(\mathbf{x}) \ /n$	Penalty
1	0.18740276496	1
2	0.002614837	5
3	0.000126266	25
4	0.0000020375	125
5	-.000000404366	625

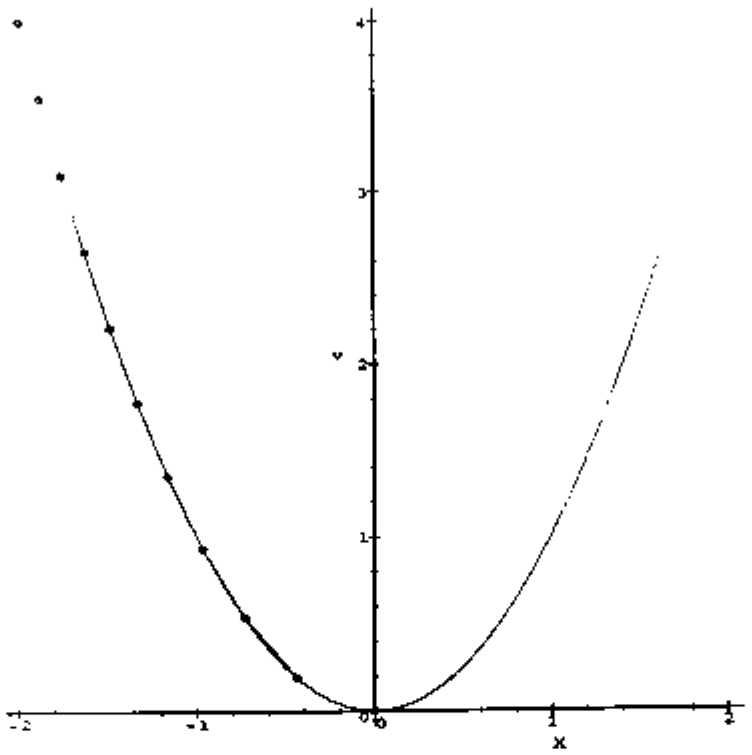


Figure 4.6. Grid after 6 iterations.

Finally we take the case when the surface is a Hyperbola. Then we define the physical surface as

$$g = x_i^2 - \frac{y_i^4}{4} - 1 = 0.$$

Given below is the initial grid. (Note we fixed the points (1,0) $(2.2\sqrt{3})$).

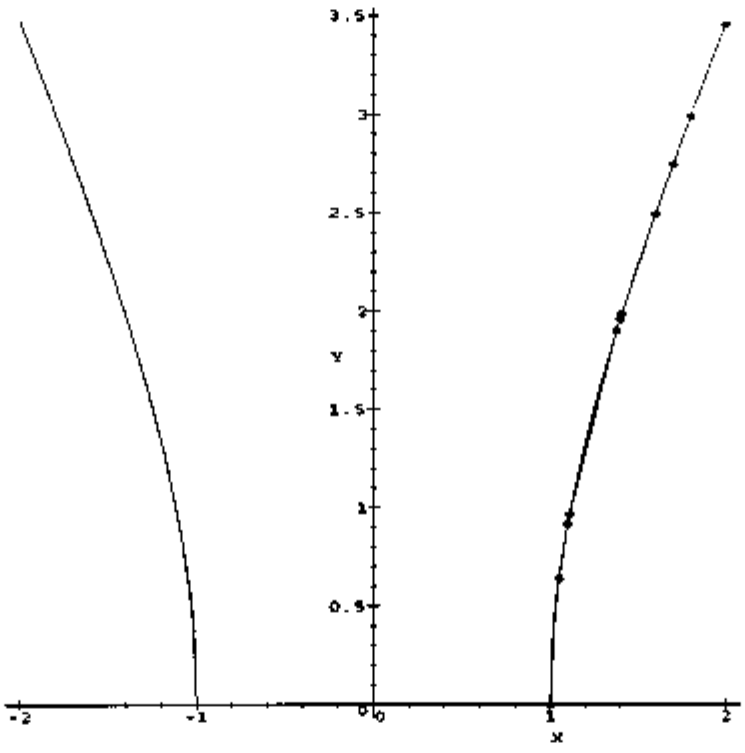


Figure 4.7. The initial grid.

Table given below shows the behavior of the gradient $\| \nabla L \|^2$ as we iterate in minimizing the Lagrangian function [step 2a(2.7)].

Table 4.8. Minimizing L for fixed penalty.

Iteration	$\| \nabla L \|^2$	Penalty	Iteration	$\| \nabla L \|^2$	Penalty
1	0..1251075910	25	1	0.002461377394	125
5	0.004927934566		5	0.002360835844	
10	0.00068154875		10	0.0006736483365	

The following table shows the average distance between the points and the surface on the i^{th} iteration.

Table 4.9. Average displacement of the points.

