Optimal Development of Doubly Curved Surfaces

by

Guoxin Yu

B.S. in Naval Architecture (1990)
M.S. in Structural Mechanics (1993)
Shanghai Jiao Tong University, P. R. China

Submitted to the Department of Ocean Engineering and the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Master of Science in Naval Architecture and Marine Engineering

and

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1999

Author ....
Department of Ocean Engineering
May 7, 1999

Certified by ....
Nicholas M. Patrikalakis, Kawasaki Professor of Engineering
Thesis Co-Supervisor

Certified by ....
Takashi Maekawa, Lecturer and Research Scientist
Thesis Co-Supervisor

Certified by ....
John N. Tsitsiklis, Professor of Electrical Engineering
Thesis Reader

Accepted by ....
Arthur B. Baggeroer, Ford Professor of Engineering
Chairman, Departmental Committee on Graduate Students
Department of Ocean Engineering

Accepted by ....
Arthur C. Smith, Professor of Electrical Engineering
Chairman, Departmental Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Optimal Development of Doubly Curved Surfaces

by

Guoxin Yu

Submitted to the Department of Ocean Engineering
and the Department of Electrical Engineering and Computer Science
on May 7, 1999, in partial fulfillment of the
requirements for the degrees of
Master of Science in Naval Architecture and Marine Engineering
and
Master of Science in Electrical Engineering and Computer Science

Abstract

Surfaces of many engineering structures are commonly fabricated as doubly curved shapes to fulfill functional requirements such as hydrodynamic, aesthetic, or structural. Given a three-dimensional design surface, the first step of the fabrication process is flattening or planar development of this surface into a planar shape so that the manufacturer can not only determine the initial shape of the flat plate but also estimate the strain distribution required to form the shape. In this thesis, we develop an algorithm for optimal development of a general doubly curved surface in the sense that the strain from the surface to its planar development is minimized. A planar development corresponding to minimum stretching or shrinkage is highly desirable for the following reasons: (1) it saves material; (2) it reduces the work needed to form the planar shape to the doubly curved design surface. The development process is modeled by tensile strains along isoparametric directions, or along principal curvature directions from the curved surface to its planar development. The distribution of the appropriate minimum strain field is obtained by solving a constrained nonlinear programming problem. Based on the strain distribution and the coefficients of the first fundamental form of the curved surface, another unconstrained nonlinear programming problem is solved to obtain the optimal developed planar shape. Convergence, complexity, and accuracy of the algorithm are studied. Examples show the effectiveness of this algorithm.

Thesis Co-Supervisor: Nicholas M. Patrikalakis, Kawasaki Professor of Engineering

Thesis Co-Supervisor: Takashi Maekawa, Lecturer and Research Scientist

Thesis Reader: John N. Tsitsiklis, Professor of Electrical Engineering

Acknowledgments

First of all, I want to thank my wife for her love, understanding, and support during my study at MIT.

I would like to thank my thesis supervisors, Professor Nicholas M. Patrikalakis and Dr. Takashi Maekawa, for their expert advice on my research work and instructive guidance on my academic study. I also want to thank Professor John N. Tsitsiklis for taking time to read the thesis and providing valuable feedback.

Thanks also go to Mr. Fred Baker, for providing a stable hardware environment for my thesis work, Stephen L. Abrams for programming assistance, and Design Laboratory fellows Dr. Wonjoon Cho, Mr. Todd R. Jackson, Mr. Guoling Shen for useful discussions.

Funding for this research was obtained in part from the New Industry Research Organization (NIRO), and from the MIT Sea Grant and Department of Ocean Engineering.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and problem statement

In engineering applications, there exist two kinds of surfaces, developable and non-developable surfaces, which are also called singly and doubly curved surfaces, respectively. A developable surface has zero Gaussian curvature at all points, while a non-developable surface has non-zero Gaussian curvature at least in some region. A developable surface is highly favorable in metal forming since it can be formed only by bending without tearing or stretching. For this reason, developable surfaces are widely used in manufacturing parts whose materials are not amenable to stretching. However, surfaces of many engineering structures are commonly fabricated as doubly curved shapes to fulfill functional requirements such as hydrodynamic, aesthetic, or structural. For example, a large portion of the shell plates of ship hulls or airplane fuselages are doubly curved surfaces.

Given a three-dimensional design surface, the first step of the fabrication process is flattening or planar development of this surface into a planar shape so that the manufacturer can not only determine the initial shape of the flat plate but also estimate the strain distribution required to form the shape. Then the planar shape is formed into the design surface by various approaches such as forming by matching dies, by continuous hammering, or by line heating using oxyacetylene torch, laser or heat by induction. This planar shape is usually not unique since theoretically, a large variety of initial planar shapes can be deformed into the curved surface if adequate stretching or shrinkage is allowed. However, in real practice, a planar development corresponding to minimum stretching or shrinkage is highly desirable for the following reasons: (1) it saves material; (2) it reduces the work needed to form the planar shape to the doubly curved design surface.

In this thesis, we develop an algorithm for optimal development of a general doubly curved surface in the sense that the strain from the surface to its planar development is minimized. A tensile strain (stretching) from the curved surface to its planar development is assumed which corresponds to forming from the planar shape to curved surface by the line heating approach.

1.2 Overview of metal forming

Forming takes place in a metal any time it is subjected to stresses that are greater than the yield point or when the deformation stress moves from the elastic to the plastic range. The two methods currently used for forming steel plates into curved shells, which may be classified according to the mechanisms used to bend the plates, are mechanical forming and thermo-mechanical forming.

In *mechanical forming* of a steel plate, the steel plate, which is initially flat at room temperature, is formed into the desired shape by producing plastic deformations in appropriate amount and distributions. One of the most common methods of producing the necessary plastic deformation is to press the plate to a die of proper shape. Another method is to feed the plate through a set of rollers (cold rolling) to produce the desired shape.

When a plate is being *thermo-mechanically* formed, plastic deformation is produced by the thermal stresses generated during the heating and subsequent cooling of the plate. The thermo-mechanical process involved in plate bending is based on the principle of heating one side of a plate while the other side is kept cool. The temperature gradient in the material causes the metal to deform in the negative direction (opposite to final desired shape). During this transient state, the expanded metal is constrained by the surrounding cooler metal, and compressive stresses result. When the heat is removed, the plate cools and the metal contracts. The plate will then deform in the direction reverse to that when it was heated, as shown in Figure 1-1.

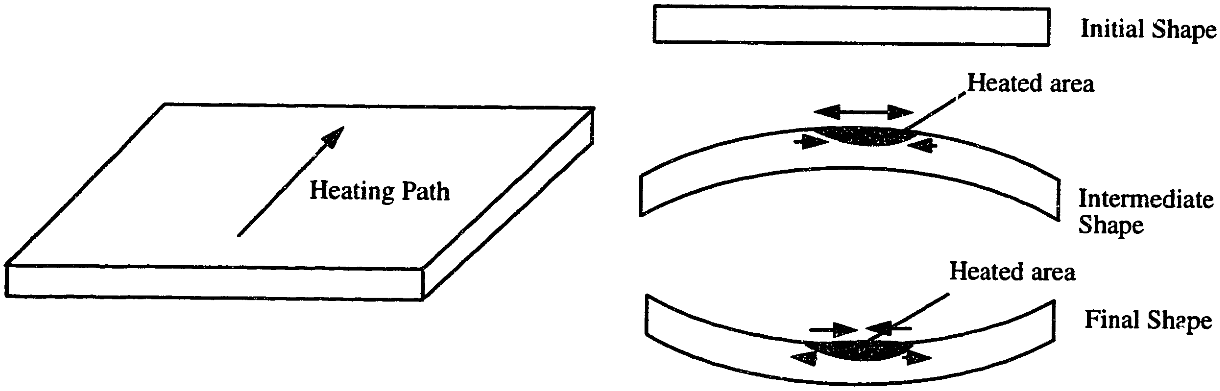

Figure 1-1: Line heating mechanism

Compared with mechanical pressing, *thermo-mechanical forming*, using an oxy-acetylene torch, is more versatile and less expensive. Steel plates can also be formed with complex double curvatures, and the resulting residual stresses are minimal. Another type of heat source is a laser. The stability and controllability of the laser beam make it preferable as a heat source for automation.

1.3 Previous work

Early surface development procedures were implemented in shipyards based on geodesic development during the last three decades, mainly for ship hull plates whose Gaussian curvature is very small. More recently, Letcher [15] presents a basic geometric theory for flattening and fabrication of doubly curved plates. The mapping from the curved surface to its planar development is modeled by adding in-plane strains to the curved surface. The strain field is obtained by solving a generalized Poisson's equation with the source term equal to the Gaussian curvature. However, since the problem is formulated as a boundary value problem, a good solution relies on a well specified boundary condition which is hard to know beforehand. Also, the differential equation is formulated in an orthogonal coordinate system and it is not trivial to formulate in a non-orthogonal coordinate system. Ueda et al. [22] investigate the relation between the final shape of a plate and the inherent strain. They compute the strain caused by deformation from the initial configuration to the final one using large deformation elastic FEM analysis. Since the initial configuration is rectangular, their approach can only be applied to the cases when the 3D surface is relatively flat, i.e. the curvature is small. Manning [17] developed a procedure for surface development based on an isometric tree. A tree of lines with a spine and branches is first drawn on the curved surface. Then the spline and the branch curves are developed isometrically onto planar curves, using the geodesic curvature of the spine and branches on the surface as the curvature of the planar curves. The envelope of the developed pattern forms the planar developed shape. Obviously, the shape of the planar development depends on the choice of the spline and branch curves, since in this development scheme, the stretching along both the spline and branch curves is zero. This procedure is applied in the shoemaking industry and may not be applicable in metal forming. Another disadvantage of this procedure is that it does not provide the field of strain (deformation). Hinds et al. [12] develop doubly curved surfaces by first approximating them by quadrilateral facets, then flattening these platelets allowing some gaps in the developed patterns. This method is applied in the clothing industry. The disadvantage of this method is that the developed shape depends on the starting edge chosen and again if used in metal forming, it is not guaranteed that the forming process is realizable from the planar shape to the curved surface. Azariadis and Aspragathos [1] extend the work by Hinds et al. [12] to reduce the gaps by minimizing the Euclidean distances of pairs of corresponding points between two successive strips. The quality of the development approaches in [12] [1] largely depends on the choice of guide-strip or starting edge. Cho et al. [4] present an algorithm to approximately develop a doubly curved surface by minimizing the mapping error function for locally isometric mapping between a given and developed surface net. The method has been applied to construct an auxiliary planar domain of triangulation for tessellating trimmed parametric surface patches, which sufficiently preserves the shape of triangles when mapped into three-dimensional space. Again, the applicability of Cho's method [4] for metal forming is unclear. The disadvantage of the available literature is that there is no general algorithm for optimal development of general curved surfaces for metal forming process.

1.4 Thesis outline

The remaining of the thesis is arranged as follows:

Chapter 2 reviews differential geometry of surfaces, the representation of surfaces in B-spline form, as well as the derivation of some important theorems on the gradient of the first fundamental form coefficients across the offset direction.

Chapter 3 presents the algorithms for surface development based on the strains along isoparametric lines.

Chapter 4 presents the algorithms for surface development based on the strains along principal curvature directions.

Chapter 5 analyzes the complexity and the accuracy of the algorithms with respect to the number of grid points.

Chapter 6 illustrates the performance of the surface development algorithms by means of several examples. Surfaces with positive, negative, and mixed Gaussian curvature are developed into two-dimensional shapes.

Chapter 7 concludes the thesis and presents suggestions for future research.

Chapter 2

Surface theory

2.1 Review of differential geometry of surfaces

This section reviews differential geometry of surfaces, which provides the background for surface development algorithms. For further detail we refer to classical differential geometry books such as [20, 14, 23, 16, 7].

2.1.1 Definition of surfaces

A surface can be defined in three different ways. They are:

(1) *Implicit surface* given by a function $F(x, y, z) = 0$. For example, the equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ represents an ellipsoid with half axes a, b, c in x, y, z directions respectively.

(2) *Explicit surface*, which is obtained by solving the implicit equation $F(x, y, z) = 0$ for one of the variables as a function of the other two. For example, $z = \frac{1}{2}(\alpha x^2 + \beta y^2)$ represents an explicit quadratic surface. When $\alpha\beta < 0$, the surface is a hyperbolic paraboloid; when $\alpha\beta > 0$, it is an elliptic paraboloid; and when either $\alpha = 0$ or $\beta = 0$, it is a parabolic cylinder.

(3) *Parametric surface* represented by $x = x(u, v)$, $y = y(u, v)$, $z = z(u, v)$. Here functions $x(u, v)$, $y(u, v)$, $z(u, v)$ have continuous partial derivatives of the

r^{th} order, and the parameters u and v are restricted to some intervals (i.e., $u_1 \leq u \leq u_2$, $v_1 \leq v \leq v_2$) leading to parametric surface patches. This rectangular domain D of u, v is called *parametric space* and it is frequently the unit square. If derivatives of the surface are continuous up to the r^{th} order, the surface is said to be of class r, denoted by C^r. In vector notation:

$$\mathbf{r} = \mathbf{r}(u, v) \tag{2.1}$$

where $\mathbf{r} = (x, y, z), \quad \mathbf{r}(u, v) = (x(u, v), y(u, v), z(u, v))$

Generally speaking, it is difficult to trace implicit surfaces, while it is easy to trace the other two surfaces; it is easy to check if a point lies on an implicit or an explicit surface, while it is difficult to check this for a parametric surface; a multi-valued surface can be represented as an implicit surface or a parametric surface, but this is not directly possible for an explicit surface. The rest of the thesis uses parametric expressions of surfaces.

2.1.2 First fundamental form (arc length)

Consider a curve on a surface $\mathbf{r} = \mathbf{r}(u(t), v(t))$, which corresponds to a curve $\beta(t) = (u(t), v(t))$ in the parametric plane. The arc length of the curve on the surface is given by [7]

$$\begin{aligned} ds &= |\dot{\mathbf{r}}|dt = |\mathbf{r}_u \dot{u} + \mathbf{r}_v \dot{v}|dt \\ &= \sqrt{(\mathbf{r}_u \dot{u} + \mathbf{r}_v \dot{v}) \cdot (\mathbf{r}_u \dot{u} + \mathbf{r}_v \dot{v})}dt \\ &= \sqrt{(\mathbf{r}_u \cdot \mathbf{r}_u)du^2 + 2\mathbf{r}_u \mathbf{r}_v dudv + (\mathbf{r}_v \cdot \mathbf{r}_v)dv^2} \\ &= \sqrt{Edu^2 + 2Fdudv + Gdv^2} \end{aligned} \tag{2.2}$$

where

$$E \;=\; \mathbf{r}_u \cdot \mathbf{r}_u, \quad F = \mathbf{r}_u \cdot \mathbf{r}_v, \quad G = \mathbf{r}_v \cdot \mathbf{r}_v \tag{2.3}$$

are called coefficients of the first fundamental form; overdot ($\dot{}$) denotes derivative with respect to parameter t and subscripts u, v denote partial derivatives with respect to u, v. The first fundamental form is defined as

$$I = d\mathbf{r} \cdot d\mathbf{r} = Edu^2 + 2Fdudv + Gdv^2 \tag{2.4}$$

Note that $E = \mathbf{r}_u \cdot \mathbf{r}_u > 0$ and $G = \mathbf{r}_v \cdot \mathbf{r}_v > 0$ if $\mathbf{r}_u \neq 0$ and $\mathbf{r}_v \neq 0$, and the first fundamental form I is positive definite. That is $I \geq 0$ and $I = 0$ if and only if $du = 0$ and $dv = 0$ since

$$I = \frac{1}{E}(E\ du + F\ dv)^2 + \frac{EG - F^2}{E} dv^2 \tag{2.5}$$

and

$$EG - F^2 = (\mathbf{r}_u \cdot \mathbf{r}_u)(\mathbf{r}_v \cdot \mathbf{r}_v) - (\mathbf{r}_u \cdot \mathbf{r}_v)^2 = |\mathbf{r}_u \times \mathbf{r}_v|^2 > 0. \tag{2.6}$$

It can be seen that I depends only on the surface and not on the parametrization.

The differential area of the surface is

$$dA = |\mathbf{r}_u du \times \mathbf{r}_v dv| = |\mathbf{r}_u \times \mathbf{r}_v| dudv$$

By using Equation (2.6), we obtain

$$\begin{aligned} A &= \int\!\!\int dA \\ &= \int\!\!\int \sqrt{EG - F^2}\, dudv \end{aligned} \tag{2.7}$$

2.1.3 Tangent plane and normal vector

Tangent plane at a point $\mathbf{r}(u_0, v_0)$ is the union of tangent vectors of all curves on the surface passing through $\mathbf{r}(u_o, v_o)$, as shown in Figure 2-1. Since the tangent vector of a curve on a parametric surface is given by $\dot{\mathbf{r}} = \mathbf{r}_u\dot{u} + \mathbf{r}_v\dot{v}$, the tangent plane is the plane of the vectors $\mathbf{r}_u$ and $\mathbf{r}_v$. The equation of the tangent plane is

$$\mathbf{T}_p(u_0, v_0) = \mathbf{r}(u_0, v_0) + \lambda\mathbf{r}_u(u_0, v_0) + \mu\mathbf{r}_v(u_0, v_0) \tag{2.8}$$

where λ and μ are real variables parameterizing the plane.

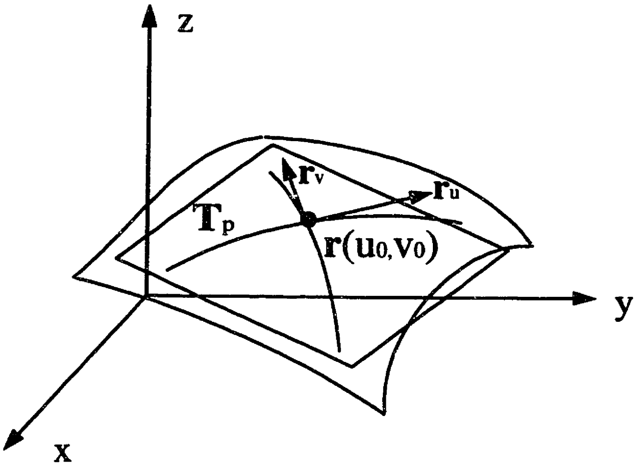

Figure 2-1: The tangent plane at a point on a surface.

The surface normal is the vector at point $\mathbf{r}(u_0, v_0)$ perpendicular to the tangent plane, see Figure 2-2. Therefore the unit normal is given by

$$\mathbf{N} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{|\mathbf{r}_u \times \mathbf{r}_v|}. \tag{2.9}$$

A *regular* (ordinary) point $\mathbf{P}$ on the surface is defined as one for which $\mathbf{r}_u \times \mathbf{r}_v \neq \mathbf{0}$. A point where $\mathbf{r}_u \times \mathbf{r}_v = \mathbf{0}$ is called a *singular* point. The condition $\mathbf{r}_u \times \mathbf{r}_v \neq \mathbf{0}$ requires that at point $\mathbf{P}$ the vectors $\mathbf{r}_u$ and $\mathbf{r}_v$ do not vanish and have different directions. In the rest of this thesis, we assume that the surfaces we use include only regular points.

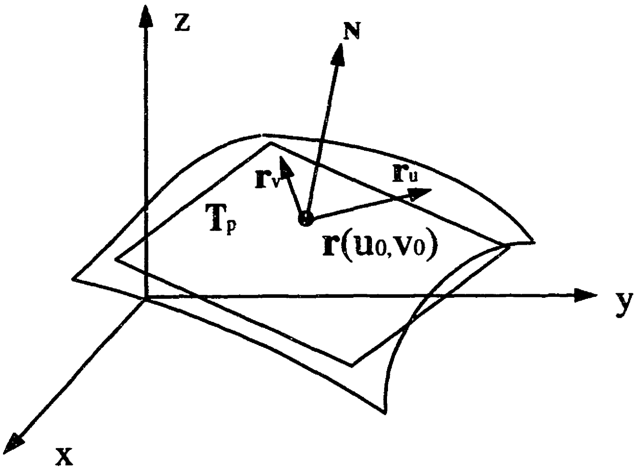

Figure 2-2: The normal to the point on a surface.

2.1.4 Second fundamental form *II*

In order to quantify the curvatures of a surface S, we consider a curve C on S which passes through a point P as shown in Figure 2-3. $\mathbf{t}$ is the unit tangent vector and $\mathbf{n}$ is the unit normal vector of the curve C at point P. Then the curvature vector $\mathbf{k}$ of the curve C can be expressed as:

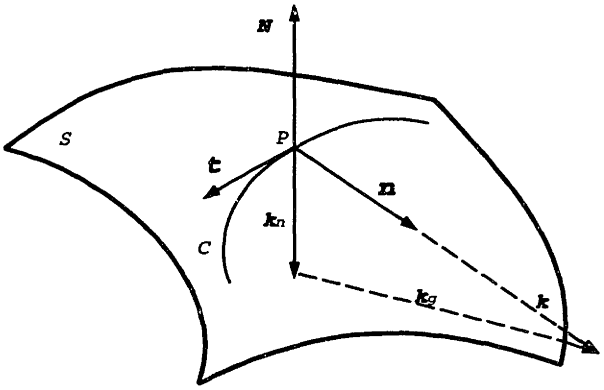

Figure 2-3: Definition of normal curvature

$$\mathbf{k} = \frac{dt}{ds} = \kappa\mathbf{n} = \mathbf{k}_n + \mathbf{k}_g \tag{2.10}$$

where $\mathbf{k}_n$ is the normal curvature vector normal to the surface; $\mathbf{k}_g$ is the geodesic curvature vector tangent to the surface. Since $\mathbf{k}_n$ is along the surface normal, it can be written as

$$\mathbf{k}_n = \kappa_n \mathbf{N}, \tag{2.11}$$

where κ_n is called the normal curvature of the surface at P in the direction $\mathbf{t}$, and $\mathbf{N}$ is the unit normal of the surface as defined in Equation (2.9).

The second fundamental form is given by

$$II = -d\mathbf{r} \cdot d\mathbf{N} = Ldu^2 + 2Mdudv + Ndv^2 \tag{2.12}$$

where

$$L = -\mathbf{r}_u \cdot \mathbf{N}_u = \mathbf{N} \cdot \mathbf{r}_{uu} \tag{2.13}$$

$$M = -(\mathbf{r}_u \cdot \mathbf{N}_v + \mathbf{r}_v \cdot \mathbf{N}_u) = \mathbf{N} \cdot \mathbf{r}_{uv} \tag{2.14}$$

$$N = -\mathbf{r}_v \cdot \mathbf{N}_v = \mathbf{N} \cdot \mathbf{r}_{vv} \tag{2.15}$$

are the coefficients of the second fundamental form. In deriving Equations (2.13-2.15) above, the relations $\mathbf{r}_u \cdot N = 0, \;\; \mathbf{r}_v \cdot N = 0$ are used. The normal curvature can be expressed by

$$\kappa_n = \frac{II}{I} = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2} \tag{2.16}$$

where $\lambda = \frac{dv}{du}$.

