1 Introduction

Non-Uniform Rational B-Splines (NURBS) are a de-facto industry standard for free-form shape representation. Automatic geometric trimming of NURBS surfaces has long been desired by engineers in many fields, especially in aircraft design. Having a clean mathematic definition of an aircraft model will enhance the communication among different disciplines in the design process. “Disciplines” in aircraft design refer to specific technical subsets of the design such as aerodynamics, structures, propulsion, etc. Usually aircraft design has three stages: conceptual design, preliminary design and detailed design. Due to the improvement of computer capabilities and technical progress in many disciplines, these three stages are no longer clearly separated, especially conceptual and preliminary design. Having a geometric model that is suitable in the conceptual design process and can be easily exported to models used in the other two stages has significant meaning. An approach to achieve this is to use NURBS to represent component surfaces and apply “geometric trimming” rather than “visual trimming” to obtain a clean representation of the aircraft. An effective trimming algorithm is given in this dissertation. Applying optimization methods to reduce trimming errors is also addressed. Due to the complexity of aircraft geometry, this dissertation will focus on the geometric trimming of wings and fuselages, which are two typical component types. However, the process can also be extended and applied to other component types. This chapter states the problems, discusses the significance of this research, and describes the objectives and organization of this dissertation.

1.1 Problem Definition

NURBS have been widely used in aircraft design. The emergence of NURBS is due to a need for representing different types of curves and surfaces in a uniform format. Before 1950's, engineers "computerized" existing design and drafting methods, which were based on the use of conics [Limi44]. Later, J. Ferguson developed a spline package for Boeing's design software [Fari]. People realized that these two methods were incompatible. They had to be unified, and thus NURBS were born. NURBS can represent most parametric and implicit curves and surfaces including all conics and quadrics without loss of accuracy. However, due to the mathematical complexity, intersecting and trimming of NURBS surfaces has been a problem for a long time. In most of the literature, "trimming" refers to tessellating only the visible portions of surfaces. This type of trimming is called "visual trimming". The other type of trimming, "geometric trimming", trims the NURBS surfaces geometrically and creates new, mathematically clean representations of NURBS surfaces. The unwanted parts no longer exist.

The fundamental idea of the geometric trimming is to find the surface intersections and resample the surface points of the retained part, then re-interpolate these points. The first step is to find the intersection of two NURBS surfaces. The existing intersection algorithms focus on rendering and do not consider the re-sampling and re-interpolation process. They are suitable for visual trimming. Thus an intersection algorithm, which aims at improving the efficiency for geometric trimming, is needed. Geometric trimming might change the topology of a surface. A single surface is not capable of representing a surface trimmed by a closed curve. The trimming curve must be subdivided and the surface needs to be divided into corresponding patches. Some schemes have been applied but cause great errors. A new method to subdivide surfaces, which are trimmed by closed curves, needs to be developed.

Furthermore, the trimmed surfaces and the corresponding portions from the original surfaces do not match exactly due to the remarkably high degree of their resulting intersection curves. Factors such as sampling locations and knot sequences have

direct influence on the trimming errors. A study of the influence of these factors is thus desired. With increased degree or number of control points, the errors decrease. However, the general rule of surface interpolating indicates when globally fitting large numbers of points, the higher the degree and the more control points, the worse the wiggle. Moreover, the geometric definition is directly linked to the number of design variables in a design optimization process. If the trimmed surfaces serve as a baseline for a design, a large number of design variables slow down the design optimization process dramatically and cause difficulties in convergence.

Representing surfaces in NURBS is highly flexible. Although NURBS have existed for decades, most applications are based on nonrational B-splines in which weights are set equal. Weights, unlike control points, are difficult to visualize. Many operations involving derivatives of NURBS are more complicated than those of nonrational B-splines. The flexibility of NURBS interpolation leaves a lot freedom. The problem remains whether varying the weights is effective in reducing errors while keeping a fixed number of control points.