Iteration	$\sum g(\mathbf{x}) / n$	Penalty
1	0.013846563	1
2	0.000485296	5
3	0.000022680	25
4	-0.000006321	125
5	-.000000294	625

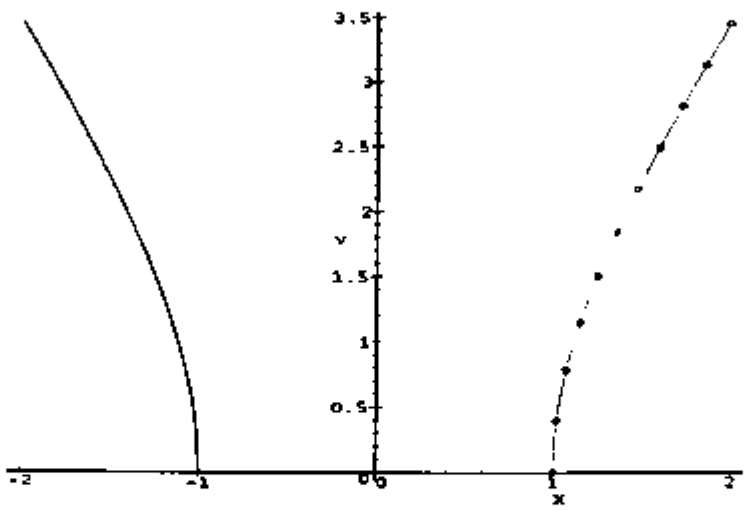


Figure 4.8. Grid after 6 iterations.

CHAPTER V
EXTENSION TO THREE DIMENSIONS AND CONCLUSION

The goal of this thesis was to gain an understanding of the issues related to generating a three-dimensional surface grid by studying a two-dimensional surface. In three dimensions, the technique would consist of composing a triangular grid on some reference grid and mapping it to the target surface via Augmented Lagrange techniques as was done in two dimensions. The issues in three-dimensions will be very similar to the two-dimensional issues. We would minimize a function of the form

$$f = t\sum_{i=1}^{n}\sum_{j=1}^{3}(l_{i,j})^2 + (1-t)\sum_{i=1}^{n}(\text{area}_i)^2 \qquad \text{for} \quad 0 \le t \le 1$$

subject to $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.

We would examine the Hessian matrix H_L to see what the smallest eigenvalue would be. The Hessian matrix would again be a sparse matrix with one block for each of the x, y, and z components. We expect that the smallest eigenvalue would be positive if a number of the points on the surface are not fixed. Studying the problem in three dimensions would be much harder than in two dimensions because the Hessian matrix would not necessarily have a structure for which the eigenvalues are known. To study the three-dimensional case, we would have to resort to using some numerical approximation for the eigenvalues of the Hessian.

We believe that we have, however gained sufficient understanding to try the three-dimensional algorithm. From our two-dimensional work, we see that we can expect that if try to minimize f with a large number of free points spread over a large surface area the algorithm could converge slowly. To get around this problem we might try to initially use a small number of points on the surface and minimize f. At this time we could fix those points and add more points to refine the grid. Trying this technique and comparing the resulting convergence speed can be the subject of future research.

BIBLIOGRAPHY

[1] Thompson,J.F..Warsi.Z.U.A..and Mastin,C.W..(1985).*Numerical Grid Generation: Foundations and Applications*,North-Holland,Elsevier.NY.

[2] Nunn,C.J.,*A Non-Folding Discrete Grid Generation Method.*,Paper submit ted for publication to Communications in Numerical Methods in Engineering.

[3] Kennon.S.R..and Dulikravich,G.S..(1986),Generation of Computational Grids Using optimization..*AIAA Journal* 24(7).1063-1073.

[4] Castillo.J.E..(1991).A Discrete Variational Grid Generation Method..*SIAM Journal Sci. Stat. Comput.* 12,454-468.

[5] Steinberg,S..and Roache,P..(1989).Variational Curve and Surface Grid Generation.,*Journal of Computational Physics*,163-178.

[6] Castillo,J,E..Steinberg,S.,and Roache,P.J.,(1988),Parameter Estimation in Variational Grid Generation,*Applied Math. Comp.*,28,155-178.

[7] Castillo.J.E.,(1991),*Mathematical Aspects of Numerical Grid Generation*.. SIAM.Philadelphia.

[8] Knupp,P.,and Steinberg,S..(1993).*Fundamentals of Grid Generation*..CRC Press ,Inc.,Boca Baton,FL.

[9] Nunn,C.J.,(1993),*Numerical Algorithms for Liquid Crystal Droplet Problems.*,Dissertation,University of North Carolina at Chapel Hill.

[10] Luenberger,D.G.,*Linear and Nonlinear Programming*,Addison Wesley,Reading.MA,1989.

[11] Lancaster,P..and Salkauskas,K.,(1990).*Curve and Surface Fitting*.Academic Press.San Diego.CA.

[12] Mastin.C.W..(1982).*Error induced by coordinate systems in Numerical Grid Generation*.,J.F.Thompson.ed.,pp31-40,North-Holland,NY.

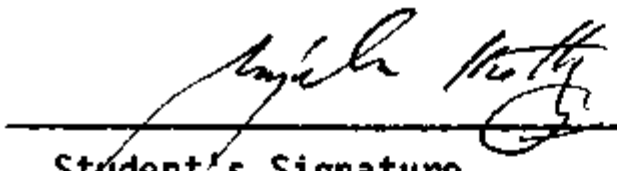
[13] Shih.T.I-P., Bailey.R.T., Nguyen,H.L.,and Roelke,R.J..(1991).Algebraic Grid Generation for Computer Geometries..*Intl. Journal for Numerical methods in Fluids*..13(1).pp1-31.

[14] Robert.T.G.,and Karney.D.L..*A Collection of Matrices for Testing Computational Algorithms*..pp10-11.Wiley-Interscience.Wiley&Sons.NY.

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)



Student's Signature

4/12/96

Date

Disagree (Permission is not granted.)

Student's Signature

Date