Suppose P is a point on a surface and Q is a point in the neighborhood of P. Taylor's expansion gives

$$\mathbf{r}(u+du, v+dv) = \mathbf{r}(u,v) + \mathbf{r}_u du + \mathbf{r}_v dv + \frac{1}{2}(\mathbf{r}_{uu}du^2 + 2\mathbf{r}_{uv}dudv + \mathbf{r}_{vv}dv^2) + H.O.T.$$

Therefore

$$\vec{PQ} = \mathbf{r}(u+du, v+dv) - \mathbf{r}(u,v) = \mathbf{r}_u du + \mathbf{r}_v dv + \frac{1}{2}(\mathbf{r}_{uu}du^2 + 2\mathbf{r}_{uv}dudv + \mathbf{r}_{vv}dv^2) + H.O.T.$$

Thus, the projection of $\vec{PQ}$ onto $\mathbf{N}$

$$d = \vec{PQ} \cdot \mathbf{N} = (\mathbf{r}_u du + \mathbf{r}_v dv) \cdot \mathbf{N} + \frac{1}{2} II$$

and since $\mathbf{r}_u \cdot \mathbf{N} = \mathbf{r}_v \cdot \mathbf{N} = 0$, we obtain

$$d = \frac{1}{2} II = \frac{1}{2}(Ldu^2 + 2Mdudv + Ndv^2)$$

Therefore, the second fundamental form evaluated at a point P is a measure of the distance from its neighboring points to the tangent plane of the surface at point P. We want to observe in which situation d is positive and negative. When $d = 0$

$$Ldu^2 + 2Mdudv + Ndv^2 = 0$$

Then we solve for du

$$du = \frac{-M \pm \sqrt{(Mdv)^2 - LNdv^2}}{L} = \frac{-M \pm \sqrt{M^2 - LN}}{L} dv \qquad (2.17)$$

We can have the following three situations:

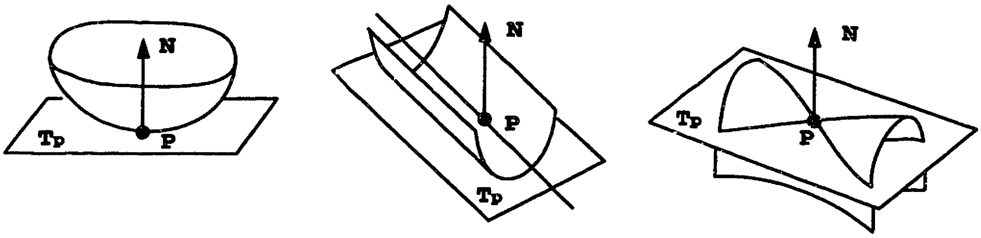

Figure 2-4: (a) Elliptic point; (b) Parabolic point; (c) Hyperbolic point.

(1) If $M^2 - LN < 0$, there is no real root. That means there is no intersection (locally) between the surface and its tangent plane except at point P. P is called an *elliptic point* (Figure 2-4(a)).

(2) If $M^2 - LN = 0$, there is a double root. The surface intersects its tangent plane (locally) with one line $du = -\frac{M}{L} dv$, which passes through point P. P is called

a *parabolic point* (Figure 2-4(b)).

(3) If $M^2 - LN > 0$, there are two roots. The surface intersects its tangent plane (locally) with two lines $du = \frac{-M \pm \sqrt{M^2 - LN}}{L} dv$, which intersect at point P. P is called a *hyperbolic point* (Figure 2-4(c)).

2.1.5 Principal curvatures

The extreme values of κ_n can be obtained by evaluating $\frac{d\kappa_n}{d\lambda} = 0$ of Equation (2.16), which gives:

$$(E + 2F\lambda + G\lambda^2)(N\lambda + M) - (L + 2M\lambda + N\lambda^2)(G\lambda + F) = 0 \qquad (2.18)$$

Since

$$\begin{aligned} E + 2F\lambda + G\lambda^2 &= (E + F\lambda) + \lambda(F + G\lambda) \\ L + 2M\lambda + N\lambda^2 &= (L + M\lambda) + \lambda(M + N\lambda) \end{aligned}$$

Equation (2.18) can be reduced to

$$(E + F\lambda)(M + N\lambda) = (L + M\lambda)(F + G\lambda) \qquad (2.19)$$

Thus

$$\kappa_n = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2} = \frac{M + N\lambda}{F + G\lambda} = \frac{L + M\lambda}{E + F\lambda} \qquad (2.20)$$

Therefore κ_n satisfies the two simultaneous equations

$$(L - \kappa_n E)du + (M - \kappa_n F)dv = 0 \qquad (2.21)$$

$$(M - \kappa_n F)du + (N - \kappa_n G)dv = 0 \qquad (2.22)$$

These equations can be simultaneously satisfied if and only if

$$\left|\begin{bmatrix} L-\kappa_n E & M-\kappa_n F \\ M-\kappa_n F & N-\kappa_n G \end{bmatrix}\right| = 0 \tag{2.23}$$

where | | denotes the determinant of a matrix. Expanding and defining K and H as

$$\begin{aligned} K &= \frac{LN-M^2}{EG-F^2} && (2.24) \\ H &= \frac{EN+GL-2FM}{2(EG-F^2)} && (2.25) \end{aligned}$$

we obtain a quadratic equation for κ_n as follows:

$$\kappa_n^2 - 2H\kappa_n + K = 0 \tag{2.26}$$

The values K and H are called Gauss (Gaussian) and mean curvature respectively. The discriminant D of quadratic equation (2.26) can be expressed as follows:

$$\begin{aligned} D &= H^2 - K \\ &= \frac{(EN+GL-2FM)^2 - 4(EG-F^2)(LN-M^2)}{4(EG-F^2)^2} \end{aligned}$$

By virtue of (2.6) and the fact that the surface includes only regular points, the denominator is always positive, so we only need to investigate the numerator. The numerator can be written as:

$$\begin{aligned} & (EN+GL-2FM)^2 - 4(EG-F^2)(LN-M^2) \\ = 4\left(\frac{EG-F^2}{E^2}\right)(EM-FL)^2 + & [EN-GL-\frac{2F}{E}(EM-FL)]^2 \geq 0 \end{aligned}$$

Thus, $D \geq 0$.

Upon solving Equation (2.26) for the extreme values of curvature, we have:

$$\kappa_{max} = H + \sqrt{H^2 - K} \tag{2.27}$$

$$\kappa_{min} = H - \sqrt{H^2 - K} \tag{2.28}$$

From equations (2.27), (2.28), it is readily seen that

$$K = \kappa_{max}\kappa_{min} \tag{2.29}$$

$$H = \frac{\kappa_{max} + \kappa_{min}}{2} \tag{2.30}$$

From equation (2.24) (since $EG - F^2 > 0$, see Equation 2.6), we have:

(1) if $K > 0$ then $LN > M^2$ which means this is an elliptic point

(2) if $K = 0$ then $LN = M^2$ which means this is a parabolic point

(3) if $K < 0$ then $LN < M^2$ which means this is a hyperbolic point

2.1.6 Gauss curvature

Alternatively, the Gaussian curvature K can be expressed as a function of E, F, G and their derivatives [20]. After substituting Equation (2.9) into Equations (2.13–2.15) and substituting Equation (2.13–2.15) into Equation (2.24), the Gaussian curvature is expressed as a function of triple products of the derivatives of $\mathbf{r}$. These triple products are then expressed by products of the partial derivatives of the first fundamental form coefficients, and the result is:

$$\begin{aligned} & 4(EG - F^2)^2K = E\left(E_vG_v - 2F_uG_v + G_u^2\right) \\ + \quad & F\left(E_uG_v - E_vG_u - 2E_vF_v + 4E_uF_v - 2F_uG_u\right) \\ + \quad & G\left(E_uG_u - 2E_uF_v + E_v^2\right) - 2(EG - F^2)\left(E_{vv} - 2F_{uv} + G_{uu}\right) \end{aligned} \tag{2.31}$$

2.2 Theorems on the gradients of the first fundamental form coefficients

In this section, some theorems are presented on the gradients of the first fundamental form coefficients of the offset surface along the offset distance direction, which correspond to the gradients of those coefficients across the thickness for a curved shell plate. These theorems show that the gradients of the first fundamental form coefficients of the offset surface provide the mechanism of surface curvature. In metal forming, this means that the non-uniformity of the tensile or compressive strains across the thickness generates surface curvature. To the author's knowledge based on a thorough literature search, these results have not been studied in the Computer Aided Geometric Design community.

For a curved shell plate with thickness h, we consider $\mathbf{r}(u, v)$ as the mid-surface if its offset surfaces with signed distances $h/2$ and $-h/2$ are the upper and lower surfaces.

Theorem 2.2.1 *The coefficients of the second fundamental form of a parametric surface can be expressed by the derivatives of the coefficients of the first fundamental form of its offset surface with respect to the offset distance d, evaluated at $d = 0$.*

Proof: Let a progenitor parametric surface (called mid-surface) be defined by (2.1) and the coefficients of its first fundamental form are shown in Equation (2.3). Then the offset surface with signed distance d along the normal from the mid-surface is:

$$\hat{\mathbf{r}}(u, v) = \mathbf{r}(u, v) + d\mathbf{N}(u, v), \tag{2.32}$$

where $\mathbf{N}$ is the unit normal vector of the surface $\mathbf{r}(u, v)$ at (u, v), as shown in Equation (2.9). The first fundamental coefficients of the offset surface are functions of u, v, d:

$$\hat{E} = \hat{\mathbf{r}}_u \cdot \hat{\mathbf{r}}_u = (\mathbf{r}_u + d\mathbf{N}_u) \cdot (\mathbf{r}_u + d\mathbf{N}_u) \tag{2.33}$$

$$\hat{F} = \hat{\mathbf{r}}_u \cdot \hat{\mathbf{r}}_v = (\mathbf{r}_u + d\mathbf{N}_u) \cdot (\mathbf{r}_v + d\mathbf{N}_v) \tag{2.34}$$

$$\hat{G} = \hat{\mathbf{r}}_v \cdot \hat{\mathbf{r}}_v = (\mathbf{r}_v + d\mathbf{N}_v) \cdot (\mathbf{r}_v + d\mathbf{N}_v) \tag{2.35}$$

Their derivatives with respect to d are:

$$\begin{aligned}
\frac{\partial \hat{E}}{\partial d} &= 2(\mathbf{r}_u + d\mathbf{N}_u) \cdot \mathbf{N}_u & (2.36) \\
\frac{\partial \hat{F}}{\partial d} &= (\mathbf{r}_u + d\mathbf{N}_u) \cdot \mathbf{N}_v + (\mathbf{r}_v + d\mathbf{N}_v) \cdot \mathbf{N}_u & (2.37) \\
\frac{\partial \hat{G}}{\partial d} &= 2(\mathbf{r}_v + d\mathbf{N}_v) \cdot \mathbf{N}_v & (2.38)
\end{aligned}$$

At mid-surface, ie. for d=0

$$\begin{aligned}
\left.\frac{\partial \hat{E}}{\partial d}\right|_{d=0} &= 2\mathbf{r}_u \cdot \mathbf{N}_u = -2L & (2.39) \\
\left.\frac{\partial \hat{F}}{\partial d}\right|_{d=0} &= \mathbf{r}_u \cdot \mathbf{N}_v + \mathbf{r}_v \cdot \mathbf{N}_u = -2M & (2.40) \\
\left.\frac{\partial \hat{G}}{\partial d}\right|_{d=0} &= 2\mathbf{r}_v \cdot \mathbf{N}_v = -2N & (2.41)
\end{aligned}$$

Therefore, the second fundamental coefficients of the mid-surface can be expressed by the derivatives of $\hat{E}$, $\hat{F}$, $\hat{G}$, evaluated at the mid-surface. ■

A similar result can be derived for the metrics along principal curvature directions.

Corollary 2.2.1 *Let the parameters along maximum principal curvature and minimum principal curvature directions be s and t. Then*

$$\begin{aligned}
\left.\frac{\partial(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)}{\partial d}\right|_{d=0} &= -2k_{max}(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)|_{d=0} & (2.42) \\
\left.\frac{\partial(\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t)}{\partial d}\right|_{d=0} &= -2k_{min}(\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t)|_{d=0} & (2.43)
\end{aligned}$$

Proof: We have for the mid-surface defined by (2.1):

$$\mathbf{r}_s = \mathbf{r}_u u_s + \mathbf{r}_v v_s \qquad (2.44)$$

and

$$\begin{aligned}\mathbf{r}_s \cdot \mathbf{r}_s &= (\mathbf{r}_u u_s + \mathbf{r}_v v_s) \cdot (\mathbf{r}_u u_s + \mathbf{r}_v v_s) \\ &= E u_s^2 + 2F u_s v_s + G v_s^2 \end{aligned} \tag{2.45}$$

For the offset at distance d along the normal from the mid-surface, as defined in Equation (2.32),

$$\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s = \hat{E} u_s^2 + 2\hat{F} u_s v_s + \hat{G} v_s^2 \tag{2.46}$$

Therefore, after taking partial derivatives of Equation (2.46) with respect to d, and using Equations (2.39 – 2.41), we obtain

$$\left.\frac{\partial(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)}{\partial d}\right|_{d=0} = -2(L u_s^2 + 2M u_s v_s + N v_s^2) \tag{2.47}$$

From Equation (2.20),

$$\kappa_{max} = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2} = \frac{L u_s^2 + 2M u_s v_s + N v_s^2}{E u_s^2 + 2F u_s v_s + G v_s^2} \tag{2.48}$$

which results in

$$L u_s^2 + 2M u_s v_s + N v_s^2 = k_{max}(E u_s^2 + 2F u_s v_s + G v_s^2) \tag{2.49}$$

Thus by substituting Equation (2.49) into Equation (2.47), and using Equation (2.45), we have

$$\left.\frac{\partial(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)}{\partial d}\right|_{d=0} = -2k_{max}(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)|_{d=0}$$

Similarly, along the minimum principal curvature direction, where the parameter is t, we obtain

$$\left.\frac{\partial(\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t)}{\partial d}\right|_{d=0} = -2k_{min}(\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t)|_{d=0}$$

■

Equations (2.39-2.41), (2.42, 2.43) play an important role in surface development algorithms, since in engineering applications, curved plates have finite thickness, no matter how thin they are.

2.3 Review of B-spline surfaces

In the area of computer aided design, parametric surfaces are usually expressed as NURBS (Non-Uniform Rational B-Spline) surfaces. NURBS surfaces are favored because of their flexibility, generality and explicit incorporation in data exchange standards [13]. NURBS surface patches are defined by

$$\mathbf{R}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij}\mathbf{R}_{ij}N_{i,k}(u)N_{j,l}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij}N_{i,k}(u)N_{j,l}(v)} \tag{2.50}$$

where weights $w_{ij} > 0$. $\mathbf{R}_{ij}$ are $(n+1)(m+1)$ control points; $N_{i,k}(u)$, $N_{j,l}(v)$ are piecewise polynomial B-spline basis functions of order k (or degree $k-1$) with $n \geq k-1$ and l (or degree $l-1$) with $m \geq l-1$, respectively. The B-spline basis functions are defined on a set $\mathbf{T}$ of non-decreasing real numbers which is called the knot vector, as follows:

$$\mathbf{T} = \{\underbrace{t_o = t_1 = \cdots = t_{k-1}}_{k\ equal\ values} < \underbrace{t_k \leq t_{k+1} \leq \cdots \leq t_n}_{n-k+1\ internal\ knots} < \underbrace{t_{n+1} = \cdots = t_{n+k}}_{k\ equal\ values}\} \tag{2.51}$$

Given the knot vector, the B-spline basis functions $N_{i,k}(u)$ are defined recursively based on the Cox-de Boor algorithm [13]

$$N_{i,1}(u) = \begin{cases} 1 & u \in [t_i, t_{i+1}) \\ 0 & u \notin [t_i, t_{i+1}) \end{cases} \tag{2.52}$$

$$N_{i,k}(u) = \frac{u - t_i}{t_{i+k-1} - t_i}N_{i,k-1}(u) + \frac{t_{i+k} - u}{t_{i+k} - t_{i+1}}N_{i+1,k-1}(u) \tag{2.53}$$

(set $\frac{0}{0} = 0$ above when it occurs)

NURBS formulation allows for exact representation of quadrics, tori, surfaces of revolution and very general free-form surfaces. If all $w_{ij} = 1$, the integral B-Spline case is recovered.

A special case of a NURBS surface is the tensor product Bézier surface defined by

$$\mathbf{R}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{R}_{ij} B_{i,n}(u) B_{j,m}(v) \qquad 0 \le u, v \le 1 \qquad (2.54)$$

where $\mathbf{R}_{ij}$ are the control points creating a control polyhedron (net), and $B_{i,n}(u)$ represents the Bernstein polynomial basis functions of degree n. The definition of a Bernstein polynomial is:

$$B_{i,n}(u) = \begin{pmatrix} n \\ i \end{pmatrix} u^i (1-u)^{n-i} \quad i = 0, 1, 2, \ldots n$$

$$\text{where } \begin{pmatrix} n \\ i \end{pmatrix} = \frac{n!}{i!(n-i)!}$$

Chapter 3

Surface development along isoparametric directions

In this chapter, algorithms for surface development are presented. The process of surface development is expressed by tensile strains along u, v isoparametric directions. This corresponds to forming a plate into a curved surface only by shrinkage which can be realized by line heating process. We seek an optimal development of the doubly curved surface in a sense that the strains are minimized. This is realized by solving a nonlinear constrained optimization problem. After the tensile strains are computed, a 2D development can be determined based on the first fundamental form coefficients of the 2D flattened shape. These first fundamental form coefficients of the 2D flattened shape are functions of the strains and the first fundamental form coefficients of the curved surface.

3.1 Determination of strain field

3.1.1 Formulation

We assume that the surface is defined by a parametric vector equation of the form

$$\mathbf{r} = \mathbf{r}(u, v) \tag{3.1}$$

The surface and its planar development are shown in Figure 3-1. The coefficients of the first fundamental form of the curved surface are given by Equation (2.3).

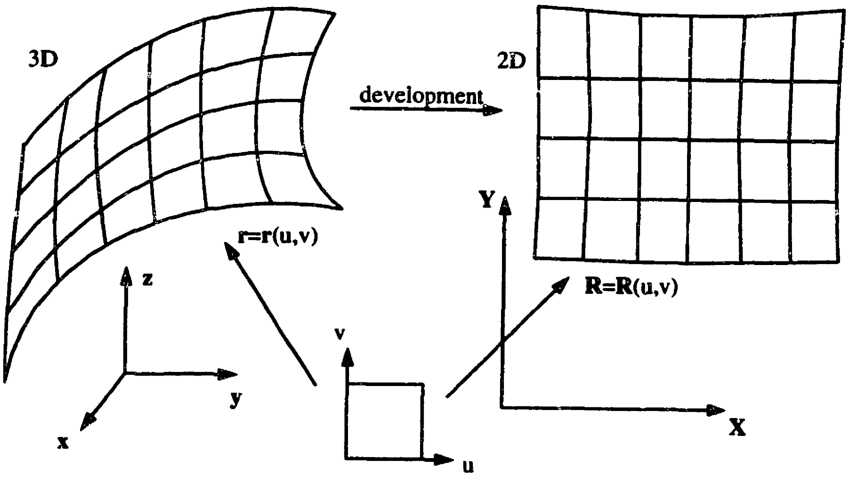

Figure 3-1: Curved surface and its planar development

Assume that during metal forming by line heating, the normal strain along u line is $\epsilon^u(u,v) \leq 0$, and the normal strain along v line is $\epsilon^v(u,v) \leq 0$. On the contrary, the strains due to expansion from curved surface to its planar development are $\varepsilon^u(u,v) \geq 0$ and $\varepsilon^v(u,v) \geq 0$, as shown in Figure 3-2. Normal strains are a non-dimensional quantity defined by the ratio of extension or shrinkage of a fiber and its original length. After surface development a strain distribution $\varepsilon^u(u,v)$ along isoparametric line $v = const$, and a strain distribution $\varepsilon^v(u,v)$ along isoparametric

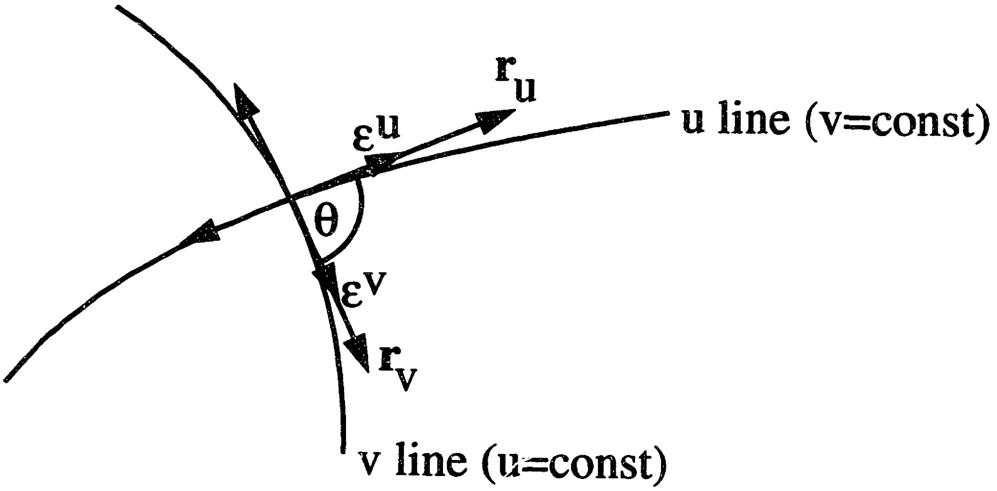

Figure 3-2: Strain distribution produced during surface development

line $u = const$ are produced. Consequently an infinitesimal length $|\mathbf{r}_u du|$ changes to $(1+\varepsilon^u)|\mathbf{r}_u du|$, and an infinitesimal length $|\mathbf{r}_v dv|$ changes to $(1+\varepsilon^v)|\mathbf{r}_v dv|$, according to the definition of strain. Thus we have

$$|\mathbf{R}_u| = (1+\varepsilon^u)|\mathbf{r}_u|, \quad |\mathbf{R}_v| = (1+\varepsilon^v)|\mathbf{r}_v| \tag{3.2}$$

where $\mathbf{R}(u,v)$ is the planar development. The first fundamental form coefficients of the developed surface $\mathbf{R}(u,v)$ are given by

$$e = \mathbf{R}_u \cdot \mathbf{R}_u, \quad f = \mathbf{R}_u \cdot \mathbf{R}_v, \quad g = \mathbf{R}_v \cdot \mathbf{R}_v \tag{3.3}$$

After substituting Equation (3.2) and the relations

$$\mathbf{R}_u \cdot \mathbf{R}_u = |\mathbf{R}_u|^2, \;\; \mathbf{R}_v \cdot \mathbf{R}_v = |\mathbf{R}_v|^2$$

into Equation (3.3), the coefficients of the first fundamental form of the planar developed surface are

$$e = (1+\varepsilon^u)^2 E, \quad f = (1+\varepsilon^u)(1+\varepsilon^v)F, \quad g = (1+\varepsilon^v)^2 G \tag{3.4}$$

Here, in computing f, we assume the angle between $\mathbf{r}_u$ and $\mathbf{r}_v$ does not change after surface development. This is equivalent to ignoring the effect of shear strain.

We then minimize the strains $\varepsilon^u(u,v)$ and $\varepsilon^v(u,v)$ which satisfy the condition that after adding these strains to the doubly curved surface, it maps to a planar shape on which Gaussian curvature is zero. This minimization is done in an integral sense using the squares of the strains. Using Equation (2.31), this results into

$$\begin{aligned} &\min \quad \int\!\!\int_D \{(\varepsilon^u)^2 + (\varepsilon^v)^2\}|\mathbf{r}_u \times \mathbf{r}_v| du dv = \\ &\min \quad \int\!\!\int \{(\varepsilon^u)^2 + (\varepsilon^v)^2\}\sqrt{EG - F^2} du dv \end{aligned} \tag{3.5}$$

$$\begin{aligned} \text{such that} \quad 0 \; = \; & \Bigg\{ e\left(\frac{\partial e}{\partial v}\cdot\frac{\partial g}{\partial v} - 2\frac{\partial f}{\partial u}\cdot\frac{\partial g}{\partial v} + (\frac{\partial g}{\partial u})^2\right) \\ + \; & f\left(\frac{\partial e}{\partial u}\cdot\frac{\partial g}{\partial v} - \frac{\partial e}{\partial v}\cdot\frac{\partial g}{\partial u} - 2\frac{\partial e}{\partial v}\cdot\frac{\partial f}{\partial v} + 4\frac{\partial e}{\partial u}\cdot\frac{\partial f}{\partial v} - 2\frac{\partial f}{\partial u}\cdot\frac{\partial g}{\partial u}\right) \\ + \; & g\left(\frac{\partial e}{\partial u}\cdot\frac{\partial g}{\partial u} - 2\frac{\partial e}{\partial u}\cdot\frac{\partial f}{\partial v} + (\frac{\partial e}{\partial v})^2\right) \\ - \; & 2(eg-f^2)\left(\frac{\partial^2 e}{\partial v^2} - 2\frac{\partial^2 f}{\partial u\partial v} + \frac{\partial^2 g}{\partial u^2}\right)\Bigg\} / 4(eg-f^2)^2 \end{aligned} \tag{3.6}$$

$$\varepsilon^u(u,v) \geq 0; \varepsilon^v(u,v) \geq 0; (u,v) \in D$$

where D denotes the parametric domain. We keep the denominator in Equation (3.6) (i.e. $4(eg-f^2)^2$) so that the constraint essentially forces the Gaussian curvature to be zero. It can be shown that minimizing the strains $\varepsilon^u(u,v)$ and $\varepsilon^v(u,v)$ is equivalent to minimizing the magnitude of the strains $\epsilon^u(u,v)$ and $\epsilon^v(u,v)$. We choose to work with $\varepsilon^u(u,v)$ and $\varepsilon^v(u,v)$ since we are starting from the curved design surface.