The problems mentioned above are related to geometric trimming per se. The application in aircraft design needs to address additional issues, such as component parameterization, and lofting of parametric, geometric models. The wing component has conventional parameters and lofting methods. The parameterization of the fuselage varies in different applications. This dissertation employs an existing method, which represents a fuselage with three subcomponents, and provides a new control-hull based method, which represents a fuselage using a single NURBS surface. These two types of fuselage components also allow the creation of a wide variety of surfaces for testing the geometric trimming algorithm.

In brief, the geometric trimming of NURBS surfaces represented aircraft components is a promising field but lacks investigation. This dissertation concentrates on providing a general, efficient and robust geometric trimming algorithm, especially suitable for aircraft design. Novel global optimization methods are applied to reduce the trimming errors.

1.2 Significance

A direct beneficiary of the clean representation of the geometry is CFD analysis. Traditional aircraft design uses analytical methods, empirical and semi-empirical analysis, wind tunnel tests and other historical data. Computational fluid dynamics (CFD) will bring more confidence to the design. However, CFD tools require detailed definition of geometric shape. Some errors, which will not be detectable in rendering or some non-CFD analysis methods, will cause grid generation tools to fail. Usually, CFD is used in preliminary design and detailed design. Recently, researchers are seeking approaches to bring CFD to conceptual design [Maso98]. One of the bottlenecks of these approaches is to rapidly create geometry, which is suitable for CFD. The traditional method is to trim aircraft models manually with conventional CAD tools before they can be input to the automatic mesh generation tools. This is a time consuming operation and requires significant expertise in CAD software. Geometrically trimmed surfaces created in modelers for conceptual design can be exported to traditional CAD systems by using standard file formats, such as IGES, though geometric trimming does not find its application only in conceptual design. A general algorithm should be applied to both conceptual design modeling tools and traditional CAD systems.

NURBS are suitable for aerodynamic shape optimization. Shape optimization finds its optimum shape for a given structural layout. By setting control points and weights as design variables, a great variety of shapes can be explored in a comparatively short time. Currently, most of the research is on NURBS curves since optimization of high-dimension configurations is a challenging task. With the advances in optimization methods, it is foreseeable that NURBS surfaces can be applied in aerodynamic shape optimization. This dissertation's exploration of optimizing trimmed surfaces to reduce the trimming errors also lays the foundation for future application in aerodynamics shape optimization.

A clean mathematical representation of the trimmed surfaces also benefits the calculation of important geometry parameters such as volume, wetted area, and mass

properties. In summary, geometric trimming would help aircraft designers obtain results more efficiently and accurately and thus allow designers to experiment with more configurations.

1.3 Terminology

Some aircraft geometric modeling is based on constructing surfaces from curves. Different curve functions are provided to add more options to traditional CAD systems. Analytic functions and polynomials are used. Curve generation is the backbone of this method [Sobi97, Trap99]. In the conceptual design stage, parametric geometric modeling is preferred. It is based on components instead of curves. "Component" is the basic geometry unit. Curves are defined by component shape parameters. For example, a wing is defined by span, root chord length, tip chord length, dihedral, twist, and root/tip airfoil type. These shape parameters, instead of curves, are used to construct a wing component. After obtaining surface points from component models, NURBS surfaces are created by interpolating these surface points. The trimming of NURBS surfaces removes the unwanted portions from the mathematical presentations. The trimmed surfaces are optimized to minimize the errors between these surfaces and the original surfaces. The term "shape optimization" is used in many areas. In geometric modeling, the objective of shape optimization is to find a more accurate surface representation. While in aircraft design, shape optimization may be to find surfaces that give better aerodynamic performances. Unless specifically stated, shape optimization in this dissertation refers to the use in the geometric modeling.