Alternatively, we can also use $(\varepsilon^u + \varepsilon^v)$ instead of $\{(\varepsilon^u)^2 + (\varepsilon^v)^2\}$ in the above integral objective function. In this case, the objective function represents the area difference between the doubly curved surface and the planar development to the first order. We use the quadratic objective function $\{(\varepsilon^u)^2+(\varepsilon^v)^2\}$ here instead of the linear one to make the solution easier. After substituting Equations (3.4) into the above formulae, we obtain an optimization problem with respect to $\varepsilon^u(u,v)$ and $\varepsilon^v(u,v)$.

This constrained minimization problem is discretized by using the finite difference method and trapezoidal rule of integration. A grid of $N_g^u \times N_g^v$ points in the parametric domain are used in the discretization. Therefore, the total number of variables is $2N_g^u N_g^v$. It can be shown that the approximations of partial derivatives are not independent, if a finite difference method is applied to all the grid points. A simple example for the one-dimensional case is the discretization of derivative $y' = \frac{dy}{dx}$ at grid point 0, 1, 2:

$$y'|_0 = \frac{y_1 - y_0}{\Delta x}, \quad y'|_1 = \frac{y_2 - y_0}{2\Delta x}, \quad y'|_2 = \frac{y_2 - y_1}{\Delta x}$$

so we have

$$y'|_0 + y'|_2 = 2y'|_1$$

Therefore, to guarantee the independence of each constraint, constraints are imposed at the internal points of the grid, so there are $(N_g^u - 2) \times (N_g^v - 2)$ constraints.

After discretization, the objective function becomes

$$\sum_{i=1}^{N_g^u} \sum_{j=1}^{N_g^v} \alpha_{ij}((\varepsilon_{ij}^u)^2 + (\varepsilon_{ij}^v)^2)\sqrt{E_{ij}G_{i,j} - F_{ij}^2}\Delta u \Delta v$$

where following the trapezoid rule of integration [6]

$$\begin{cases} \alpha_{ij} = 1 & \text{when } 1 < i < N_g^u;\ 1 < j < N_g^v \\ \alpha_{ij} = 0.5 & \text{when } 1 < i < N_g^u;\ j = 1 \text{ or } j = N_g^v \\ \alpha_{ij} = 0.5 & \text{when } i = 1 \text{ or } i = N_g^u;\ 1 < j < N_g^v \\ \alpha_{ij} = 0.25 & \text{when } i = j = 0 \text{ or } i = N_g^u, j = N_g^v \\ \alpha_{ij} = 0.25 & \text{when } i = N_g^u, j = 0 \text{ or } i = 0, j = N_g^v \end{cases} \tag{3.7}$$

We use the central difference method to approximate all the derivatives in Equation (3.6) at internal points of the grid.

$$\frac{\partial e}{\partial u}|_{ij} = \frac{(1+\varepsilon_{i+1,j}^u)^2 E_{i+1,j} - (1+\varepsilon_{i-1,j}^u)^2 E_{i-1,j}}{2\Delta u} \tag{3.8}$$

$$\frac{\partial e}{\partial v}|_{ij} = \frac{(1+\varepsilon_{i,j+1}^u)^2 E_{i,j+1} - (1+\varepsilon_{i,j-1}^u)^2 E_{i,j-1}}{2\Delta v} \tag{3.9}$$

$$\frac{\partial f}{\partial u}|_{ij} = \frac{(1+\varepsilon_{i+1,j}^v)(1+\varepsilon_{i+1,j}^u) F_{i+1,j} - (1+\varepsilon_{i-1,j}^v)(1+\varepsilon_{i-1,j}^u) F_{i-1,j}}{2\Delta u} \tag{3.10}$$

$$\frac{\partial f}{\partial v}|_{ij} = \frac{(1+\varepsilon_{i,j+1}^v)(1+\varepsilon_{i,j+1}^u) F_{i,j+1} - (1+\varepsilon_{i,j-1}^v)(1+\varepsilon_{i,j-1}^u) F_{i,j-1}}{2\Delta v} \tag{3.11}$$

$$\frac{\partial g}{\partial u}|_{ij} = \frac{(1+\varepsilon_{i+1,j}^u)^2 G_{i+1,j} - (1+\varepsilon_{i-1,j}^u)^2 G_{i-1,j}}{2\Delta u} \tag{3.12}$$

$$\frac{\partial g}{\partial v}|_{ij} = \frac{(1+\varepsilon_{i,j+1}^u)^2 G_{i,j+1} - (1+\varepsilon_{i,j-1}^u)^2 G_{i,j-1}}{2\Delta v} \tag{3.13}$$

$$\frac{\partial^2 e}{\partial v^2}|_{ij} = \frac{(1+\varepsilon_{i,j+1}^u)^2 E_{i,j+1} - 2(1+\varepsilon_{i,j}^u)^2 E_{i,j} + (1+\varepsilon_{i,j-1}^u)^2 E_{i,j-1}}{(\Delta v)^2} \tag{3.14}$$

$$
\begin{aligned}
\frac{\partial^2 f}{\partial u \partial v}|_{ij} &= \frac{(1+\varepsilon^v_{i+1,j+1})(1+\varepsilon^u_{i+1,j+1})F_{i+1,j+1}}{4\Delta u \Delta v} \\
&+ \frac{(1+\varepsilon^v_{i-1,j-1})(1+\varepsilon^u_{i-1,j-1})F_{i-1,j-1}}{4\Delta u \Delta v} \\
&- \frac{(1+\varepsilon^v_{i-1,j+1})(1+\varepsilon^u_{i-1,j+1})F_{i-1,j+1}}{4\Delta u \Delta v} \\
&- \frac{(1+\varepsilon^v_{i+1,j-1})(1+\varepsilon^u_{i+1,j-1})F_{i+1,j-1}}{4\Delta u \Delta v} \qquad (3.15) \\
\frac{\partial^2 g}{\partial u^2}|_{ij} &= \frac{(1+\varepsilon^v_{i+1,j})^2 G_{i+1,j} - 2(1+\varepsilon^u_{i,j})^2 G_{i,j} + (1+\varepsilon^v_{i-1,j})^2 G_{i-1,j}}{(\Delta u)^2} \qquad (3.16)
\end{aligned}
$$

As $\Delta u, \Delta v \to 0$, the errors in Equations (3.8)-(3.16) due to central difference approximation are of the order $(\Delta u)^2$ or $(\Delta v)^2$, or $\Delta u \cdot \Delta v$ as is well known [6].

After discretization, we obtain a nonlinear optimization problem with a convex cost function and nonlinear polynomial constraints. This nonlinear programming problem is solved by using the Fortran NAG routine E04VDF [18], which is designed to solve the nonlinear programming problem – the minimization of a smooth nonlinear function subject to a set of constraints on the variables.

3.1.2 Solution method

The minimization problem in Section 3.1.1 is a special case of the following nonlinear programming problem:

$$
\begin{aligned}
&\text{minimize} \quad \psi(q) \\
&\text{subject to} \quad l \leq \begin{bmatrix} q \\ A_L q \\ c(q) \end{bmatrix} \leq u
\end{aligned}
\qquad (3.17)
$$

where $\psi(q)$ is a smooth nonlinear function, A_L is a constant matrix, and $c(q)$ is a vector of smooth nonlinear constraint functions. The matrix A_L and/or the vector $c(q)$ may be empty. This form allows full generality in specifying other types of constraints. In particular, the i (th) constraint may be defined as an equality by setting $l_i = u_i$. If certain bounds are not present, the associated elements of l or u can be set to special values that will be treated as $-\infty$ or $+\infty$. E04VDF is an implementation of a sequential quadratic programming (SQP) method [11] [2]. Let q_0 denote the initial estimate of the solution. During the k(th) "major iteration" of E04VDF (k=0, 1,...), a new estimate is defined by

$$q_{k+1} = q_k + \alpha_k p_k,$$

where the vector p_k is the solution of a QP subproblem, to be described below. The positive scalar α_k is chosen to produce a sufficient decrease in an augmented Lagrange function; the procedure that determines α_k is the line search method.

The QP subproblem that defines p_k is of the form

$$\begin{array}{ll} \text{minimize} & g^T p + \frac{1}{2} p^T H p \\ \text{subject to} & \bar{l} \leq \begin{bmatrix} p \\ A_Q p \end{bmatrix} \leq \bar{u} \end{array} \tag{3.18}$$

where the vector g is the gradient of ψ at q_k; the matrix H is a positive definite quasi-Newton approximation to the Hessian of an augmented Lagrangian function.

Let m_L denote the number of linear constraints (the number of rows in A_L), and m_N denote the number of nonlinear constraints (the dimension of $c(q)$). The matrix A_Q in (3.18) has $m_L + m_N$ rows, and is defined as

$$A_Q = \begin{bmatrix} A_L \\ A_N \end{bmatrix}$$

where A_N is the Jacobian matrix of $c(q)$ evaluated at q_k. Let l in (3.17) be partitioned into three sections: the first n component (denoted by l_B), corresponding to the bound constraints; the next m_L components (denoted by l_L), corresponding to the linear constraints; and the last m_N components (denoted by l_N), corresponding to the nonlinear constraints. The vector $\bar{l}$ in (3.18) is partitioned in the same way, and is defined as

$$\bar{l}_B = l_B - q_k, \;\; \bar{l}_L = l_L - A_L q_k, \;\; \text{and } \bar{l}_N = l_N - c_k$$

where c_k is $c(q)$ evaluated at q_k. The vector $\bar{u}$ is defined in an analogous fashion.

In general, solving the QP subproblem for p_k is itself an iterative procedure. Hence a "minor iteration" of E04VDF corresponds to an iteration within the QP algorithm. In our implementation, the starting point of the minimization is that all the strains are chosen to be zero.

3.1.3 Strain gradients

After solving the strain distribution at the mid-surface, we can determine the gradient of the strains across the thickness. As mentioned in Chapter 2, the strain gradients provide the mechanism of surface curvature in metal forming process. Based on Equation (3.4), and

$$F = \mathbf{r}_u \cdot \mathbf{r}_v = |\mathbf{r}_u||\mathbf{r}_v| sin\theta = \sqrt{E}\sqrt{G} sin\theta \tag{3.19}$$

where θ is the angle between $\mathbf{r}_u$ and $\mathbf{r}_v$, the coefficients of the first fundamental form of the offset surface of planar development at distance d from the mid-surface are:

$$\hat{e} = (1+\hat{\varepsilon}^u)^2 \hat{E}, \quad \hat{f} = (1+\hat{\varepsilon}^u)(1+\hat{\varepsilon}^v)\sqrt{\hat{E}\hat{G}} sin(\hat{\theta} + \Delta\theta), \quad \hat{g} = (1+\hat{\varepsilon}^v)^2 \hat{G} \tag{3.20}$$

where $\hat{E}, \hat{F}, \hat{G}$ are the coefficients of the first fundamental form of the offset surface; $\hat{e}, \hat{f}, \hat{g}$ are the coefficients of the first fundamental form of its development; $\hat{\theta}$ is the angle between isoparametric lines $u = const$ and $v = const$ on the offset surface; $\Delta\theta$ is the change of this angle after development. According to the assumption in this chapter, $\Delta\theta = 0$ at $d = 0$. Ideally, after development, the 2D shape is the same across the thickness, thus we have

$$\frac{\partial \hat{e}}{\partial d} = 0 \tag{3.21}$$

$$\frac{\partial \hat{f}}{\partial d} = 0 \tag{3.22}$$

$$\frac{\partial \hat{g}}{\partial d} = 0 \tag{3.23}$$

After substituting $\hat{e}$ in Equation (3.20) into Equation (3.21), we have

$$2\hat{E}(1+\hat{\varepsilon}^u)\frac{\partial(1+\hat{\varepsilon}^u)}{\partial d} + (1+\hat{\varepsilon}^u)^2\frac{\partial \hat{E}}{\partial d} = 0 \tag{3.24}$$

After dividing the above equation by $(1+\hat{\varepsilon}^u)$ and using Equation (2.39), at $d=0$,

$$2E\frac{\partial(1+\varepsilon^u)}{\partial d}\bigg|_{d=0} + (1+\varepsilon^u)(-2L) = 0, \tag{3.25}$$

so

$$\frac{\partial[ln(1+\varepsilon^u)]}{\partial d}\bigg|_{d=0} = \frac{L}{E} \tag{3.26}$$

Similarly, Equation (3.23) leads to

$$\frac{\partial[ln(1+\varepsilon^v)]}{\partial d}\bigg|_{d=0} = \frac{N}{G} \tag{3.27}$$

After substituting $\hat{f}$ in Equation (3.20) into Equation (3.22), we obtain

$$\sqrt{\hat{E}\hat{G}}sin(\hat{\theta}+\Delta\theta)\frac{\partial(1+\hat{\varepsilon}^u)(1+\hat{\varepsilon}^v)}{\partial d} + (1+\hat{\varepsilon}^u)(1+\hat{\varepsilon}^v)\frac{\partial\left(\sqrt{\hat{E}\hat{G}}sin(\hat{\theta}+\Delta\theta)\right)}{\partial d} = 0 \tag{3.28}$$

At $d = 0$, the above equation results in

$$\sqrt{EG}sin\theta\frac{\partial(1+\varepsilon^u)(1+\varepsilon^v)}{\partial d}\Bigg|_{d=0} + (1+\varepsilon^u)(1+\varepsilon^v)\frac{\partial\left(\sqrt{\hat{E}\hat{G}}sin(\hat{\theta}+\Delta\theta)\right)}{\partial d}\Bigg|_{d=0} = 0 \qquad (3.29)$$

Since

$$\begin{aligned}
\frac{\partial\sqrt{\hat{E}\hat{G}}sin(\hat{\theta}+\Delta\theta)}{\partial d}\Bigg|_{d=0} &= \frac{\partial(\sqrt{\hat{E}\hat{G}}sin\hat{\theta}cos\Delta\theta + \sqrt{\hat{E}\hat{G}}cos\hat{\theta}sin\Delta\theta)}{\partial d}\Bigg|_{d=0} \\
&= \frac{\partial(\hat{F}cos\Delta\theta)}{\partial d}\Bigg|_{d=0} + \frac{\partial(\hat{F}cot\hat{\theta}sin\Delta\theta)}{\partial d}\Bigg|_{d=0} \\
&= \frac{\partial\hat{F}}{\partial d}cos\Delta\theta\;|_{d=0} - \hat{F}sin\Delta\theta\frac{\partial(\Delta\theta)}{\partial d}\Bigg|_{d=0} \\
&+ \hat{F}cot\hat{\theta}cos\Delta\theta\frac{\partial(\Delta\theta)}{\partial d}\Bigg|_{d=0} + \frac{\partial(\hat{F}cot\hat{\theta})}{\partial d}sin\Delta\theta\Bigg|_{d=0} \\
&= -2M + Fcot\theta\frac{\partial(\Delta\theta)}{\partial d}\Bigg|_{d=0} \qquad (3.30)
\end{aligned}$$

from Equation (3.29), we have

$$F\frac{\partial(1+\varepsilon^u)(1+\varepsilon^v)}{\partial d}\Bigg|_{d=0} + (1+\varepsilon^u)(1+\varepsilon^v)\left(-2M + Fcot\theta\frac{\partial(\Delta\theta)}{\partial d}\Bigg|_{d=0}\right) \qquad (3.31)$$

or equivalently,

$$\frac{\partial[ln(1+\varepsilon^u)+ln(1+\varepsilon^v)]}{\partial d}\Bigg|_{d=0} = \frac{2M}{F} - cot\theta\frac{\partial(\Delta\theta)}{\partial d}\Bigg|_{d=0} \qquad (3.32)$$

During the derivation of Equation (3.30), Equation (2.40) is used, as well as the relation

$$\Delta\theta|_{d=0} = 0$$

3.2 Determination of planar developed shape

After solving the nonlinear minimization problem, we obtain the strains ε^u and ε^v at all grid points. We now determine the planar coordinates (X_{ij}, Y_{ij}) of the grid points at the corresponding planar development. Ideally, these coordinates (X_{ij}, Y_{ij}) should satisfy the following equations at all grid points:

$$\mathbf{R}_u \cdot \mathbf{R}_u = e, \quad \mathbf{R}_u \cdot \mathbf{R}_v = f, \quad \mathbf{R}_v \cdot \mathbf{R}_v = g \tag{3.33}$$

where $\mathbf{R} = (X, Y)$, and e, f, g are obtained from Equation (3.4) as functions of u and v. Equations (3.33) can be expressed as

$$\begin{aligned} X_u^2 + Y_u^2 &= e \\ X_u X_v + Y_u Y_v &= f \\ X_v^2 + Y_v^2 &= g \end{aligned} \tag{3.34}$$

After discretization of the above equations (3.34) using finite difference method (central difference for internal points, and forward or backward difference for boundary points), we obtain a system of over-determined nonlinear polynomial equations. Instead of solving the system directly, we solve the following least squares error unconstrained minimization problem

$$\min \sum_{i=1}^{N_g^u} \sum_{j=1}^{N_g^v} (\mathbf{R}_u \cdot \mathbf{R}_u|_{ij} - e_{ij})^2 + (\mathbf{R}_u \cdot \mathbf{R}_v|_{ij} - f_{ij})^2 + (\mathbf{R}_v \cdot \mathbf{R}_v|_{ij} - g_{ij})^2 \tag{3.35}$$

This optimization problem can be solved by using the quasi-Newton method [2] for finding an unconstrained minimization of a sum-of-squares of M_1 nonlinear functions in M_2 variables ($M_1 \geq M_2$). This can be done by using the NAG Fortran library routine E04GBF [18]. In the implementation, rigid body motion of the developed planar shape is prohibited by forcing the coordinates of the grid points:

$$(X_{ij}, Y_{ij})|_{(i=0,j=0)} = (0,0), \quad \text{and} \quad Y_{ij}|_{(i=0,j=1)} = 0$$

The starting points of the minimization are given by

$$(X_{ij}, Y_{ij}) = (\frac{i}{N_g^u}, \frac{j}{N_g^v}) \quad (i = 1, ..., N_g^u; j = 1, ..., N_g^v).$$

Chapter 4

Surface development along principal curvature directions

In Chapter 3, surface development is expressed by tensile strains along the isoparametric lines. The assumption made is that the angle between isoparametric directions remains unchanged after a doubly curved surface is developed into a two-dimensional shape. This assumption is reasonable when the angle between isoparametric directions is large and the strains are small. In the case when the angle between $\mathbf{r}_u$ and $\mathbf{r}_v$ is small at some area of the surface, this assumption may cause errors which can not be ignored.

In this chapter, algorithms for surface development based on strains along principal curvature directions are presented. Since the principal curvature directions are independent of the parametrization of surfaces and are unique except at umbilic points, this surface development is more general. Also, since the angle between two principal curvature directions is a right angle, the assumption that this angle does not change after development is more reasonable.

4.1 Determination of strain field

We assume that the surface is defined by the parametric vector equation (3.1). The surface and its planar development are shown in Figure 3-1. The coefficients of the first fundamental form of the curved surface are given by Equation (2.3). We further assume that during the surface development process, the strains due to expansion from curved surface to its planar development are $\varepsilon^s(u,v) \geq 0$ and $\varepsilon^t(u,v) \geq 0$, along the maximum and minimum principal curvature directions, respectively. Therefore an infinitesimal length $|\mathbf{r}_s ds|$ changes to $(1+\varepsilon^s)|\mathbf{r}_s ds|$, and an infinitesimal length $|\mathbf{r}_t dt|$ changes to $(1+\varepsilon^t)|\mathbf{r}_t dt|$, according to the definition of strain. Thus we have

$$|\mathbf{R}_s| = (1+\varepsilon^s)|\mathbf{r}_s|, \quad |\mathbf{R}_t| = (1+\varepsilon^t)|\mathbf{r}_t| \tag{4.1}$$

where $\mathbf{R}(u,v)$ is the planar development. $\mathbf{R}(u,v)$ can also be considered as a parametric surface with first fundamental form coefficients defined by Equation (3.3). Since

$$\begin{aligned} \mathbf{R}_s \cdot \mathbf{R}_s &= (\mathbf{R}_u u_s + \mathbf{R}_v v_s) \cdot (\mathbf{R}_u u_s + \mathbf{R}_v v_s) \\ &= e u_s^2 + f u_s v_s + g v_s^2 \end{aligned} \tag{4.2}$$

and

$$\begin{aligned} \mathbf{r}_s \cdot \mathbf{r}_s &= (\mathbf{r}_u u_s + \mathbf{r}_v v_s) \cdot (\mathbf{r}_u u_s + \mathbf{r}_v v_s) \\ &= E u_s^2 + F u_s v_s + G v_s^2 \end{aligned} \tag{4.3}$$

using the relations in Equation (4.1, 4.2, 4.3), we obtain

$$e u_s^2 + 2 f u_s v_s + g v_s^2 = (1+\varepsilon^s)^2 (E u_s^2 + 2F u_s v_s + G v_s^2) \tag{4.4}$$

Similarly, along minimum principal curvature direction, we have

$$e u_t^2 + 2 f u_t v_t + g v_t^2 = (1+\varepsilon^t)^2 (E u_t^2 + 2F u_t v_t + G v_t^2) \tag{4.5}$$

We also assume that after development, the principal curvature directions remain orthogonal, which gives

$$\mathbf{R}_s \cdot \mathbf{R}_t = (\mathbf{R}_u u_s + \mathbf{R}_v v_s) \cdot (\mathbf{R}_u u_s + \mathbf{R}_v v_s) = 0 \tag{4.6}$$

Simplifying the above equation gives

$$e u_s u_t + f(u_s v_t + u_t v_s) + g v_s v_t = 0 \tag{4.7}$$

Then we have a system of three linear equations (4.4, 4.5, 4.7) in e, f, g whose solution is given by

$$e = \frac{v_t^2\left[E u_s^2 + 2F u_s v_s + G v_s^2\right](1+\varepsilon^s)^2 + v_s^2\left[E u_t^2 + 2F u_t v_t + G v_t^2\right](1+\varepsilon^t)^2}{(v_s u_t - u_s v_t)^2} \tag{4.8}$$

$$f = -\frac{u_t v_t\left[E u_s^2 + 2F u_s v_s + G v_s^2\right](1+\varepsilon^s)^2 + u_s v_s\left[E u_t^2 + 2F u_t v_t + G v_t^2\right](1+\varepsilon^t)^2}{(v_s u_t - u_s v_t)^2} \tag{4.9}$$

$$g = \frac{u_t^2\left[E u_s^2 + 2F u_s v_s + G v_s^2\right](1+\varepsilon^s)^2 + u_s^2\left[E u_t^2 + 2F u_t v_t + G v_t^2\right](1+\varepsilon^t)^2}{(v_s u_t - u_s v_t)^2} \tag{4.10}$$

We minimize the strains $\varepsilon^s(u,v)$ and $\varepsilon^t(u,v)$ which satisfy the condition that after adding these strains to the doubly curved surface along principal curvature directions, the surface maps to a planar shape on which Gaussian curvature is zero. Using Equation (2.31), this results into

$$\begin{aligned} \min \quad & \int\!\!\int_D \{(\varepsilon^s)^2 + (\varepsilon^t)^2\} |\mathbf{r}_u \times \mathbf{r}_v| du dv = \\ \min \quad & \int\!\!\int \{(\varepsilon^s)^2 + (\varepsilon^t)^2\} \sqrt{EG - F^2} du dv \end{aligned} \tag{4.11}$$

$$\begin{aligned}
\text{such that} \quad 0 \;=\; & \left\{ e\left(\frac{\partial e}{\partial v}\cdot\frac{\partial g}{\partial v} - 2\frac{\partial f}{\partial u}\cdot\frac{\partial g}{\partial v} + (\frac{\partial g}{\partial u})^2\right)\right. \\
+ \; & f\left(\frac{\partial e}{\partial u}\cdot\frac{\partial g}{\partial v} - \frac{\partial e}{\partial v}\cdot\frac{\partial g}{\partial u} - 2\frac{\partial e}{\partial v}\cdot\frac{\partial f}{\partial v} + 4\frac{\partial e}{\partial u}\cdot\frac{\partial f}{\partial v} - 2\frac{\partial f}{\partial u}\cdot\frac{\partial g}{\partial u}\right) \\
+ \; & g\left(\frac{\partial e}{\partial u}\cdot\frac{\partial g}{\partial u} - 2\frac{\partial e}{\partial u}\cdot\frac{\partial f}{\partial v} + (\frac{\partial e}{\partial v})^2\right) \\
- \; & \left. 2(eg-f^2)\left(\frac{\partial^2 e}{\partial v^2} - 2\frac{\partial^2 f}{\partial u \partial v} + \frac{\partial^2 g}{\partial u^2}\right)\right\} / 4(eg-f^2)^2 \qquad (4.12) \\
& \varepsilon^s(u,v) \geq 0; \varepsilon^t(u,v) \geq 0; (u,v) \in D
\end{aligned}$$

where D denotes the parametric domain. After substituting Equations (4.8-4.10) into the above formulae, we obtain an optimization problem with respect to $\varepsilon^s(u,v)$ and $\varepsilon^t(u,v)$. As shown in Chapter 3, this constrained minimization problem is discretized by using the finite difference method and trapezoidal rule of integration. The final formulation is similar to that in Chapter 3 except that $\varepsilon^u, \varepsilon^v$ are replaced by $\varepsilon^s, \varepsilon^t$. Again, the nonlinear constrained minimization problem is solved by using the Fortran NAG routine E04VDF [18].

4.2 Strain gradients

After solving the strain distribution at the mid-surface, we then can determine the gradient of the strains across the thickness. As mentioned in Chapter 2, the strain gradients provide the mechanism of surface curvature in metal forming process. Based on Equations (4.4, 4.5), the relations of the first fundamental form coefficients of the offset surface of distance d along the normal from the mid-surface are:

$$\hat{e}u_s^2 + 2\hat{f}u_sv_s + \hat{g}v_s^2 = (1+\hat{\varepsilon}^s)^2(\hat{E}u_s^2 + 2\hat{F}u_sv_s + \hat{G}v_s^2) \qquad (4.13)$$

$$\hat{e}u_t^2 + 2\hat{f}u_tv_t + \hat{g}v_t^2 = (1+\hat{\varepsilon}^t)^2(\hat{E}u_t^2 + 2\hat{F}u_tv_t + \hat{G}v_t^2) \qquad (4.14)$$

Since after development, the 2D shape is the same across the thickness, we have

$$\frac{\partial(\hat{e}u_s^2 + 2\hat{f}u_s v_s + \hat{g}v_s^2)}{\partial d} = 0 \tag{4.15}$$

$$\frac{\partial(\hat{e}u_t^2 + 2\hat{f}u_t v_t + \hat{g}v_t^2)}{\partial d} = 0 \tag{4.16}$$

By using the expression (2.46), we have

$$\frac{\partial}{\partial d}\left((1+\hat{\varepsilon}^s)^2(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)\right) = 0 \tag{4.17}$$

Expanding the above equation, we obtain at $d = 0$,

$$2(1+\varepsilon^s)\frac{\partial(1+\varepsilon^s)}{\partial d}(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)\bigg|_{d=0} + (1+\varepsilon^s)^2\frac{\partial(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)}{\partial d}\bigg|_{d=0} = 0 \tag{4.18}$$

This results in

$$\frac{\partial[ln(1+\varepsilon^s)]}{\partial d}\bigg|_{d=0} = -\frac{1}{\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s}\frac{\partial(\hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_s)}{\partial d}\bigg|_{d=0} = \kappa_{max} \tag{4.19}$$

The last equality comes from Equation (2.42). Similarly,

$$\frac{\partial[ln(1+\varepsilon^t)]}{\partial d}\bigg|_{d=0} = -\frac{1}{\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t}\frac{\partial(\hat{\mathbf{r}}_t \cdot \hat{\mathbf{r}}_t)}{\partial d}\bigg|_{d=0} = \kappa_{min} \tag{4.20}$$

After solving ε^s and ε^t by the first nonlinear constrained optimization, and subsequently e, f, g in the second optimization, the reverse process of determining E, F, G can not be achieved by solving E, F, G from the three equations (4.8, 4.9, 4.10). We have the following results:

Theorem 4.2.1 *The determinant of the linear system (4.8, 4.9, 4.10) for solving for E, F, G is 0.*

Proof: Denote $\alpha = (1+\varepsilon^s)^2$ and $\beta = (1+\varepsilon^t)^2$. The determinant of the linear system

(4.8, 4.9, 4.10) for solving for E, F, G is

$$det = \begin{vmatrix} u_s^2v_t^2\alpha + u_t^2v_s^2\beta & 2u_sv_sv_t^2\alpha + 2u_tv_tv_s^2\beta & v_s^2v_t^2\alpha + v_s^2v_t^2\beta \\ u_tv_tu_s^2\alpha + u_sv_su_t^2\beta & 2u_sv_su_tv_t(\alpha + \beta) & u_tv_tv_s^2\alpha + u_sv_sv_t^2\beta \\ u_s^2v_s^2(\alpha + \beta) & 2u_sv_su_t^2\alpha + 2u_tv_tu_s^2\beta & u_t^2v_s^2\alpha + u_s^2v_t^2\beta \end{vmatrix} \qquad (4.21)$$

We expand the determinant in Equation (4.21) according to the second row, and cancelling out similar terms, we obtain

$$\begin{aligned} det &= 2(u_tv_tu_s^2\alpha + u_sv_su_t^2\beta)(u_s^3v_sv_t^4 + v_s^4u_t^3v_t - u_sv_s^3u_t^2v_t^2 - u_s^2v_s^2u_tv_t^3)\alpha\beta \\ &- 2(u_sv_su_tv_t\alpha + u_sv_su_tv_t\beta)(u_s^4v_t^4 + u_t^4v_s^4 - 2u_s^2u_t^2v_s^2v_t^2)\alpha\beta \\ &+ 2(u_tv_tv_s^2\alpha + u_sv_sv_t^2\beta)(u_s^4u_tv_t^3 + u_sv_s^3u_t^4 - u_s^2v_s^2u_t^3v_t - u_s^3v_su_t^2v_t^2)\alpha\beta \\ &= c_1\alpha^2\beta + c_2\alpha\beta^2 \qquad (4.22) \end{aligned}$$

where the coefficients

$$\begin{aligned} c_1 &= 2u_tv_tu_s^2(u_s^3v_sv_t^4 + v_s^4u_t^3v_t - u_sv_s^3u_t^2v_t^2 - u_s^2v_s^2u_tv_t^3) \\ &- 2u_sv_su_tv_t(u_s^4v_t^4 + u_t^4v_s^4 - 2u_s^2u_t^2v_s^2v_t^2) \\ &+ 2u_tv_tv_s^2(u_s^4u_tv_t^3 + u_sv_s^3u_t^4 - u_s^2v_s^2u_t^3v_t - u_s^3v_su_t^2v_t^2) \\ &= 0 \qquad (4.23) \\ c_2 &= 2u_sv_su_t^2(u_s^3v_sv_t^4 + v_s^4u_t^3v_t - u_sv_s^3u_t^2v_t^2 - u_s^2v_s^2u_tv_t^3) \\ &- 2u_sv_su_tv_t(u_s^4v_t^4 + u_t^4v_s^4 - 2u_s^2u_t^2v_s^2v_t^2) \\ &+ 2u_sv_sv_t^2(u_s^4u_tv_t^3 + u_sv_s^3u_t^4 - u_s^2v_s^2u_t^3v_t - u_s^3v_su_t^2v_t^2) \\ &= 0 \qquad (4.24) \end{aligned}$$

The last equalities of Equations (4.23) and (4.24) are obtained by expanding all terms.

Therefore,

$$det = 0$$

■

Geometrically, this means that there are multiple solutions for E, F, G to Equations (4.8, 4.9, 4.10) if other parameters are given; i.e., if only the stretches at two principal directions are given, the stretches along two isoparametric lines and the change of the angle between them are indeterminate.

We thus choose to determine E, F, G from the following equations instead, i.e., we force the principal directions in the parametric domain to be the same for both the reconstructed surface and the original surface:

$$\left[Eu_s^2 + 2Fu_sv_s + Gv_s^2\right](1+\varepsilon^s)^2 = \left[eu_s^2 + 2fu_sv_s + gv_s^2\right] \tag{4.25}$$

$$\left[Eu_t^2 + 2Fu_tv_t + Gv_t^2\right](1+\varepsilon^t)^2 = \left[eu_t^2 + 2fu_tv_t + gv_t^2\right] \tag{4.26}$$

$$Eu_su_t + F(v_su_t + u_sv_t) + Gu_su_t = 0 \tag{4.27}$$

where Equations (4.25) and (4.26) come from Equations (4.4) and (4.5), and Equation (4.27) is the orthogonality condition for s, t being the principal directions. The reconstructed surface is the surface which is produced from the planar development and the strains by metal forming, which is the reverse process of surface development.

The obtained E, F, G can be used to reconstruct the surface to see the error due to surface development and reconstruction processes, which is equivalent to the error between the manufactured surface and the design surface.

Chapter 5

Analysis of the algorithms

In this chapter, we discuss convergence, complexity, and accuracy of the algorithms presented in Chapters 3 and 4. The symbols in this chapter follow the general practice in the optimization literature, especially, vectors are expressed as a normal letter instead of a bold one.

5.1 Convergence analysis

In this section, we analyze the convergence of the surface development algorithm, i.e, the convergence of the discrete solution to the continuous solution $\varepsilon^u(u,v), \varepsilon^v(u,v)$ to the optimization problem (3.5).

Theorem 5.1.1 *The error on the right side of Equation (3.6) due to discretization is $O((\Delta u)^2, (\Delta v)^2, \Delta u \Delta v)$.*

Proof: Let $h(\varepsilon^u(u,v), \varepsilon^v(u,v))$ be the right side of Equation (3.6), then discretization introduces errors $O((\Delta u)^2)$, $O((\Delta v)^2)$, or $O(\Delta u \Delta v)$ into all the derivatives in $h(\varepsilon^u(u,v), \varepsilon^v(u,v))$. By substituting $\frac{\partial e}{\partial u}$ with $\frac{\partial e}{\partial u} + O((\Delta u)^2)$, $\frac{\partial e}{\partial v}$ with $\frac{\partial e}{\partial v} + O((\Delta v)^2)$, etc. and expand the right side of Equation (3.6), we obtain

$$\bar{h}(\varepsilon^u(u,v), \varepsilon^v(u,v)) = h(\varepsilon^u(u,v), \varepsilon^v(u,v)) + O((\Delta u)^2, (\Delta v)^2, \Delta u \Delta v) \tag{5.1}$$

where $\bar{h}$ is the constraint after discretization. ∎

Theorem 5.1.2 *The equations and inequalities satisfied by the discrete solution $\varepsilon^u|_{ij}$, $\varepsilon^v|_{ij}$ to the optimization problem (3.5 - 3.6) converge to those satisfied by the continuous solution $\varepsilon^u(u,v), \varepsilon^v(u,v)$ as $\Delta u \to 0, \Delta v \to 0$.*

Proof: Before discretization, problem (3.5 - 3.6) is a constrained minimization of a functional, i.e., function of functions. Let

$$J(\varepsilon^u(u,v), \varepsilon^v(u,v)) = \int\int \{(\varepsilon^u)^2 + (\varepsilon^v)^2\}\sqrt{EG - F^2}dudv \tag{5.2}$$

be the objective functional, and the constraints be

$$h(\varepsilon^u(u,v), \varepsilon^v(u,v)) = 0 \tag{5.3}$$

$$\varepsilon^u(u,v) \geq 0 \tag{5.4}$$

$$\varepsilon^v(u,v) \geq 0 \tag{5.5}$$

where $h(\varepsilon^u(u,v), \varepsilon^v(u,v))$ is the right side of Equation (3.6), which is also a functional of $\varepsilon^u(u,v), \varepsilon^v(u,v)$. Let us denote

$$\phi(\varepsilon^u, \varepsilon^v) = \{(\varepsilon^u)^2 + (\varepsilon^v)^2\}\sqrt{EG - F^2} \tag{5.6}$$

then the constrained minimization is equivalent to minimize [9]

$$\tilde{J}(\varepsilon^u(u,v), \varepsilon^v(u,v)) = \int\int [\phi(\varepsilon^u, \varepsilon^v) + \lambda(u,v)h(\varepsilon^u, \varepsilon^v) + \mu_1(u,v)\varepsilon^u + \mu_2(u,v)\varepsilon^v]\, dudv \tag{5.7}$$

where $\lambda(u,v)$, $\mu_1(u,v)$, $\mu_2(u,v)$ are Lagrange multipliers. From the theory of variational calculus, the stationary functions $\varepsilon^u(u,v), \varepsilon^v(u,v)$ and the corresponding Lagrange multipliers $\lambda(u,v)$, $\mu_1(u,v)$, $\mu_2(u,v)$ of problem (3.5 - 3.6) satisfy [9]

$$\phi_{\varepsilon^u} + \lambda h_{\varepsilon^u} + \mu_1 + \lambda\frac{\partial}{\partial u}(h_{\varepsilon^u_u}) + \lambda\frac{\partial}{\partial v}(h_{\varepsilon^u_v}) = 0 \tag{5.8}$$

$$\phi_{\varepsilon^v} + \lambda h_{\varepsilon^v} + \mu_2 + \lambda\frac{\partial}{\partial u}(h_{\varepsilon^v_u}) + \lambda\frac{\partial}{\partial v}(h_{\varepsilon^v_v}) = 0 \tag{5.9}$$

$$h(u,v,\varepsilon^u,\varepsilon^v,\varepsilon^u_u,\varepsilon^u_v,\varepsilon^v_u,\varepsilon^v_v) = 0 \tag{5.10}$$

$$\varepsilon^u\mu_1 = 0 \tag{5.11}$$

$$\varepsilon^v\mu_2 = 0 \tag{5.12}$$

$$\varepsilon^u \geq 0 \quad \varepsilon^v \geq 0 \tag{5.13}$$

$$\mu_1 \leq 0 \quad \mu_2 \leq 0 \tag{5.14}$$

where ε^u_u, ε^u_v, ε^v_u, ε^v_v denote $\frac{\partial\varepsilon^u}{\partial u}$, $\frac{\partial\varepsilon^u}{\partial v}$, $\frac{\partial\varepsilon^v}{\partial u}$, $\frac{\partial\varepsilon^v}{\partial v}$ respectively, and $h_{\varepsilon^u_u}$, $h_{\varepsilon^u_v}$, $h_{\varepsilon^v_u}$, $h_{\varepsilon^v_v}$ denote the derivatives of h with respect to the corresponding subscript.

If we denote

$$\frac{Dh}{D\varepsilon^u} = \frac{\partial h}{\partial\varepsilon^u} + \frac{\partial}{\partial u}(h_{\varepsilon^u_u}) + \frac{\partial}{\partial v}(h_{\varepsilon^u_v}) \tag{5.15}$$

$$\frac{Dh}{D\varepsilon^v} = \frac{\partial h}{\partial\varepsilon^v} + \frac{\partial}{\partial u}(h_{\varepsilon^v_u}) + \frac{\partial}{\partial v}(h_{\varepsilon^v_v}) \tag{5.16}$$

then the system of equations and inequalities (5.8-5.14) becomes

$$\phi_{\varepsilon^u}(u,v) + \lambda(u,v)\frac{Dh}{D\varepsilon^u}(u,v) + \mu_1(u,v) = 0 \tag{5.17}$$

$$\phi_{\varepsilon^v}(u,v) + \lambda(u,v)\frac{Dh}{D\varepsilon^v}(u,v) + \mu_2(u,v) = 0 \tag{5.18}$$

$$h(\varepsilon^u,\varepsilon^v) = 0 \tag{5.19}$$

$$\varepsilon^u(u,v)\mu_1(u,v) = 0 \tag{5.20}$$

$$\varepsilon^v(u,v)\mu_2(u,v) = 0 \tag{5.21}$$

$$\varepsilon^u(u,v) \geq 0 \quad \varepsilon^v(u,v) \geq 0 \tag{5.22}$$

$$\mu_1(u,v) \leq 0 \quad \mu_2(u,v) \leq 0 \tag{5.23}$$

These are the equations and inequalities that the solution of the discrete version of the optimization problem satisfies approximately. Let $\bar{\varepsilon}^u_{ij}$, $\bar{\varepsilon}^v_{ij}$, $\bar{\lambda}_{ij}$, $\bar{\mu}_1|_{ij}$, $\bar{\mu}_2|_{ij}$ be this solution, then it satisfies

$$\phi_{\varepsilon^u}|_{ij} + \bar{\lambda}|_{ij}\frac{Dh}{D\varepsilon^u}|_{ij} + \bar{\mu}_1|_{ij} + O((\Delta u)^2) = 0 \tag{5.24}$$

$$\phi_{\varepsilon^v}|_{ij} + \bar{\lambda}|_{ij}\frac{Dh}{D\varepsilon^v}|_{ij} + \bar{\mu}_2|_{ij} + O((\Delta v)^2) = 0 \tag{5.25}$$

$$h(u_{ij}, v_{ij}, \bar{\varepsilon}^u_{ij}, \bar{\varepsilon}^v_{ij}) + O((\Delta u)^2, (\Delta v)^2, \Delta u \Delta v)) = 0 \tag{5.26}$$

$$\bar{\varepsilon}^u_{ij}\bar{\mu}_1|_{ij} = 0 \tag{5.27}$$

$$\bar{\varepsilon}^v_{ij}\bar{\mu}_2|_{ij} = 0 \tag{5.28}$$

$$\bar{\varepsilon}^u|_{ij} \geq 0 \quad \bar{\varepsilon}^v|_{ij} \geq 0 \tag{5.29}$$

$$\bar{\mu}_1|_{ij} \leq 0 \quad \bar{\mu}_2|_{ij} \leq 0 \tag{5.30}$$

Therefore, as $\Delta u \to 0, \Delta v \to 0$, these equations and inequalities converge to those satisfied by the continuous optimization solution. ■

We now discuss about the errors of the variables ε^u_{ij}, ε^v_{ij} and the objective function due to the error in constraints coming from discretization.

If the solution to the system of equations and inequalities (5.17-5.23) is unique, then this solution evaluated at grid points (u_i, v_j) satisfies the following system of equations and inequalities:

$$\phi_{\varepsilon^u}(u_i, v_j) + \lambda(u_i, v_j)\frac{Dh}{D\varepsilon^u}(u_i, v_j) + \mu_1(u_i, v_j) = 0 \tag{5.31}$$

$$\phi_{\varepsilon^v}(u_i, v_j) + \lambda(u_i, v_j)\frac{Dh}{D\varepsilon^v}(u_i, v_j) + \mu_2(u_i, v_j) = 0 \tag{5.32}$$

$$h(\varepsilon^u(u_i, v_j), \varepsilon^v(u_i, v_j)) = 0 \tag{5.33}$$

$$\varepsilon^u(u_i, v_j)\mu_1(u_i, v_j) = 0 \tag{5.34}$$

$$\varepsilon^v(u_i, v_j)\mu_2(u_i, v_j) = 0 \tag{5.35}$$

$$\varepsilon^u(u_i, v_j) \geq 0 \quad \varepsilon^v(u_i, v_j) \geq 0 \tag{5.36}$$

$$\mu_1(u_i, v_j) \leq 0 \quad \mu_2(u_i, v_j) \leq 0 \tag{5.37}$$

Denote V^* all the free variables; i.e., variables $\varepsilon^u(u_i, v_j)$, $\varepsilon^v(u_i, v_j)$, $\mu_1(u_i, v_j)$, $\mu_2(u_i, v_j)$ which are nonzero, and all $\lambda(u_i, v_j)$. Further denote the equations that these free variables satisfy as

$$A(V^*) = 0 \tag{5.38}$$

Then by keeping the zero variables in the system (5.24-5.30) be the same as those in (5.31-5.37), the remaining variables satisfy

$$A(V) = A(V^*) + \nabla A(V^*)'(V - V^*) + O(|V - V^*|^2) = O((\Delta u)^2, (\Delta v)^2, \Delta u \Delta v) \tag{5.39}$$

Therefore, when the Jacobian matrix $\nabla A(V^*)$ is nonsingular, we have

$$V - V^* = O((\Delta u)^2, (\Delta v)^2, \Delta u \Delta v) \tag{5.40}$$

This means that the discrete solution $\varepsilon^u|_{ij}, \varepsilon^v|_{ij}$ converges to the continuous solution $\varepsilon^u(u_i, v_j), \varepsilon^v(u_i, v_j)$ as $\Delta u \to 0, \Delta v \to 0$, and the error of variables ε^u_{ij}, ε^v_{ij} due to discretization error is of the order $O((\Delta u)^2, (\Delta v)^2, \Delta u \Delta v)$. The error of e, f, g in (3.33) is also of the second order, which means the error of the final X, Y coordinates of the developed shape is of the second order with respect to $\Delta u = \Delta v$. Since the error of variables ε^u_{ij}, ε^v_{ij} due to discretization error is of the second order, the error of the objective function is also of the second order.

5.2 Complexity analysis

5.2.1 The algorithm for strain determination

As shown in Section 3.1.2, the first constrained optimization (strain determination algorithm) involves solution of a sequential quadratic programming problems. Operations include

1. In the major iteration, operations mainly include

   (1) Formulation of the H matrix, a positive definite quasi-Newton approximation to the Hessian of an augmented Lagrangian function L_A.

(2) Computation of the vector g, gradient of $\psi(q)$.

(3) Determination of the step α_k by line search.

The dominant operations are in formulation of the H matrix by quasi-Newton method:

$$\begin{aligned}
p^k &= q^{k+1} - q^k \\
y^k &= \nabla L_A(q^{k+1}) - \nabla L_A(q^k) \\
C^k &= \frac{p^k(p^k)'}{(p^k)'y^k} - \frac{H^k y^k y^{k'} H^k}{y^{k'} H^k y^k} \\
H^{k+1} &= H^k + C^k
\end{aligned}$$

and $H^{(0)}$ is an arbitrary positive definite matrix.

The total number of operations is $O(n^3)$, where n is the dimension of q. In our problem, $n = 2(N_g^u - 2)(N_g^v - 2)$, so the total number of operations is $O((N_g^u N_g^v)^3)$.

2. Solving the QP subproblem (3.18). The method has two distinct phases. In the first (the LP phase), an iterative procedure is carried out to determine a feasible point. The second phase (the QP phase) generates a sequence of feasible iterates in order to minimize the quadratic objective function. In both phases, a subset of the constraints - called the working set - is used to define the search direction at each iteration; typically, the working set includes constraints that are satisfied within the corresponding tolerance.

   Let p_k denote the estimate of the solution at kth iteration, and the next iterate be defined by

$$p_{k+1} = p_k + \beta_k \bar{p}_k$$

   The general procedures for a typical iteration in the QP phase are [18]

   (1) Assume that the working (active) set contains t_k linearly independent constraints, and C_k (size $t_k \times n$) denotes the matrix of coefficients of the bounds and general constraints in the current working set.

(2) Solve for Z_k, a $n \times n_z$ matrix whose columns form a basis for the null space of C_k, so that $C_k Z_k = 0$. $n_z = n - t_k$. Z_k is obtained from the TQ factorization of C_k in which C_k is represented as

$$C_k Q_k = (0 \; T_k)$$

and Z_k is taken as the first n_z columns of Q_k.

(3) The search direction

$$\bar{p}_k = Z_k p_z$$

and the vector p_z is obtained by solving the equations

$$Z_k^T H Z_k p_z = -Z_k^T (c + H p_k)$$

This is solved by using the Cholesky factorization

$$Z_k^T H Z_k = L_k^T L_k$$

where L_k is upper triangular.

(4) Determine β_k by line search.

The LP phase for determining a feasible point need only be carried out once in each major iteration, and possibly less than once, when the solution from the previous major iteration is feasible. The cost of linear programming depends on the method used; so let us assume the simplex method is used. Though the worst case performance of the method is exponential [3], this method operates on a $m \times n$ matrix to generate a solution usually in $O(m^2 n)$ time [8]. The constraint matrix in problem (3.18) is of the size $[2N_g^u N_g^v + (N_g^u - 2)(N_g^v - 2)] \times 2N_g^u N_g^v$, therefore, the LP takes $O((N_g^u N_g^v)^3)$ time.