1.4 Dissertation Objectives

The goal of this dissertation is to create a method and algorithms for trimming NURBS represented fuselage and wing surfaces geometrically. The method can be

applied to trimming of general NURBS surfaces if certain conditions are satisfied. The breakdown of the process is listed below:

- Determine the parameterization and create models of fuselages and wings.
- Create a NURBS surface intersection algorithm specifically suitable for geometric trimming.
- Create an algorithm for geometric trimming NURBS surfaces of fuselages and wings.
- Evaluate trimming errors and discuss the factors, which influence the trimming results.
- Employ global optimization methods suitable for reducing trimming errors.

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows: Chapter 2 is the literature review of NURBS surface geometric trimming and parametric, geometric modeling of aircraft fuselage and wing components. Chapter 3 discusses the parameterization and surface lofting of these components. Chapter 4 describes an improved NURBS surface intersection algorithm, which is created specifically for geometric trimming. Chapter 5 describes the novel geometric trimming algorithm. The trimming errors are investigated in Chapter 6. A hybrid optimization method is used to reduce these errors. Chapter 7 gives the implementation details of this approach. Trimmed wing surfaces are optimized. Chapter 8 draws conclusions and discusses future research.

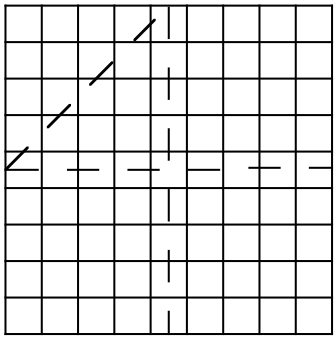
2 Literature Review

The procedure of creating and geometric trimming of fuselage and wing surfaces can be stated as: first, create surface points of fuselage and wing components from parameters, interpolate surface points into NURBS surfaces, and then obtain the surface intersection and reconstruct surfaces for the retained portions defined by the original surfaces and trimming curves. Only limited publications can be found on geometric trimming of NURBS surfaces. They will be reviewed first. Then the background of trimming errors and the optimization methods used in geometric modeling will be provided. Following that is the review of related research, which includes geometric modeling and parameterization of aircraft components.

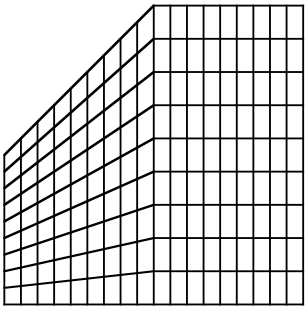
2.1 Intersecting and Trimming NURBS Surfaces

Geometric trimming was first addressed by Hoscheck, et al, who divided the parametric space of a given B-spline surface into rectangle subsets [Hosc87]. Applegarth [Appl89] developed an approach later by clipping each isoparametric curve lying on a bicubic B-spline surface and creating the clipped curve to construct trimmed patches. Consider two surfaces $S_1(u_1,v_1)$ and $S_2(u_2,v_2)$. Let $v_2 = const$, the intersection can be given by $S_1(u_1,v_1) - S_2(u_2,v_2) = 0$. There are three unknowns and three equations in x, y, z coordinates. The Newton's method is applied to solve these equations. With the u_2 at intersection known, the isoparametric curve $v_2 = const$ can be truncated. Then re-sample the surface points, and interpolate these points to create a new surface. Applegarth also discussed other cases when the trimming curve does not pass opposite boundaries. Decomposition of the original surface is resorted to solve these cases. Figure 2.1 and 2.2 show the decomposition for cases in which trimming curves pass the adjacent boundaries or trimming curves are closed. The method of obtaining intersections is not efficient and

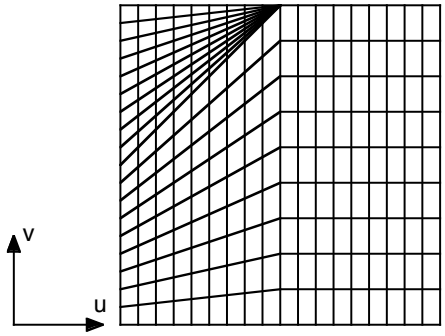
robust, and the decomposition of the closed trimming curve breaks the curve into four segments, at whose joint the continuities are changed.