In the QP phase, step (2) takes $O(n^3)$ operations for the first iteration, and $O(n^2)$ operations for remaining iterations. Step (3) takes $O(n_z n^2)$ operations for computing the matrix $Z_k H Z_k$ in the first iteration, and $O(n_z n)$ for updating the matrix $Z_k H Z_k$

in each of the remaining iterations. Cholesky factorization takes $O(n_z^3)$ operations in the first iteration [5], and updating it takes $O(n_z^2)$ operations in each of the remaining iterations. After Cholesky factorization, solving for p_z takes $O(n_z^2)$ operations in each iteration. Since in each of the early minor iterations, a bound constraint leaves the active set, and when the active constraint set is the same as that for the solution, the convergence is superlinear, the average number of minor iterations should be $O(n)$. Thus steps (2) and (3) both take $O(n^3)$ time. For the strain determination problem, $n = 2N_g^u N_g^v$, therefore, the QP phase takes $O(n^3) = O((N_g^u N_g^v)^3)$ time.

In summary, each major iteration takes $O((N_g^u N_g^v)^3)$ time.

5.2.2 The algorithm for planar developed shape determination

The unconstrained minimization problem (3.35) is a least square minimization of $m = 3N_g^u N_g^v$ equations with $n = 2N_g^u N_g^v$ variables. The general algorithm for nonlinear least-square problems [10]

$$F(x) = \sum_{i=1}^{m}[f_i(x)]^2, \quad x \in E^n, \quad m \le n$$

where the gradient vector $g(x)$ and Hessian matrix $G(x)$ of $F(x)$ are given by $2J(x)^T f(x)$ and $2(J(x)^T J(x) + B(x))$ respectively, $J(x)$ is the $m \times n$ Jacobian matrix of $f(x) = (f_1(x), ..., f_m(x))^T$, $B(x) = \Sigma_{i=1}^{m} f_i(x) G_i(x)$ and $G_i(x)$ is the Hessian matrix of $f_i(x)$, follows:

1. Set $x^{(0)}$, $f^{(0)}$, $J^{(0)}$ and $g^{(0)} = 2J^{(0)T} f^{(0)}$.

2. If $x^{(k)}$ is an adequate approximation to a stationary point, the algorithm is terminated. Otherwise continue at step 3.

3. Compute the singular-value decomposition of $J^{(k)}$:

$$J^{(k)} = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T$$

4. Select r, the grade of $J^{(k)}$. The value of r is generally equal to the number of dominant singular values of J and is rarely significantly less than n.

5. Compute the Gauss-Newton direction in the space spanned by V_1:

$$p_1 = -V_1 S_1^{-1} f_1$$

6. If a correction to the Gauss-Newton direction is not required, set $p_2 = 0$ and continue at step 7. Otherwise compute or approximate the matrices $Y = V_2^T B^{(k)}$ and $Q = YV_2$. Use the modified LDL^T factorization to solve the equations

$$(S_2^2 + Q)y = -S_2 f_2 - Y p_1,$$

and set $p_2 = V_2 y$.

The matrix $B^{(k)}$ is approximated by $H^{(k)}$ using a quasi-Newton approximation:

$$H^{(k+1)} = H^{(k)} + C^{(k)}$$

$$H^{(0)} = 0, \; C^{(k)} = \frac{1}{\alpha^{(k)} y^{(k)T} p^{(k)}} y^{(k)} y^{(k)T} - \frac{1}{p^{(k)} W^{(k)} p^{(k)}} W^{(k)} p^{(k)} p^{(k)T} W^{(k)},$$

where $W^{(k)} = J^{(k+1)T} J^{(k+1)} + H^{(k)}$ and $y^{(k)} = J^{(k+1)T} f^{(k+1)} - J^{(k)T} f^{(k)}$.

7. Set $p^{(k)} = p_1 + p_2$. Let $\sigma (\sigma > 0)$ be a small pre-assigned scalar:
if $-g^{(k)} p^{(k)} / (||g^{(k)}|| \; ||p^{(k)}||) < \sigma$ and $r > 0$ then set $r = 0$ and return to step 5 to recompute $p^{(k)}$.

8. Compute a step length $\alpha^{(k)}$ such that

$$F^{(k)} - F(x^{(k)} + \alpha^{(k)} p^{(k)}) > \phi \left(-g^{(k)} p^{(k)} / (||g^{(k)}|| \; ||p^{(k)}||) \right)$$

where $\phi(t)$ is a function such that $\lim_{k \to \infty} \phi(t_k) = 0$ implies $\lim_{k \to \infty} t_k = 0$

9. Compute $x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)}$, $f^{(k+1)}$, $J^{(k+1)}$ and $g^{(k+1)}$; set $k = k + 1$ and otinue at step 2.

We can see that the relatively expensive steps are steps 1, 3, 5, 6. Step 1 takes $O(nm^2)$ operations; step 3 is the singular value decomposition, and is usually solved by a two-phased computation: first reduce the matrix to bidiagonal form, and then the bidiagonal matrix is diagonalized [21]. Phase 1 involves $O(mn^2)$ operations. Phase 2 in principle requires an infinite number of operations, but the standard algorithms converge superlinearly, so convergence is achieved in $O(n)$ iterations, and each iteration requires only $O(n)$ operations. Phase 2 therefore requires $O(n^2)$ operations all together. Thus step 3 requires $O(mn^2)$ operations. In step 5, since matrix S_1 is a diagonal matrix, only $O(nr) = O(n^2)$ operations are needed. In step 6, computing $C^{(k)}$ requires $O(mn^2)$ operations, and solving the equation by using LDL^T factorization requires $O((n-r)^3) = O(n^3)$. Therefore, since $m = 3N_g^u N_g^v$ and $n = 2N_g^u N_g^v$, overall, in each iteration of the non-constrained minimization, the required number of operations is $O((N_g^u N_g^v)^3)$.

Chapter 6

Examples

In this chapter, we demonstrate how the algorithms work for surface development based on strains along isoparametric lines and along lines of curvature. The surfaces in the examples include surfaces with all elliptical points (positive Gaussian curvature), surfaces with all hyperbolic points (negative Gaussian curvature), and more complex surfaces that have both positive and negative Gaussian curvature regions. Compared to the surfaces in shipbuilding industry, the surfaces used in the examples have much larger absolute value of Gaussian curvature and hence they are more difficult to develop. All examples were run on a Silicon Graphics workstation O_2 R5000 running at 200 MHz.

6.1 Results on surface development along isoparametric lines

6.1.1 Example 1

We use this example to demonstrate the complexity and accuracy of the algorithms.

The first surface is a bicubic Bézier surface $\mathbf{r}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3}\mathbf{r}_{ij}B_{i,3}(u)B_{j,3}(v)$, where all the points on the surface are elliptic. The control points are given as follows:

$(0,0,0)$	$(0,1/3,0.15)$	$(0,2/3,0.15)$	$(0,1,0)$
$(1/3,0,0.25)$	$(1/3,1/3,0.5)$	$(1/3,2/3,0.5)$	$(1/3,1,0.25)$
$(2/3,0,0.25)$	$(2/3,1/3,0.5)$	$(2/3,2/3,0.5)$	$(2/3,1,0.25)$
$(1,0,0)$	$(1,1/3,0.15)$	$(1,2/3,0.15)$	$(1,1,0)$

Equivalently, this surface can be expressed as an explicit surface

$$z = 0.35625 - 0.975(x-0.5)^2 - 0.675(y-0.5)^2 + 0.9(x-0.5)^2(y-0.5)^2$$

where $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

The surface along with its control polygon is shown in Figure 6-1. The constrained minimization problem (3.5-3.6) is discretized at 13 × 13 grid points which are equally distributed in u, v domain. Figure 6-2 shows the strain distribution after the constrained minimization problem was solved using tolerances 10^{-5} for the constraints and 10^{-4} for the objective function. The strains are scaled to fit into the figure. The extreme values of the strain field are located at $(u,v) = (0,0.5)$ or $(u,v) = (1.0,0.5)$ with $(\varepsilon^u,\varepsilon^v) = (0.0012101, 0.203391)$, and at $(u,v) = (0.5,0)$ or $(u,v) = (0.5,1.0)$

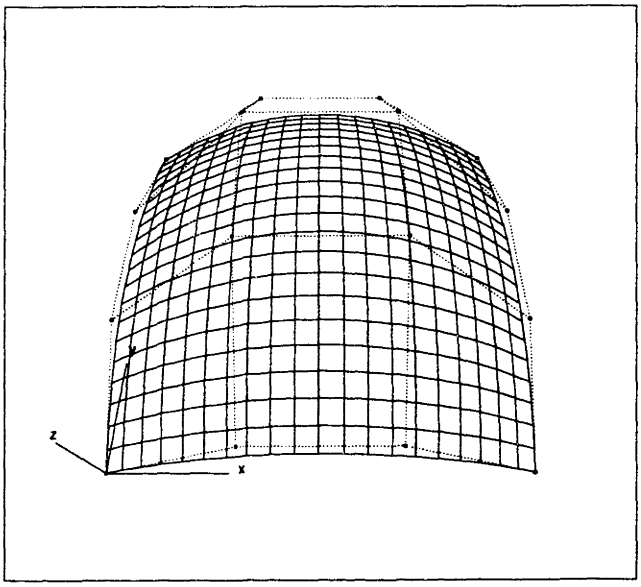

Figure 6-1: The bi-cubic Bézier surface in Example 1

with $(\varepsilon^u, \varepsilon^v) = (0.242961, 0.00136569)$. The objective function converges to the value of 6.3857×10^{-3} at the solution, and all the constraints are within the tolerance of 1.0×10^{-5}.

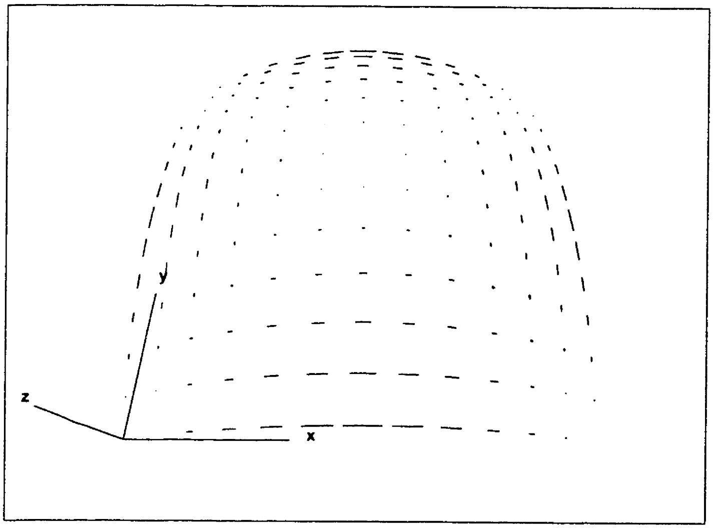

Figure 6-2: The strain distribution of the surface in Example 1, developed along isoparametric lines (length of line segments at grid points shows the scaled magnitude of the two strains ε^u and ε^v)

After development, the planar shape is shown in Figure 6-3. The four end points have coordinates of (0,0), (-0.14799,1.10615), (1.18199,0.15838), (1.03399,1.26452) respectively. The final value of the formula (3.35) is less than 10^{-4} of the value $\sum_{i=1}^{N_g^u} \sum_{j=1}^{N_g^v} (e_{ij}^2 + f_{ij}^2 + g_{ij}^2)$, the sum of the squares of the right side of system (3.34) at all grid points.

Figure 6-4 and Figure 6-5 show the ideal strain gradient $\frac{\partial[ln(1+\varepsilon^u)]}{\partial d}$ and $\frac{\partial[ln(1+\varepsilon^v)]}{\partial d}$ evaluated at grid points.

In order to estimate the accuracy of this surface development algorithm, we computed E, F, G based on the e, f, g and the strains ε^u, ε^v and strain gradients we

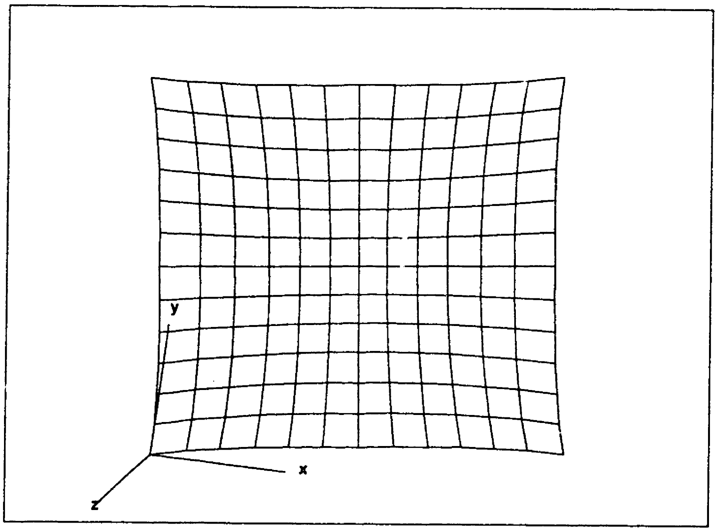

Figure 6-3: The planar development of the surface in Example 1, developed along isoparametric lines

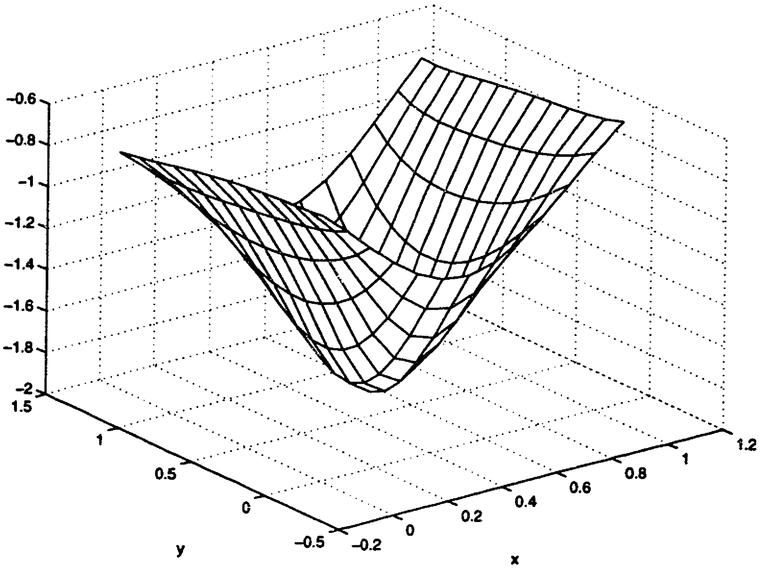

Figure 6-4: Logarithmic strain gradient along u-isoparametric line of the surface in Example 1, developed along isoparametric lines

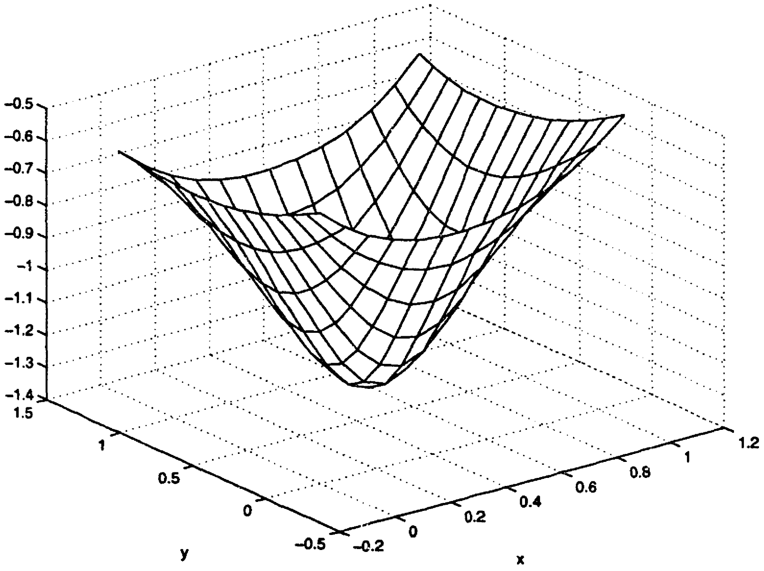

Figure 6-5: Logarithmic strain gradient along v-isoparametric line of the surface in Example 1, developed along isoparametric lines

obtained after the surface development. Then we solvcd a reverse problem

$$
\begin{aligned}
\mathbf{r}_u \cdot \mathbf{r}_u &= E \\
\mathbf{r}_u \cdot \mathbf{r}_v &= F \\
\mathbf{r}_v \cdot \mathbf{r}_v &= G \\
\mathbf{r}_{uu} \cdot (\mathbf{r}_u \times \mathbf{r}_v) &= L\sqrt{EG - F^2} \\
\mathbf{r}_{uv} \cdot (\mathbf{r}_u \times \mathbf{r}_v) &= M\sqrt{EG - F^2} \\
\mathbf{r}_{vv} \cdot (\mathbf{r}_u \times \mathbf{r}_v) &= N\sqrt{EG - F^2}
\end{aligned}
$$

to obtain $\mathbf{r} = (x, y, z)$ using the least squares error minimization as follows:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{N_u}\sum_{j=1}^{N_v} (\mathbf{r}_u \cdot \mathbf{r}_u|_{ij} - E_{ij})^2 + (\mathbf{r}_u \cdot \mathbf{r}_v|_{ij} - F_{ij})^2 + (\mathbf{r}_v \cdot \mathbf{r}_v|_{ij} - G_{ij})^2 \\
+ \quad & \left(\mathbf{r}_{uu} \cdot (\mathbf{r}_u \times \mathbf{r}_v)|_{ij} - L_{ij}\sqrt{E_{ij}G_{ij} - F_{ij}^2}\right)^2 \\
+ \quad & \left(\mathbf{r}_{uv} \cdot (\mathbf{r}_u \times \mathbf{r}_v)|_{ij} - M_{ij}\sqrt{E_{ij}G_{ij} - F_{ij}^2}\right)^2 \\
+ \quad & \left(\mathbf{r}_{vv} \cdot (\mathbf{r}_u \times \mathbf{r}_v)|_{ij} - N_{ij}\sqrt{E_{ij}G_{ij} - F_{ij}^2}\right)^2
\end{aligned} \tag{6.1}
$$

We fixed 6 variables to avoid the rigid body motion in the surface reconstruction process. We set $(x, y, z) = (0,0,0)$ at $(u, v) = (0,0)$, $(x, y) = (0,0)$ at $(u, v) = (0,1)$, and $z = 0$ at $(u, v) = (1,0)$. After solving the problem (6.1), the obtained reconstructed surface is shown in Figure 6-6 (solid line) along with the original surface (dotted line). We see an excellent match of the reconstructed surface with the original surface. The maximum error (distance) between the grid points of reconstructed surface and that of the original surface is 0.00459319.

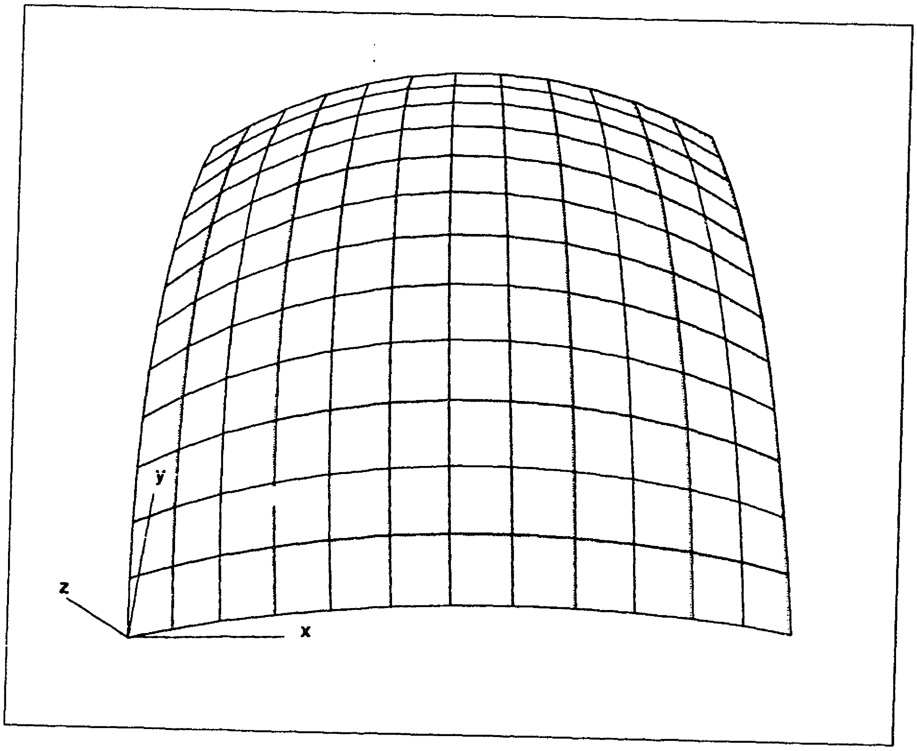

Figure 6-6: The reconstructed (solid line) and the original surfaces (dotted line) in Example 1, when developed along isoparametric lines

Table 6.1 shows the CPU time spent on each optimization for various numbers of grid points, objective functions, etc., where Ng is the number of grid points in both u and v directions; Niter1 is the number of iterations in the first optimization; obj1 is the converged value of the objective function in the first optimization; CPU1 is the CPU time spent on the first optimization; CPU1/Niter1 is the CPU time per iteration in the first optimization; Niter2 is the number of iterations in the second optimization; CPU2 is the CPU time spent on the second optimization; and CPU2/Niter2 is the

CPU time per iteration in the second optimization. For various numbers of grid points, the tolerance for constraints is 10^{-5}, and the tolerance for objective function is 10^{-4}.

Table 6.1: CPU time for each optimization at various number of grid points (Example 1, development along isoparametric lines)

Ng	Niter1	obj1 (10^{-3})	CPU1 (s)	CPU1/Niter1 (s)	Niter2	CPU2 (s)	CPU2/Niter2 (s)
7	2	6.658	0.71	0.355	11	5.86	0.533
9	8	6.492	8.53	1.066	6	12.67	2.112
11	7	6.428	34.72	4.960	5	37.00	7.400
13	18	6.386	177.49	9.861	5	110.66	22.132
15	13	6.359	401.26	30.866	6	333.23	55.538
17	15	6.341	1097.28	73.152	5	658.71	131.742
19	33	6.327	3869.18	117.25	5	1390.80	278.16
21	37	6.317	8168.48	220.77	5	2761.26	555.25

The CPU time per iteration for each optimization is shown in Figures 6-7 and 6-8. We see from Figures 6-7 that the CPU time per iteration in the first optimization agrees well with the theoretical results in Chapter 5; i.e., when $N_g^u = N_g^v = N_g$, the CPU time per iteration in the first optimization is $O(N_g^6)$. However, the performance observed for the second optimization is slightly worse than the theoretical results. Instead of $O(N_g^6)$ CPU time per iteration, we observed $O(N_g^{6.4})$ time per iteration. It is not clear why this happened. One possibility may be because the NAG routine uses iterative methods inside each iteration, so that as N_g increases, the problems are increasingly ill-conditioned, thus requiring more mini-iterations. A thorough track of the running time for different phases of the algorithm may be used to resolve this problem.

Table 6.2 and Figure 6-9 show the maximum error due to the development and reconstruction process for various numbers of grid points. A data fitting process was carried out which fitted the data in Table 6.2 with the function

$$E = e^c(\Delta u)^a$$

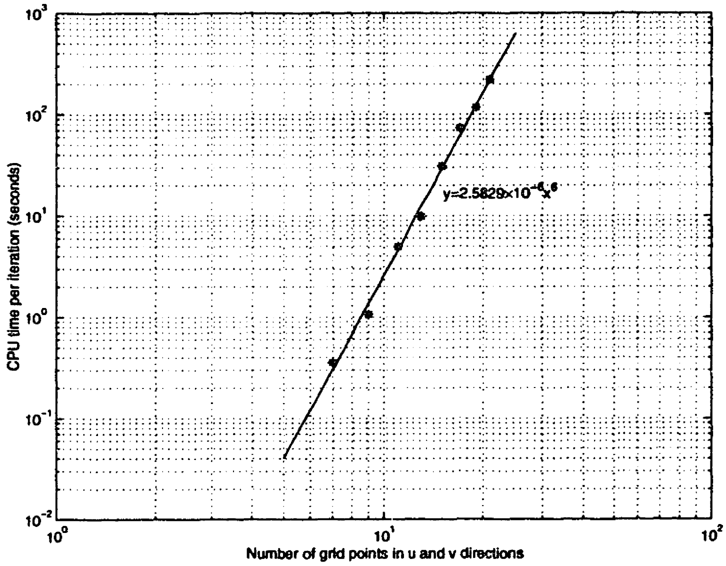

Figure 6-7: CPU time per iteration for 1st optimization for the surface in Example 1, developed along isoparametric lines

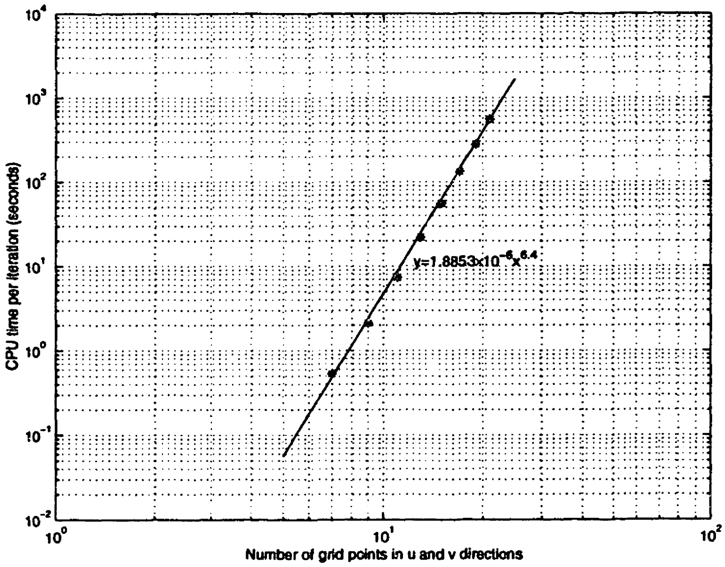

Figure 6-8: CPU time per iteration for 2nd optimization, for the surface in Example 1, developed along isoparametric lines

(where $e = 2.71828...$) or equivalently,

$$lnE = aln(\Delta u) + c \tag{6.2}$$

We solve the coefficients a and c by minimizing the errors in Equation (6.2) for all the data points, which gives out the following two equations:

$$\begin{aligned} \sum_{i=0}^{8}(lnE_i - aln\Delta u_i - c)ln\Delta u_i &= 0 \\ \sum_{i=0}^{8}(lnE_i - aln\Delta u_i - c) &= 0 \end{aligned}$$

The solution is

$$a = 1.9397 \quad c = -0.5641$$

The fitted curve is also plotted in Figure 6-9. Here we see the error function due to the surface development and reconstruction process is of the order $a < 2$. This is partly because the assumption that the angle between isoparametric lines does not change after development introduces extra errors.

Table 6.2: Accuracy of the surface development process (Example 1, development along isoparametric lines)

Grid number	7	9	11	13	15	17	19	21
$\Delta u = \Delta v$	1/6	1/8	1/10	1/12	1/14	1/16	1/18	1/20
Error ($\times 10^{-3}$)	17.763	9.935	6.580	4.549	3.428	2.612	2.113	1.697

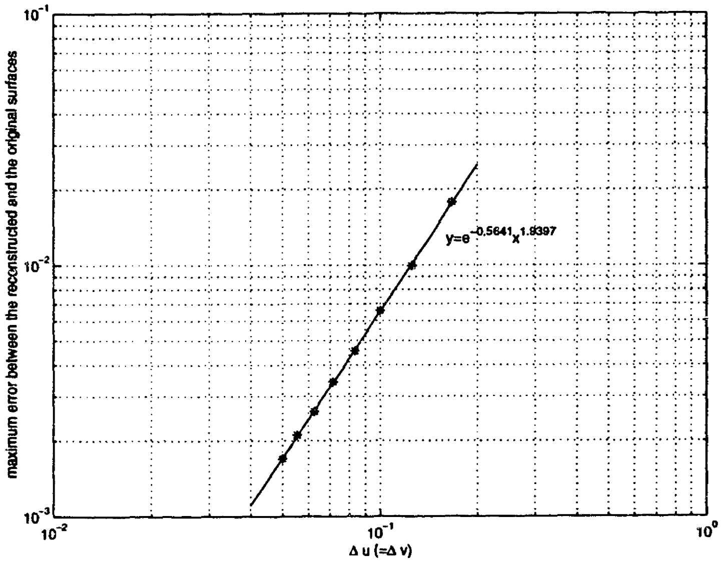

Figure 6-9: Error due to development and reconstruction process for various numbers of grid points for the surface in Example 1, developed along isoparametric lines

6.1.2 Example 2

We use this example to show how the algorithms work and the convergence of the objective function during the process.

The second surface is a bicubic Bézier surface $\mathbf{r}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{r}_{ij}B_{i,3}(u)B_{j,3}(v)$, where all the points on the surface are hyperbolic. The control points of the saddle-shaped surface are given by:

$(0,0,0.25)$	$(0,1/3,0.1)$	$(0,2/3,-0.1)$	$(0,1,-0.25)$
$(1/3,0,0.1)$	$(1/3,1/3,0.05)$	$(1/3,2/3,-0.05)$	$(1/3,1,-0.1)$
$(2/3,0,-0.1)$	$(2/3,1/3,-0.05)$	$(2/3,2/3,0.05)$	$(2/3,1,0.1)$
$(1,0,-0.25)$	$(1,1/3,-0.1)$	$(1,2/3,0.1)$	$(1,1,0.25)$

The surface is shown in Figure 6-10. Again, 13×13 grid points are used in discretization. Figure 6-11 shows the strain distribution after the constrained minimization problem was solved using tolerances of 10^{-5} for the constraints and 10^{-4} for the objective function. The strains are scaled to fit into the figure. The extreme values of the strain field are located at $(u,v) = (0.5, 0.5)$ with $(\varepsilon^u, \varepsilon^v) = (0.0867214, 0.0886954)$. The objective function converges to the value of 4.005672×10^{-3} at the solution, and

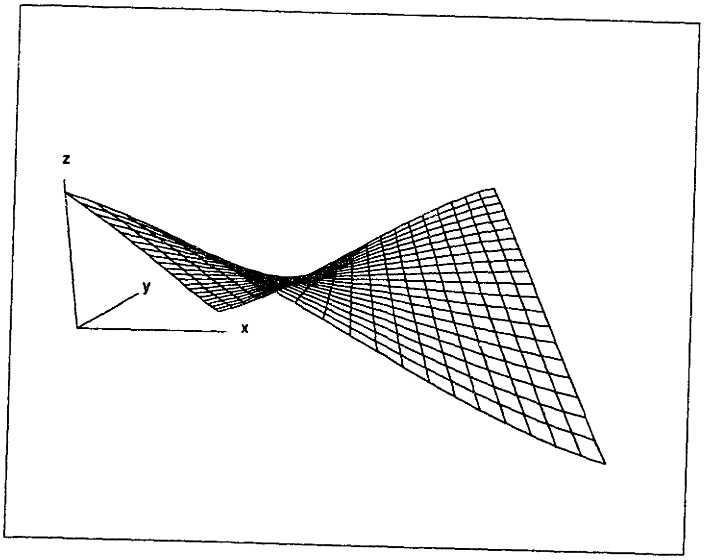

Figure 6-10: The bicubic Bézier surface in Example 2

all the constraints are within the tolerance of 10^{-5}.

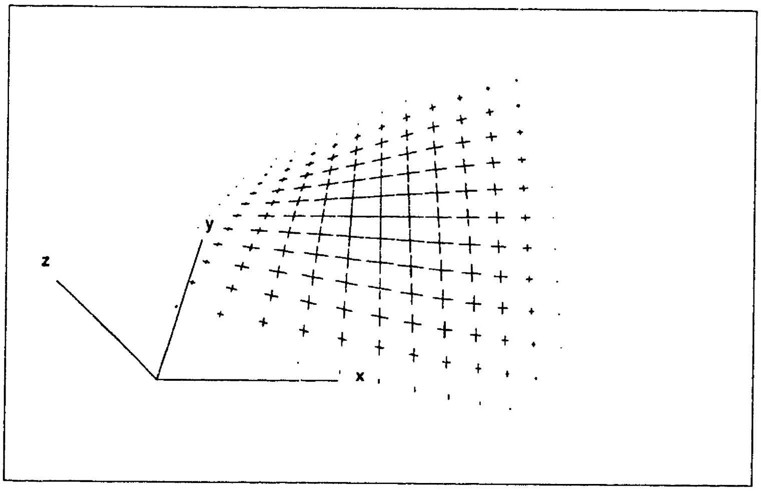

Figure 6-11: The strain distribution of the surface in Example 2, developed along isoparametric lines (length of line segments at grid points show the scaled magnitude of the two strains ε^u and ε^v)

After development, the planar shape is shown in Figure 6-12. The four end points have coordinates of (0,0), (-0.09223, 1.11186), (1.11534,0.09258), (1.02311, 1.20444), respectively. The final value of the formula (3.35) is less than 10^{-5} of the value $\sum_{i=1}^{N_g^u}\sum_{j=1}^{N_g^v}(e_{ij}^2+f_{ij}^2+g_{ij}^2)$, the sum of the squares of the right side of system (3.34) at all grid points. Here we see a planar development similar to that in Example 1.

Table 6.3 and Figure 6-13 show the variation of the objective function in the first optimization with respect to $\Delta u (= \Delta v)$ and number of grid points. As pointed in Chapter 5, a quadratic convergence is observed in the objective function of the first optimization. If we allow extrapolation, we can estimate the objective function approaches 0.003909 as $\Delta u = \Delta v \rightarrow 0$.

Table 6.3: The objective function of the 1st optimization (Example 2, development along isoparametric lines)

Grid number	7	9	11	13	15	17	19	21
$\Delta u = \Delta v$	1/6	1/8	1/10	1/12	1/14	1/16	1/18	1/20
obj1 ($\times 10^{-3}$)	4.2831	4.125	4.038	4.006	3.970	3.964	3.953	3.945

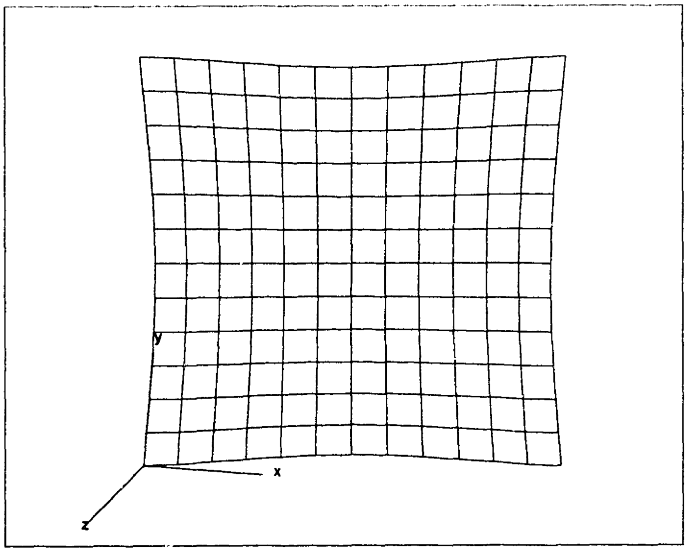

Figure 6-12: The planar development of the surface in Example 2, developed along isoparametric lines

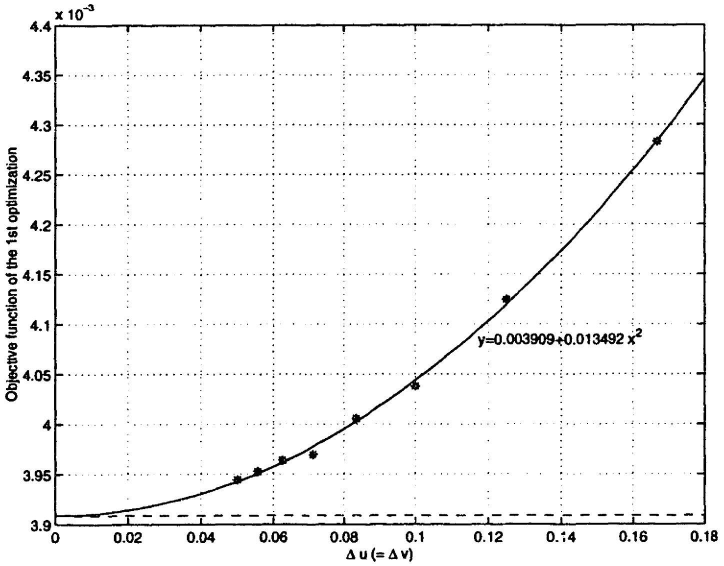

Figure 6-13: Variation of objective function in 1st optimization for the surface in Example 2, developed along isoparametric lines

6.1.3 Example 3

This example shows how the algorithms work for a surface with both positive and negative Gaussian curvature regions.

One eighth of a torus as shown in Figure 6-14 can be expressed as a rational biquadratic B-spline surface [13] [19]

$$\mathbf{r}(u,v) = \frac{\sum_{i=0}^{2}\sum_{j=0}^{4}\mathbf{r}_{ij}w_{ij}N_{i,3}(u)N_{j,3}(v)}{\sum_{i=0}^{2}\sum_{j=0}^{4}w_{ij}N_{i,3}(u)N_{j,3}(v)}$$

with the following control points and weights (x, y, z, w):

$(15, 0, 0, 1)$	$(15, 0, 5, \frac{\sqrt{2}}{2})$	$(10, 0, 5, 1)$	$(5, 0, 5, \frac{\sqrt{2}}{2})$	$(5, 0, 0, 1)$
$(15, 15, 0, \frac{\sqrt{2}}{2})$	$(15, 15, 5, 0.5)$	$(10, 10, 5, \frac{\sqrt{2}}{2})$	$(5, 5, 5, 0.5)$	$(5, 5, 0, \frac{\sqrt{2}}{2})$
$(0, 15, 0, 1)$	$(0, 15, 5, \frac{\sqrt{2}}{2})$	$(0, 10, 5, 1)$	$(0, 5, 5, \frac{\sqrt{2}}{2})$	$(0, 5, 0, 1)$

The knot vector in u direction is (0 0 0 1 1 1) and that in v direction is (0 0 0 0.5 0.5 1 1 1).

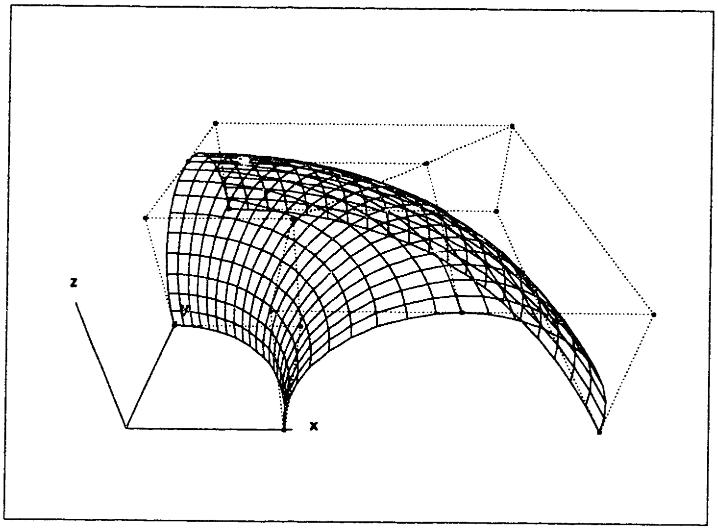

Figure 6-14: Part of a torus surface in example 3

Figure 6-15 shows the strain distribution after the constrained minimization problem was solved using the tolerance of 10^{-8} for constraints and 10^{-4} for objective function. The objective function converges to the value of 1.1641 at the solution, and all the constraints are within the tolerance of 10^{-8}. After the unconstrained nonlinear minimization was solved, the planar development was obtained as shown in Figure 6-16. The final value of the formula (3.35) is less than 10^{-4} of the value $\sum_{i=1}^{N_g^u}\sum_{j=1}^{N_g^v}(e_{ij}^2+f_{ij}^2+g_{ij}^2)$, the sum of the squares of the right side of system (3.34) at all grid points.

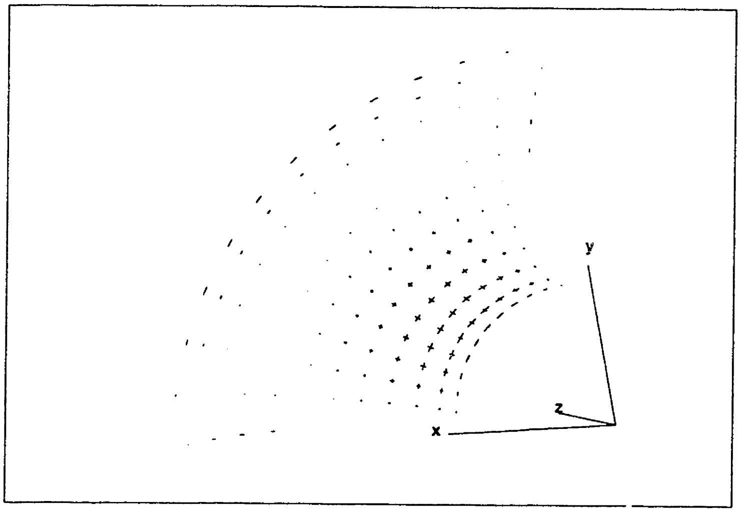

Figure 6-15: Strain distribution of the torus in example 3, developed along isoparametric lines (length of line segments at grid points show the scaled magnitude of the two strains ε^u and ε^v)

We use this example also to illustrate the performance of the algorithms. For both optimization problems, the number of equations and/or the number of constraints increase with the increment of the number of grid points. We solved the surface development problem for various number of grid points and recorded the CPU time spent on each optimization. Table 6.4 shows the CPU time on each optimization at various numbers of grid points. A polynomial estimation of CPU time can be found from Figure 6-17 and Figure 6-18. Again, the CPU time per iteration for the first optimization is $O(N_g^6)$ while that for the second optimization is slightly worse than that.

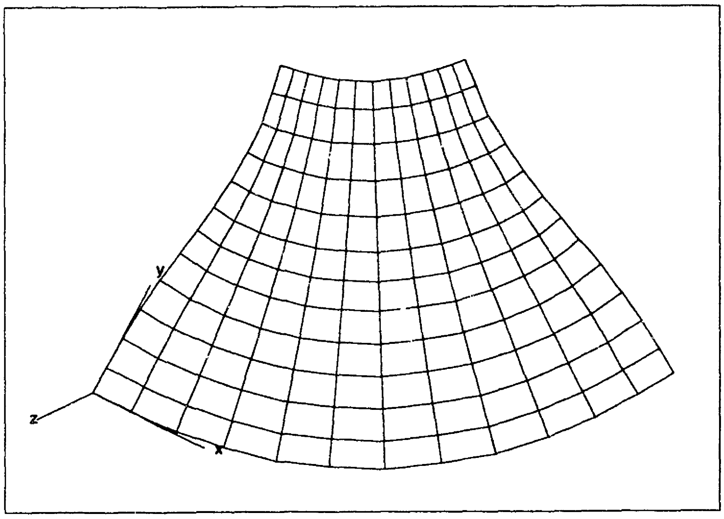

Figure 6-16: The planar development of the torus in example 3

Table 6.4: CPU time for each optimization at various numbers of grid points (Example 3, development along isoparametric lines)

Ng	Niter1	obj1	CPU1 (s)	CPU1/Niter1 (s)	Niter2	CPU2 (s)	CPU2/Niter2 (s)
7	15	1.0966	1.72	0.1146	10	5.01	0.501
9	27	1.1495	12.81	0.4744	11	24.68	2.244
11	17	1.1796	32.41	1.9065	7	54.13	7.733
13	10	1.1641	70.39	7.0390	9	206.63	22.959
15	8	1.1894	170.09	21.261	7	424.76	60.680
17	5	1.1245	305.58	61.116	8	1094.76	136.845
19	5	1.0491	676.56	135.31	7	1964.05	280.579
21	5	0.9829	1232.20	246.44	8	4496.91	562.114
25	53	0.8568	18892.05	356.45	11	19358.88	1759.90
31	83	0.7464	100383	1209.43	9	59900.63	6655.63

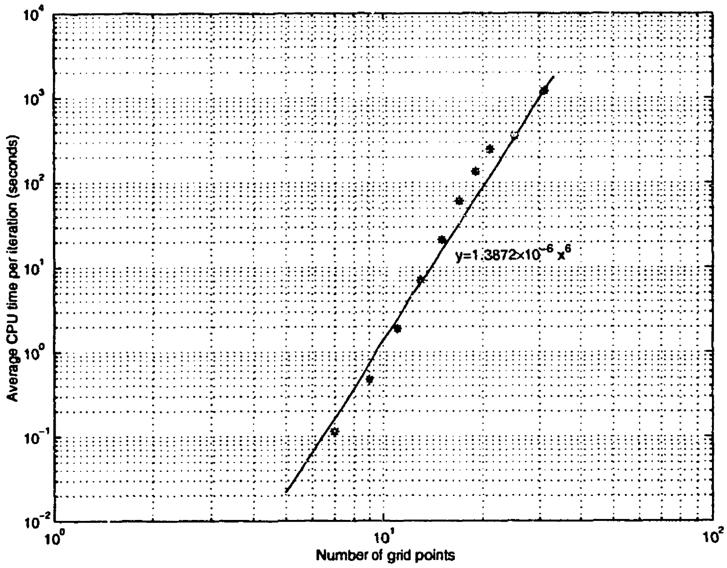

Figure 6-17: Average CPU time per iteration for 1st optimization at various numbers of grid points for the surface in Example 3, developed along isoparametric lines

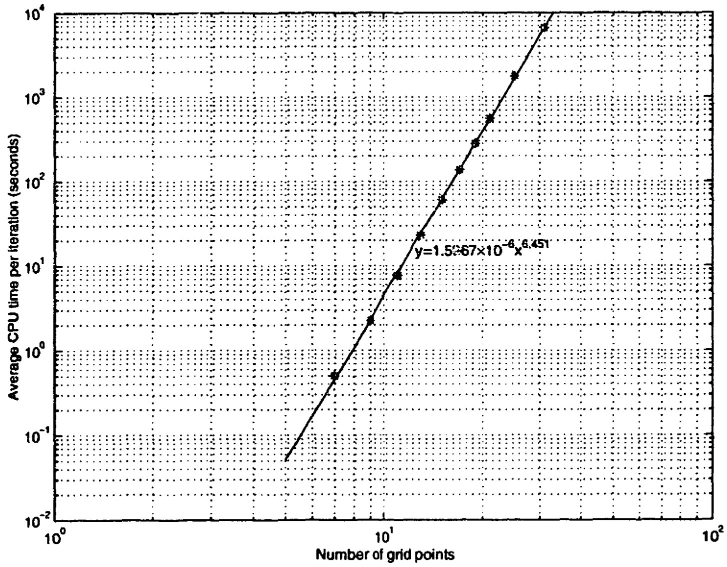

Figure 6-18: Average CPU time per iteration for 2nd optimization at various numbers of grid points for the surface in Example 3, developed along isoparametric lines

6.1.4 Example 4

This example shows the performance of the algorithms on a general B-spline surface. A wave-like bicubic integral B-spline surface

$$\mathbf{r}(u,v)=\sum_{i=0}^{4}\sum_{j=0}^{4}\mathbf{r}_{ij}N_{i,4}(u)N_{j,4}(v)$$

on a uniform knot vector with the following control points (x,y,z)

$(0,0,0)$	$(0,0.25,0)$	$(0,0.5,0)$	$(0,0.75,0)$	$(0,1,0)$
$(0.25,0,0)$	$(0.25,0.25,0.2)$	$(0.25,0.5,0)$	$(0.25,0.75,-0.2)$	$(0.25,1,0)$
$(0.5,0,0)$	$(0.5,0.25,0)$	$(0.5,0.5,0)$	$(0.5,0.75,0)$	$(0.5,1,0)$
$(0.75,0,0)$	$(0.75,0.25,-0.2)$	$(0.75,0.5,0)$	$(0.75,0.75,0.2)$	$(0.75,1,0)$
$(1,0,0)$	$(1,0.25,0)$	$(1,0.5,0)$	$(1,0.75,0)$	$(1,1,0)$

is shown in Figure 6-19. 17 × 17 grid points are used in discretization. Figure 6-20 shows the strain distribution after the constrained minimization problem was solved using tolerances of 10^{-7} for the constraints and 10^{-4} for the objective function. The strains are scaled to fit into the figure. The objective function converges to the

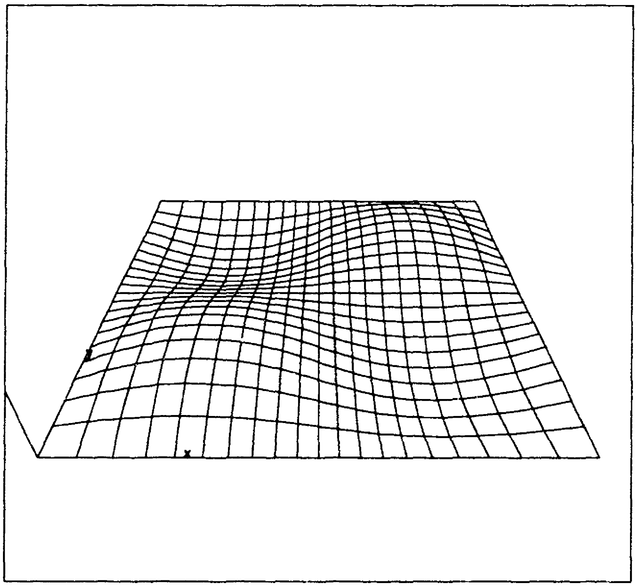

Figure 6-19: A wave-like B-spline surface in example 4

value of 0.000984842 at the solution, and all the constraints are within the tolerance of 10^{-7}. After development, the planar shape is shown in Figure 6-21. The four end points have coordinates of (0,0), (0.00315,1.03617), (1.03618,-0.00312), (1.03933, 1.03306), respectively. The strains at the center of the surface $(u, v) = (0.5, 0.5)$ are $(\varepsilon^u, \varepsilon^v) = (0.0555, 0.0550)$. The final value of the formula (3.35) is about 10^{-4} of the value $\sum_{i=1}^{N_g^u} \sum_{j=1}^{N_g^v} (e_{ij}^2 + f_{ij}^2 + g_{ij}^2)$, the sum of the squares of the right side of system (3.34) at all grid points.