(a) Clipping curve cutting adjacent boundaries



(b) Resulting patches



(c) Patches reassembled

Figure 2.1 Surface decomposition for open trimming curves passing adjacent boundaries ([Appl89])

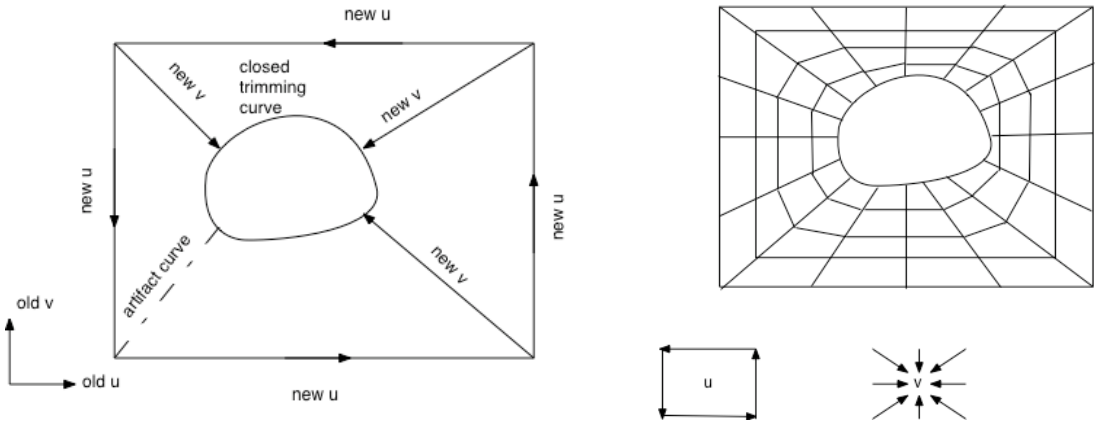


Figure 2.2 Surface decomposition for closed trimming curves ([Wang01])

Rojas' method differs from Applegarth's in the way of finding intersections [Roja94]. Rojas uses Peng's divide-and-conquer method [Peng84] and also includes Lasser's [Lass86] contribution on intersection of the Bernstein-Bezier representation. Two surfaces are divided into small patches iteratively until they can be considered as planar polygons. The intersection is then computed by intersecting these polygons. Intersection points are sorted to create the intersection curve. Then each isoparametric curve is trimmed by the intersection curve. Resampling surface points in the retained portion and interpolating these points create trimmed surfaces. Rojas uses the same scheme as the one in Applegarth's method to decompose the original surface. His approach gives a good result only in the open trimming curve cases. The trimming error is much higher when a trimming curve is closed since the continuity of the closed trimming curve cannot be maintained after decomposition.

Wang changed the scheme for closed trimming curves. His approach converts a closed trimming curve to two open trimming curves (see Figure 2.3), which provides a more accurate approximation. Intersection of two B-spline surfaces is not addressed in [Wang 01].

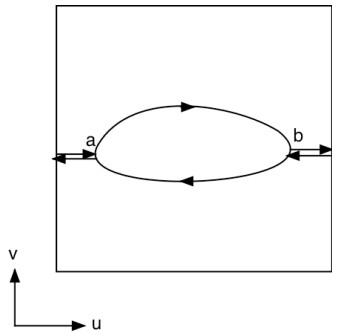


Figure 2.3 Conversion of a closed trimming curve to two open trimming curves by Wang's approach

As the intersection algorithm plays an important role in geometric trimming, different intersection algorithms will be reviewed below.