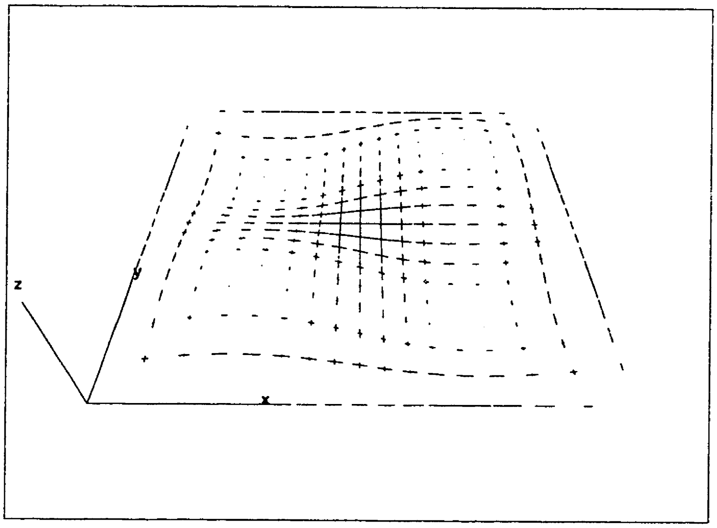

Figure 6-20: The strain distribution of the surface in Example 4, developed along isoparametric lines (length of line segments at grid points show the scaled magnitude of the two strains ε^u and ε^v)

Table 6.5 shows the number of iterations and CPU time spent on each optimization, as well as the objective functions. Figures 6-22 and 6-23 show the CPU time per iteration for each optimization problem for various numbers of grid points.

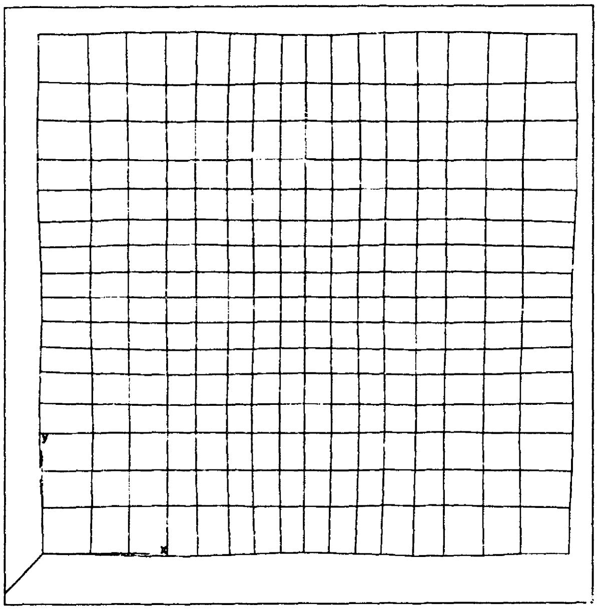

Figure 6-21: The planar development of the surface in Example 4, developed along isoparametric lines

Table 6.5: CPU time for each optimization at various numbers of grid points (Example 4, development along isoparametric lines)

Ng	Niter1	obj1 (10^{-4})	CPU1 (s)	CPU1/Niter1 (s)	Niter2	CPU2 (s)	CPU2/Niter2 (s)
7	10	9.362	8.96	0.898	9	4.35	0.483
9	13	8.453	38.25	2.942	12	25.01	2.084
11	13	9.434	110.05	8.465	6	43.54	7.257
13	20	8.873	348.69	17.435	6	129.20	21.533
15	32	10.192	1269.36	39.668	6	329.66	54.943
17	23	9.848	1788.00	77.739	5	659.31	131.862
19	8	10.457	1877.89	234.736	5	1360.02	272.00
21	17	10.441	5157.98	303.411	5	2681.23	536.246
25	22	10.729	19387.45	881.247	5	8294.06	1658.812

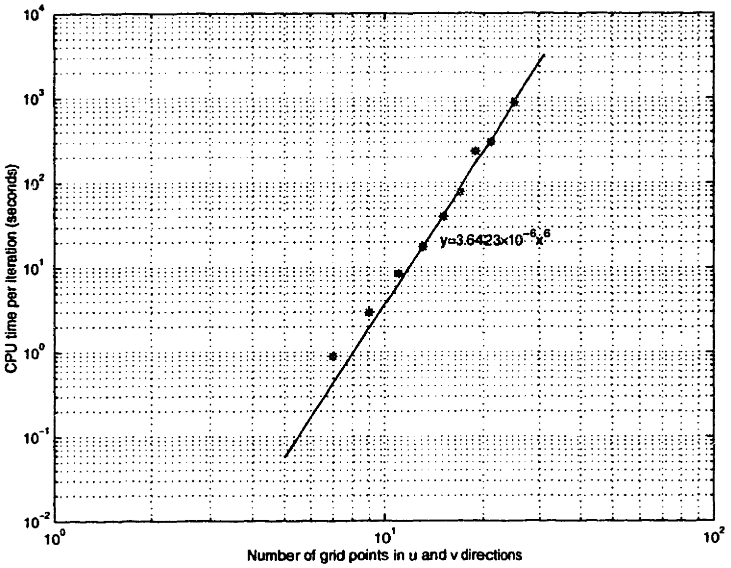

Figure 6-22: CPU time per iteration for the first optimization for the surface in Example 4, developed along isoparametric lines

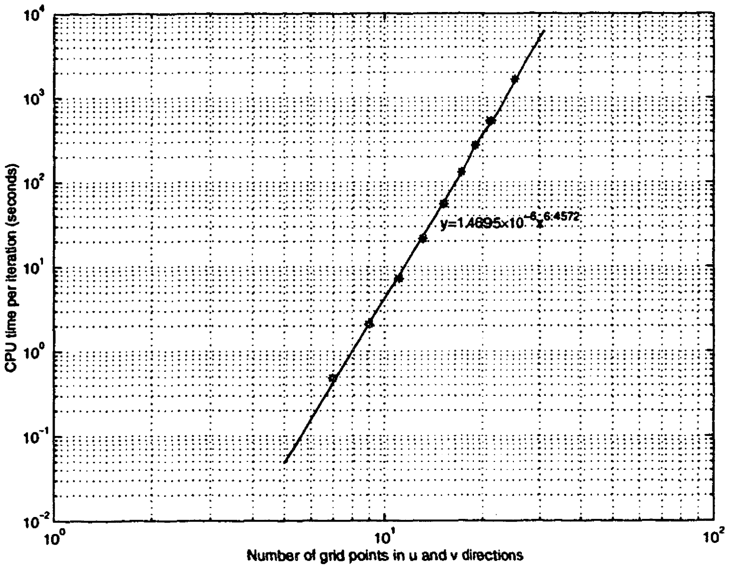

Figure 6-23: CPU time per iteration for the second optimization for the surface in Example 4, developed along isoparametric lines

6.2 Results on surface development along principal curvature directions

6.2.1 Example 1

In this example, we use the surface whose points are all elliptic as in example 1 of Section 6.1, but allow it to stretch along principal curvature directions. The surface along with its control polygon is shown in Figure 6-1. The constrained minimization problem (4.11-4.12) is discretized at 13 × 13 grid points which are equally distributed in u, v domain. Figure 6-24 shows the strain distribution after the constrained minimization problem was solved using tolerances 10^{-5} for the constraints and 10^{-4} for the objective function. The strains are scaled to fit into the figure. The extreme values of the strain field are located at $(u, v) = (0, 0.5)$ or $(u, v) = (1.0, 0.5)$ with $(\varepsilon^s, \varepsilon^t) = (0.1808969, 0.006270948)$, and at $(u, v) = (0.5, 0)$ or $(u, v) = (0.5, 1.0)$ with $(\varepsilon^s, \varepsilon^t) = (0.001567142, 0.1790917)$. The objective function converges to 6.78804×10^{-3} at the solution, and all the constraints are within the tolerance of 1.0×10^{-5}.

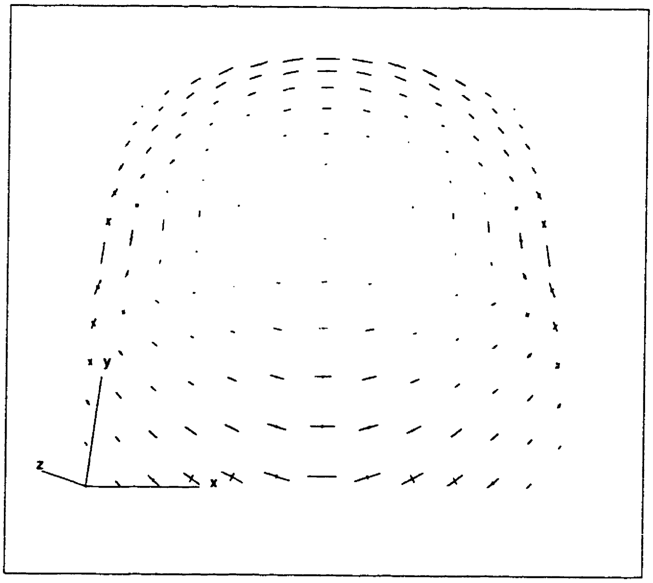

Figure 6-24: The strain distribution of the surface in Example 1, developed along the principal curvature directions (length of line segments at grid points show the scaled magnitude of the two strains ε^u and ε^v)

After development, the planar shape is shown in Figure 6-25. The four end points have coordinates of (0,0), (-0.11802,1.09373), (1.17949,0.12786), (1.06147,1.22159) respectively. The final value of the formula (3.35) is less than 2×10^{-4} of the value $\sum_{i=1}^{N_g^u}\sum_{j=1}^{N_g^v}(e_{ij}^2 + f_{ij}^2 + g_{ij}^2)$, the sum of the squares of the right side of system (3.34) at all grid points.

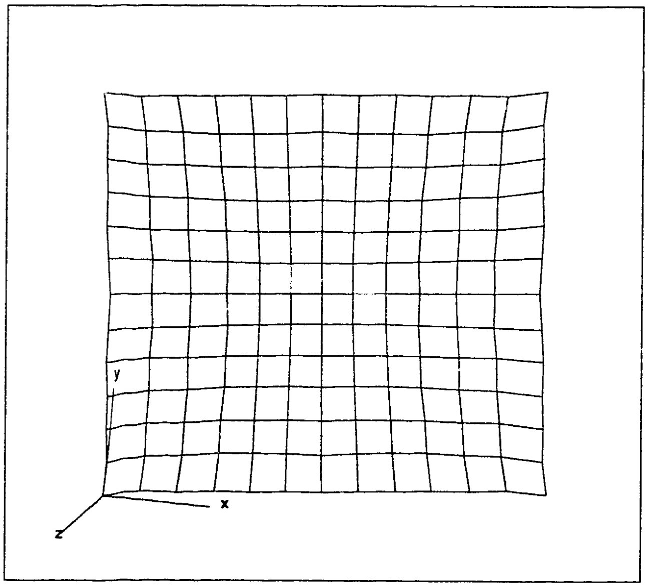

Figure 6-25: The planar development of the surface in Example 1, developed along the principal curvature directions

Figure 6-26 and Figure 6-27 show the ideal strain gradient $\frac{\partial[ln(1+\varepsilon^s)]}{\partial d}$ and $\frac{\partial[ln(1+\varepsilon^t)]}{\partial d}$ evaluated at grid points.

After solving ε^{max} and ε^{min} by the first nonlinear constrained optimization, and subsequently e, f, g in the second optimization, we can then compute E, F, G by Equations (4.25-4.27). Under the condition of ideal strain gradient and E, F, G evaluated by Equations (4.25-4.27), the surface reconstruction process is equivalent to solving Equation (6.1). After getting rid of the rigid body motion, the obtained reconstructed surface is shown in Figure 6-28 (solid line) along with the original surface (dotted line). We see an excellent match of the reconstructed surface with the original surface. The maximum error (distance) between the grid points of reconstructed surface and that of the original surface is 0.00509951.

Table 6.6 shows the CPU time spent on each optimization for various number of

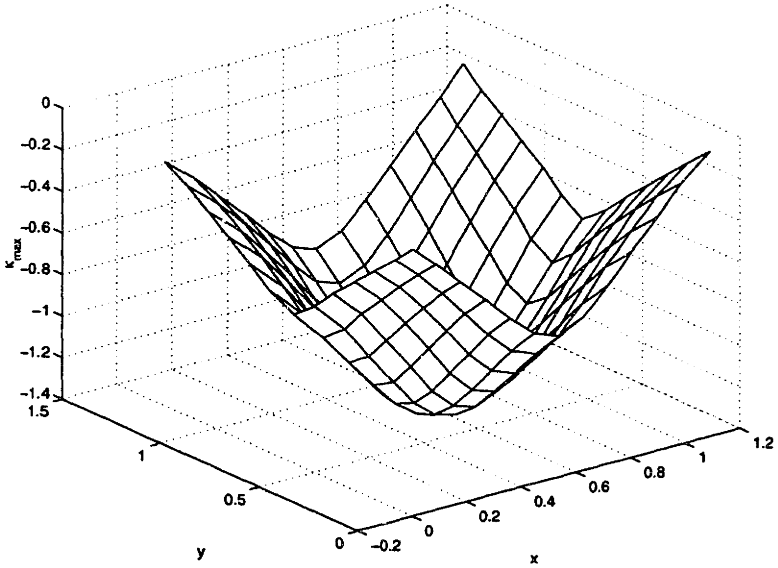

Figure 6-26: Logarithmic strain gradient along maximum curvature direction for the surface in Example 1, developed along the principal curvature directions

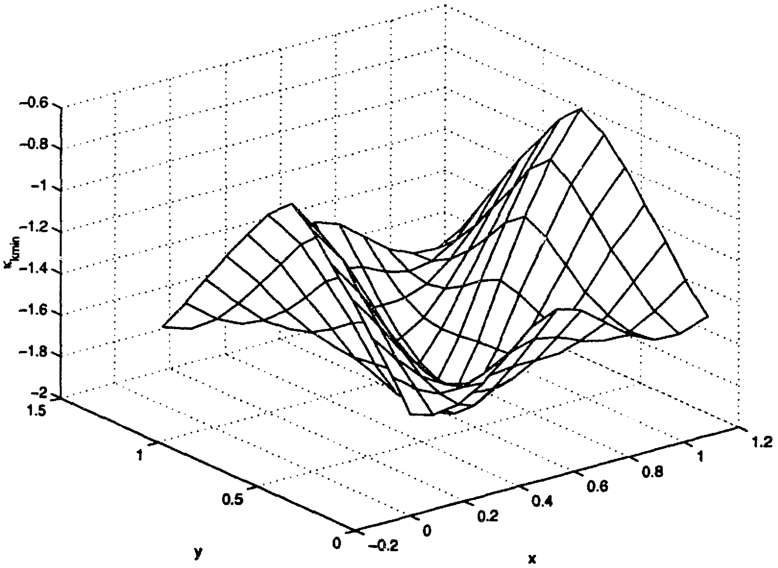

Figure 6-27: Logarithmic strain gradient along minimum curvature direction for the surface in Example 1, developed along the principal curvature directions

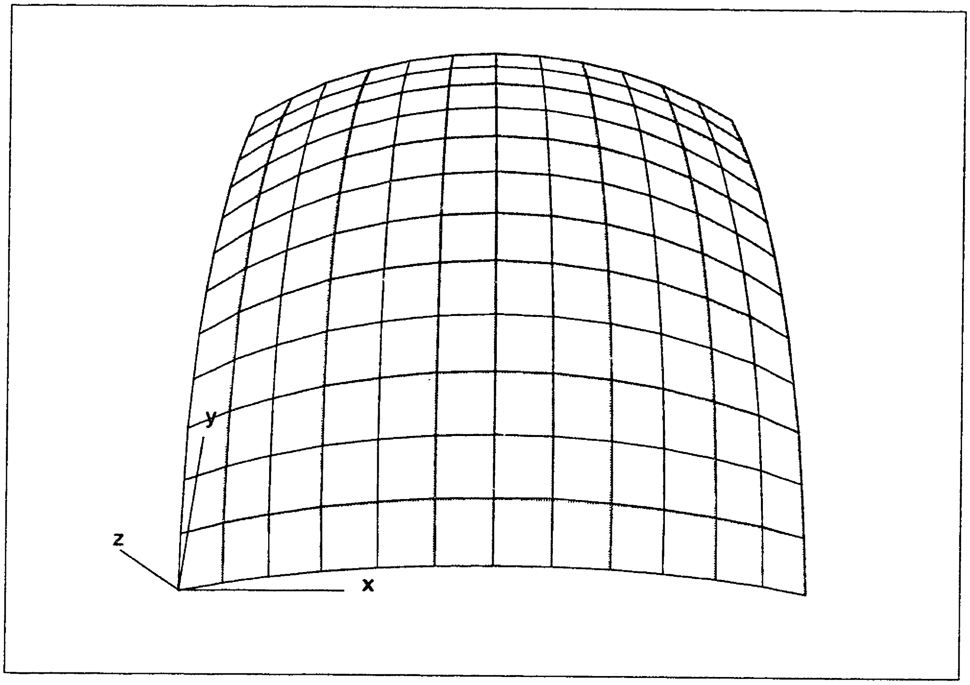

Figure 6-28: The reconstructed (solid line) and the original surfaces (dotted line) for the surface in Example 1, developed along the principal curvature directions

grid points, objective functions, etc.

Table 6.6: CPU time for each optimization at various numbers of grid points (Example 1, development along principal curvature directions)

Ng	Niterl	objl (10^{-3})	CPUl(s)	CPUl/Niterl(s)	Niter2	CPU2(s)	CPU2/Niter2(s)
7	6	6.389	1.45	0.242	11	5.66	0.515
9	6	6.610	6.52	1.087	6	11.84	1.973
11	9	6.703	32.16	3.573	6	41.75	6.958
13	13	6.788	124.99	9.615	5	106.97	21.394
15	16	6.826	413.91	25.870	5	298.96	59.792
17	15	6.842	942.79	62.853	5	682.85	136.57
19	18	6.846	2341.75	130.10	5	1419.11	283.82
21	28	6.850	5770.87	206.10	5	2761.5	552.30

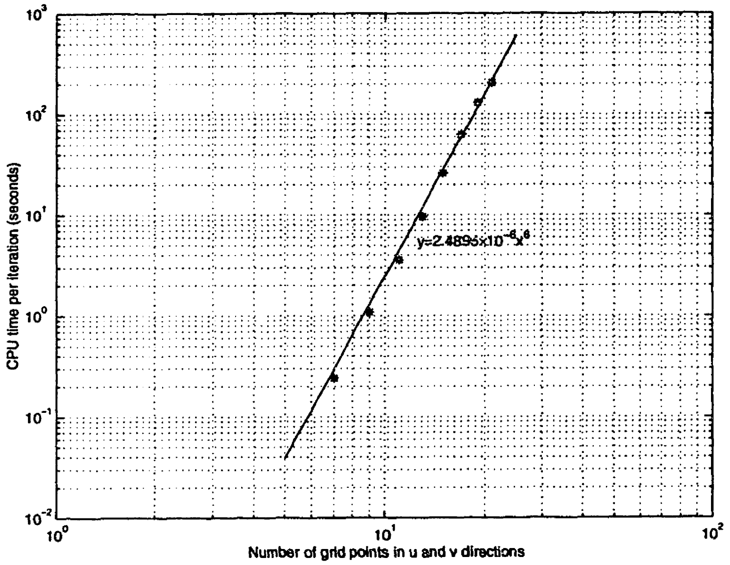

Figure 6-29: CPU time per iteration for 1st optimization for the surface in Example 1, developed along the principal curvature directions

The CPU time per iteration for each optimization is shown in Figures 6-29 and 6-30. The straight line in Figure 6-30 was obtained by data fitting using Equation (6.2). The solution is

$$a = 6.4623 \quad c = -13.4212 \tag{6.3}$$

We see from Figures 6-29 that the CPU time per iteration in the first optimization agrees well with the theoretical results in Chapter 5; i.e., When $N_g^u = N_g^v = N_g$, the

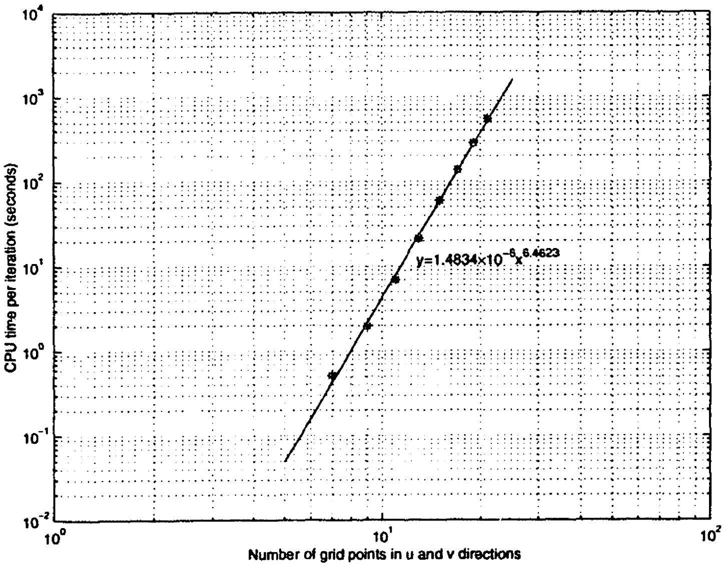

Figure 6-30: CPU time per iteration for 2nd optimization for the surface in Example 1, developed along the principal curvature directions

CPU time per iteration in the first optimization is $O(N_g^6)$. However, the performance observed for the second optimization is slightly worse than the theoretical results. Instead of $O(N_g^6)$ CPU time per iteration, we observed $O(N_g^{6.46})$ time per iteration.

Table 6.7 and Figure 6-31 show the maximum error due to the development and reconstruction process for various number of grid points. Again, a data fitting process was carried out which fitted the data in Table 6.7 with the function

$$E = e^c(\Delta u)^a$$

Here we also see the error function due to the surface development and reconstruction process is of the order $a < 2$. This is partly because the assumption that the angle between curvature directions does not change after development introduces extra errors.

Table 6.7: Accuracy of the surface development process (Example 1, development along principal curvature directions)

Grid number	7	9	11	13	15	17	19	21
$\Delta u = \Delta v$	1/6	1/8	1/10	1/12	1/14	1/16	1/18	1/20
Error ($\times 10^{-3}$)	18.1004	9.9129	7.1817	5.0995	3.9349	3.0282	2.4936	1.9664

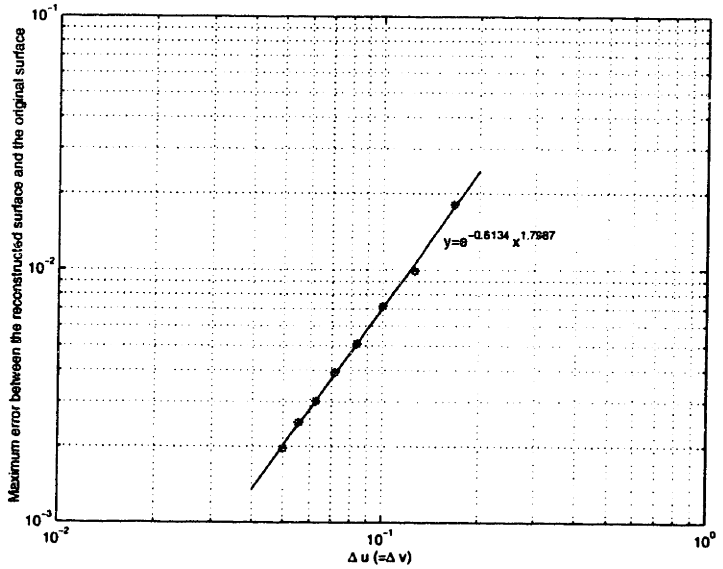

Figure 6-31: Error due to development and reconstructed process for various number of grid points for the surface in Example 1, developed along the principal curvature directions

6.2.2 Example 2

In this example, we use the same surface whose points are all hyperbolic as in example 2 of Section 6.1, but allow it to stretch along principal curvature directions. The surface is shown in Figure 6-10. The constrained minimization problem (4.11-4.12) is discretized at 13 × 13 grid points which are equally distributed in u, v domain. Figure 6-32 shows the strain distribution after the constrained minimization problem was solved using tolerances 10^{-5} for the constraints and 10^{-4} for the objective function. The strains are scaled to fit into the figure. As a comparison with the results in Section 6.1, the strains at $(u, v) = (0.5, 0.5)$ are $(\varepsilon^s, \varepsilon^t) = (0.05627, 0.05581)$. The objective function is 2.791×10^{-3} at the solution, and all the constraints are within the tolerance of 1.0×10^{-5}.

After development, the planar shape is shown in Figure 6-33. The four end points have coordinates of (0,0), (-0.09909,1.11083), (1.11505,0.09953), (1.01596,1.21035) respectively. The final value of the formula (3.35) is less than 3×10^{-5} of of the value $\sum_{i=1}^{N_g^u} \sum_{j=1}^{N_g^v} (e_{ij}^2 + f_{ij}^2 + g_{ij}^2)$, the sum of the squares of the right side of system (3.34) at all grid points.

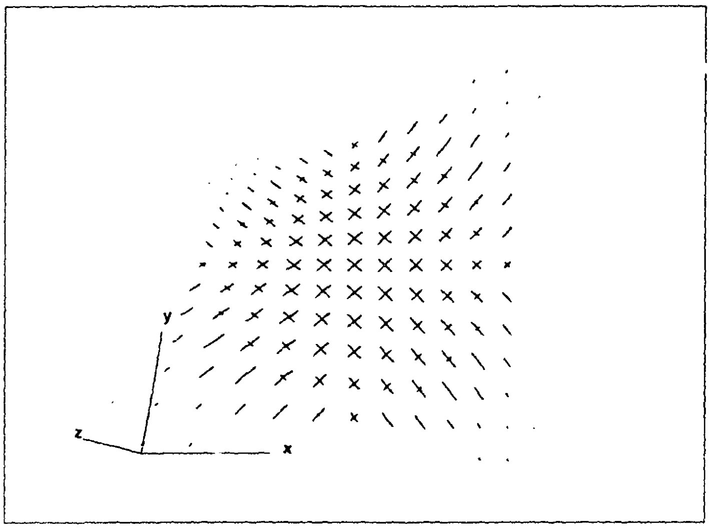

Figure 6-32: The strain distribution of the surface in Example 2, developed along the principal curvature directions (length of line segments at grid points show the scaled magnitude of the two strains ε^s and ε^t)

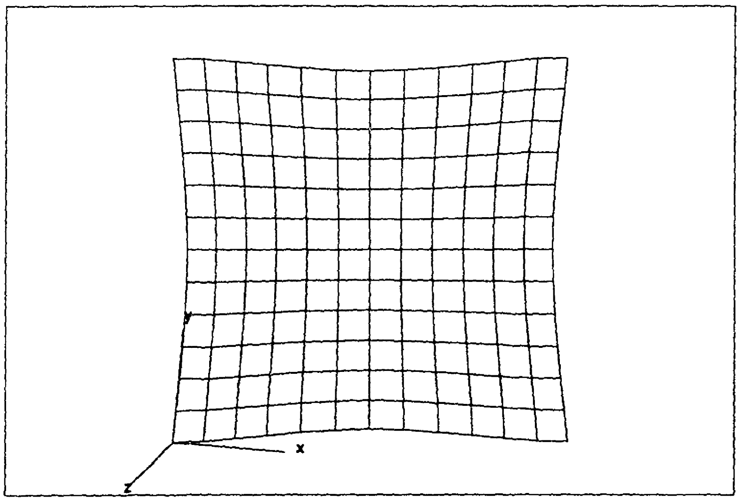

Figure 6-33: The planar development of the surface in Example 2, developed along the principal curvature directions

Figure 6-34 and Figure 6-35 show the ideal strain gradient $\frac{\partial[ln(1+\varepsilon^s)]}{\partial d}$ and $\frac{\partial[ln(1+\varepsilon^t)]}{\partial d}$ evaluated at grid points.

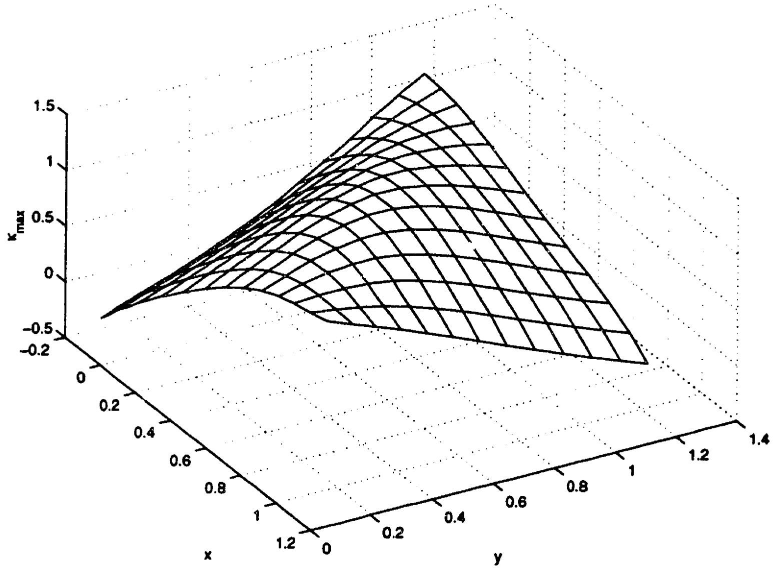

Figure 6-34: Logarithmic strain gradient along maximum curvature direction for the surface in Example 2, developed along the principal curvature directions

Table 6.8 and Figure 6-36 shows the variation of the objective function in the first optimization with respect to $\Delta u(=\Delta v)$ and number of grid points. As pointed in Chapter 5, a quadratic convergence is observed of the objective function of the first optimization. If we allow extrapolation, we can estimate the objective function approaches 0.002689 as $\Delta u = \Delta v \to 0$.

Table 6.8: The objective function of the 1st optimization (Example 2, development along principal curvature directions)

Grid number	7	9	11	13	15	17	19	21
$\Delta u = \Delta v$	1/6	1/8	1/10	1/12	1/14	1/16	1/18	1/20
objl ($\times 10^{-3}$)	3.1611	2.93382	2.8422	2.7909	2.7661	2.7486	2.7388	2.7331

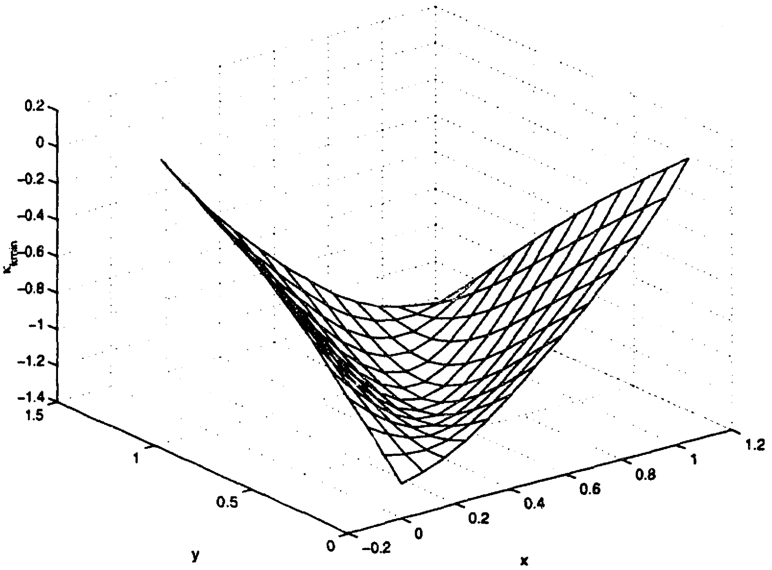

Figure 6-35: Logarithmic strain gradient along minimum curvature direction for the surface in Example 2, developed along the principal curvature directions

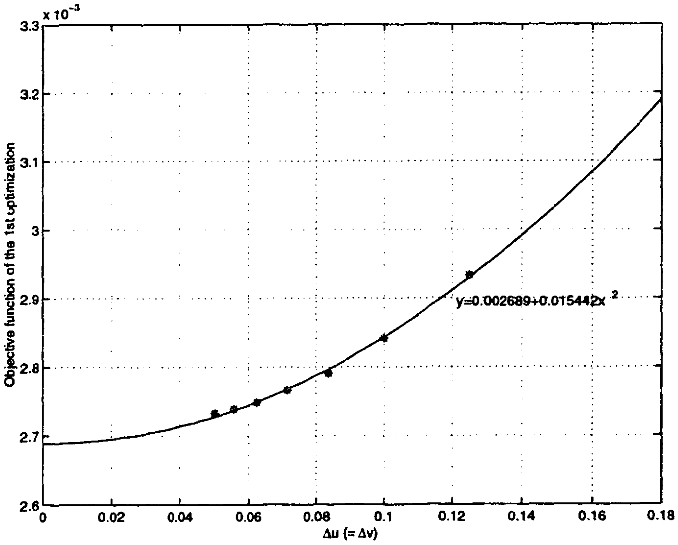

Figure 6-36: Variation of objective function in 1st optimization for the surface in Example 2, developed along the principal curvature directions

6.2.3 Example 4

In this example, we use the same wave-like B-spline surface as in example 4 of Section 6.1, but develop it by using strains along principal curvature directions. The surface is shown in Figure 6-19. The constrained minimization problem (4.11-4.12) is discretized at 17 × 17 grid points which are equally distributed in u, v domain. Figure 6-37 shows the strain distribution after the constrained minimization problem was solved using the tolerance of 10^{-8} for the constraints and 10^{-4} for the objective function. The strains are scaled to fit into the figure. As a comparison with the results in Section 6.1, the strains at $(u, v) = (0.5, 0.5)$ are $(\varepsilon^s, \varepsilon^t) = (0.08258, 0.08267)$. The objective function is 1.6018×10^{-3} at the solution, and all the constraints are within the tolerance of 1.0×10^{-8}. After development, the planar shape is shown in Figure 6-38. The four end points have coordinates of $(0, 0)$, $(0.01073, 1.03467)$, $(1.03467, -0.01074)$, $(1.04540, 1.02393)$, respectively.

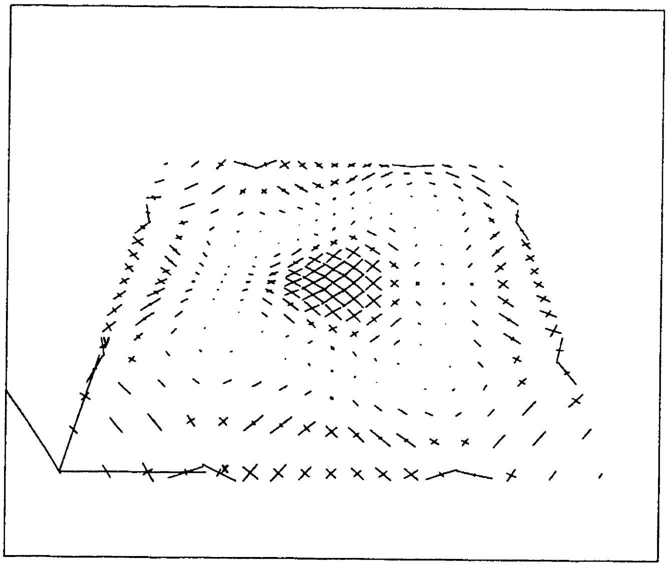

Figure 6-37: The strain distribution of the surface in Example 4, developed along the principal curvature directions (length of line segments at grid points show the scaled magnitude of the two strains ε^u and ε^v)

Table 6.9 shows the number of iterations and CPU time spent on each optimization, as well as the objective functions. Figures 6-39 and 6-40 show the CPU time per iteration for each optimization problem for various number of grid points.

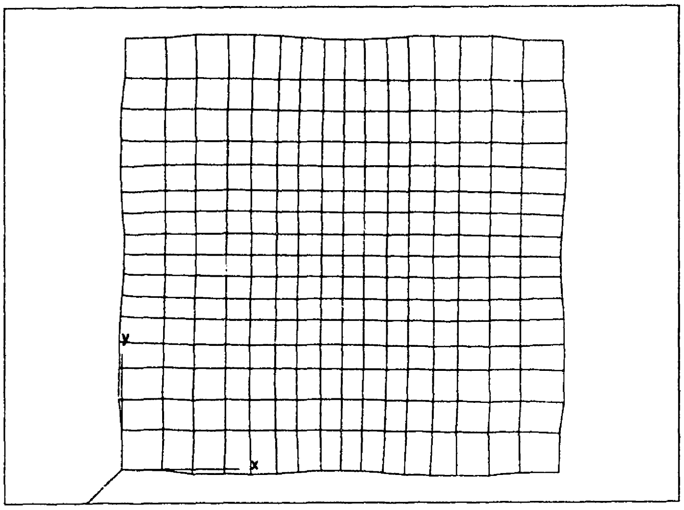

Figure 6-38: The planar development of the surface in Example 4, developed along the principal curvature directions

Table 6.9: CPU time for each optimization at various numbers of grid points (Example 4, development along principal curvature directions)

Ng	Niter1	obj1 (10^{-3})	CPU1(s)	CPU1/Niter1(s)	Niter2	CPU2(s)	CPU2/Niter2(s)
7	11	1.1857	1.70	0.1545	15	7.40	0.493
9	11	1.2901	9.66	0.878	8	16.81	2.101
11	27	1.2652	69.06	2.558	8	60.60	7.575
13	20	1.520	174.96	8.748	7	157.61	22.516
15	48	1.371	892.42	18.592	7	406.91	58.13
17	80	1.602	3099.44	38.743	8	1050.71	131.339
19	35	1.537	3423.56	97.816	6	1584.16	264.027
21	49	1.576	7742.83	158.017	5	2697.65	539.53
25	63	1.550	29482.15	467.97	5	8364.47	1672.894

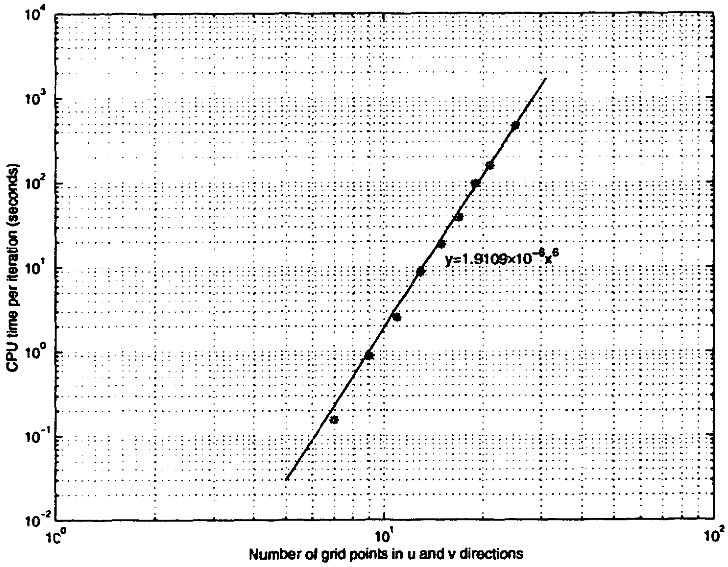

Figure 6-39: CPU time per iteration for the first optimization for the surface in Example 4, developed along the principal curvature directions

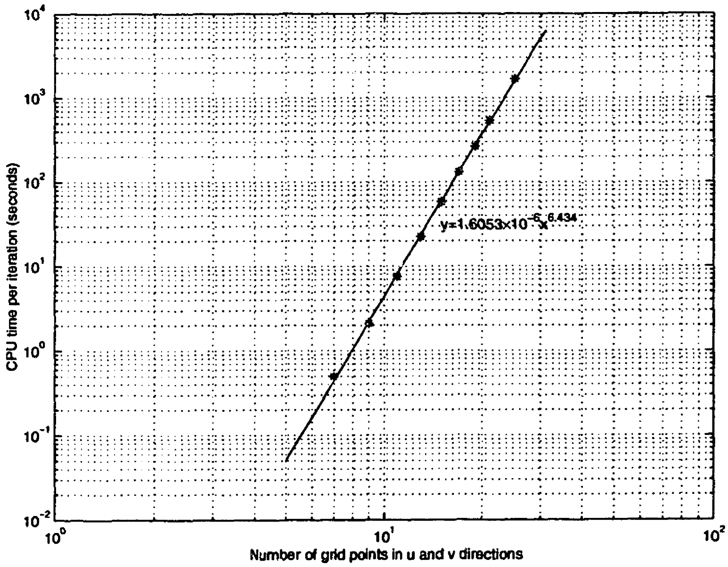

Figure 6-40: CPU time per iteration for the second optimization for the surface in Example 4, developed along the principal curvature directions

6.3 Discussion

The examples in this chapter show that the algorithms for surface development along isoparametric lines and principal curvature directions work well geometrically. The strains obtained and the CPU time spent on both methods are at the same magnitude. The 2D developed shapes are similar. Physically, however, development along principal curvature directions is more realizable. This can be seen from the equations that the strain gradients must satisfy. In chapter 3, the ideal strains ε^u and ε^v must satisfy Equations (3.26), (3.27) and (3.32). Since the left hand side of Equation (3.32) equals the sum of the left hand sides of Equations (3.26) and (3.27), we have

$$\frac{L}{E} + \frac{N}{G} = \frac{2M}{F} - cot\theta \frac{\partial(\Delta\theta)}{\partial d}\bigg|_{d=0} \tag{6.4}$$

This gradient of the angle change $\frac{\partial(\Delta\theta)}{\partial d}\big|_{d=0}$ is hard to control during metal forming process, since $\frac{\partial(\Delta\theta)}{\partial d}\big|_{d=0}$ is not directly related to the strain gradients. $\Delta\theta$ is due to shear strain, which is of the second order compared with the normal strains, and is not simply related to the temperature distribution during metal forming process by line heating. The gradients of the ideal principal strains in Equations (4.19, 4.20), however, are easier to control by controlling the temperature gradient throughout the plate thickness.

Chapter 7

Concluding remarks

An algorithm based on nonlinear optimization for development of a doubly curved surface has been presented in this thesis. Examples of development of a surface with all elliptical points, a surface with all hyperbolic points, a torus and a general B-spline surface show the effectiveness of the algorithm. Compared with the available algorithms for surface development, the algorithm proposed here always finds a solution that only stretching is required from curved surface to its planar development, or only shrinkage is required from planar development to the curved surface. This corresponds to forming of the surface by using (laser or torch) line heating. For other manufacturing process, the formulation of the minimization problem only needs to be slightly modified to take account of dilations from a planar shape to a curved surface.

Comparison of the two surface development methods along isoparametric lines and principal curvature directions shows no significant difference between their performance, although development along principal curvature directions gives out strain gradients which are more realizable.

The examples show that the algorithm is quite time-consuming when the number of grid points is large. An improvement may be possible if we explore the banded properties of the Jacobian matrix after discretization of the constraints in the first optimization and the least squares functions of the second optimization. Because of the finite difference method in approximating all derivatives, after discretization, the constraint in the first optimization or the least squares function in the second optimization only involve the variables at the neighboring points. We may also subdivide a surface into a number of subpatches and optimally develop each of them sequentially. This way, the total CPU time would be cut significantly. Of course the continuity between neighboring subpatches needs to be enforced, and the final solution may not be a global optimal solution. These are tasks for future research.

Bibliography

[1] P. Azariadis and N. Aspragathos. Design of plane developments of doubly curved surfaces. *Computer Aided Design*, 29(10):675–685, 1997.

[2] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, USA, 1995.

[3] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, Massachusetts, USA, 1997.

[4] W. Cho, N. M. Patrikalakis, and J. Peraire. Approximate development of trimmed patches for surface tessellation. *Computer Aided Design*, 30(14):1077–1087, 1998.

[5] P. G. Ciarlet. *Introduction to Numerical Linear Algebra and Optimization.* Cambridge University Press, Cambridge, GB, 1989.

[6] G. Dahlquist and A. Björck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

[7] P. M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[8] S. I. Gass. *Linear Programming: Methods and Applications.* McGraw-Hill, New York, 1985.

[9] I. M. Gelfand and S. V. Fomin. *Calculus of Variations.* Prentice-Hall, Englewood Cliffs, NJ, 1963.

[10] P. E. Gill and W. Murray. Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal of Numerical Analysis*, 15:977–992, 1978.

[11] P. E. Gill, W. Murray, and A. Wright. *Practical Optimization.* Academic Press, New York, 1981.

[12] B. K. Hinds, J. McCartney, and G. Woods. Pattern development for 3D surfaces. *Computer Aided Design*, 23(8):583–592, 1991.

[13] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* A. K. Peters, Wellesley, MA, 1993. Translated by L. L. Schumaker.

[14] E. Kreyszig. *Differential Geometry.* University of Toronto Press, Toronto, 1959.

[15] J. S. Letcher, Jr. Lofting and fabrication of compound-curved plates. *Journal of Ship Research*, 37(2):166–175, 1993.

[16] M. M. Lipschutz. *Theory and Problems of Differential Geometry.* Schaum's Outline Series: McGraw-Hill, 1969.

[17] J. R. Manning. Computerized pattern cutting: Methods based on an isometric tree. *Computer Aided Design*, 12(1):43–47, 1980.

[18] Numerical Algorithms Group, Oxford, England. *NAG Fortran Library Manual, Volumes 1-8*, Mark 14 edition, 1990.

[19] L. Piegl and W. Tiller. *The NURBS Book.* Springer, New York, 1995.

[20] D. J. Struik. *Lectures on Classical Differential Geometry.* Addison-Wesley, Cambridge Mass., 1950.

[21] L. N. Trefethen and D. III Bau. *Numerical Linear Algebra.* SIAM, Philadelphia, PA, 1997.

[22] K. Ueda, H. Murakawa, A. M. Rashwan, Y. Okumoto, and R. Kamichika. Development of computer-aided process planning system for plate bending by line

heating (report 1) – relation between final form of plate and inherent strain. *Journal of Ship Production*, 10(1):59–67, 1994.

[23] T. J. Willmore. *An Introduction to Differential Geometry.* Clarendon Press, Oxford, 1959.

THESIS PROCESSING SLIP

FIXED FIELD: ill. ______________ name ______________

index ______________ biblio ______________

▶ COPIES: [✓ Archives] Aero Dewey [✓ Eng] Hum

Lindgren Music Rotch Science

TITLE VARIES: ▶□ ______________

NAME VARIES: ▶□ ______________

IMPRINT: (COPYRIGHT) ______________

▶ COLLATION: 100 P

▶ ADD: DEGREE: S.M. ▶ DEPT.: E.E.

SUPERVISORS: ______________

NOTES:

cat'r: date:

page:

▶ J1581160

▶ DEPT: O.E.

▶ YEAR: 1999 ▶ DEGREE: S.M.

▶ NAME: YU, Guoxin