Newton's method is a classic root-finding algorithm. It uses the first few terms of the Taylor series of a function in the vicinity of a suspected root. Finding the intersection is actually a three dimensional problem. Consider the system $\begin{cases} f_1(x,y,z)=0 \\ f_2(x,y,z)=0 \\ f_3(x,y,z)=0 \end{cases}$, the initial approximation is P_0, to find the P_{k+1} from P_k, the steps are

1. Evaluate the function $F(P_k)=\begin{Bmatrix} f_1(x_k,y_k,z_k) \\ f_2(x_k,y_k,z_k) \\ f_3(x_k,y_k,z_k) \end{Bmatrix}$

2. Evaluate the Jacobian $J(P_k)=\begin{pmatrix} \frac{\partial}{\partial x}f_1(x_k,y_k,z_k) & \frac{\partial}{\partial y}f_1(x_k,y_k,z_k) & \frac{\partial}{\partial z}f_1(x_k,y_k,z_k) \\ \frac{\partial}{\partial x}f_2(x_k,y_k,z_k) & \frac{\partial}{\partial y}f_2(x_k,y_k,z_k) & \frac{\partial}{\partial z}f_2(x_k,y_k,z_k) \\ \frac{\partial}{\partial x}f_3(x_k,y_k,z_k) & \frac{\partial}{\partial y}f_3(x_k,y_k,z_k) & \frac{\partial}{\partial z}f_3(x_k,y_k,z_k) \end{pmatrix}$

3. Solve the linear system $J(P_k)\Delta P=-F(P_k)$ for ΔP

4. Compute the next approximation $P_{k+1}=P_k+\Delta P$

To find surface intersection, this method is used to solve the system $S_1(u_1, v_1) - S_2(u_2) = 0$. The initial value needs to be close to the root in order to make the iterative process converge. This method does not converge in some situations when we cannot make a close approximation for the initial values.

Peng's algorithm is a divide-and-conquer method [Peng84]. The underlying idea is: both surfaces involved are repeatedly subdivided until they can finally be approximated by planar polygons. The process of finding the intersection lines is then carried out between these planar polygons. In Peng's method each surface is organized as a b-branch tree. For each intersection line, an initial point is detected after a depth-first search along one tree. Extrapolation methods are then used to trace the entire length of the line, thus the line appears naturally in a continuous form. By the employment of an adaptive division strategy and by the careful choice of the representation basis of the patches on both surfaces, this algorithm achieves high efficiency.

Lasser's algorithm is slightly different from Peng's. Although this algorithm is for Bezier surface intersection, it provides the insight for B-spline surface intersections since B-spline surfaces can be converted to Bezier surfaces by knot insertion. In Lasser's algorithm, the polygon-polygon intersection is done by 'three point' (triangles) instead of quadrilaterals. The computational time and accuracy of the subdivision technique are also discussed in [Lass86]. A subpatch is approximated by the control hull when it is considered planar, and whether the subpatch is planar is determined by a tolerance, or maximum thickness. The tolerance affects the accuracy. Lasser shows when the maximum thickness is 1/4 of the original, the CPU time increases by a factor of two.

With a given tolerance, Peng's and Lasser's algorithms obtain a large number of small curve segments. Connecting these curve segments is expensive and tolerance sensitive. These methods are suitable for rendering and do not aim at obtaining intersections for geometric trimming.

2.2 Trimming Errors

The geometrically trimmed surface does not match the original surface exactly. Metrics are needed to tell how well the trimmed surface approximates the original one.

Jain studied the error visualization in comparison of B-spline surfaces [Jain99]. He uses surface plots and ordinate intercept plots for vector difference in position values. The error is measured by the maximum distance between two surface points with uniformly spaced parameters. For example, the error between two curves at the parameter value u is given by $e(u) = \sqrt{E_x(u)^2 + E_y(u)^2 + E_z(u)^2}$, where $E(u) = P(u) - Q(u)$, $P(u)$ and $Q(u)$ are corresponding curve points and the u ranges of the two curves are the same. The surface plot gives the errors at different u and v locations on a surface. To create an ordinate intercept plot, linear regression is used to fit a straight line to the error data of an isoparameteric curve (for example, the curve with parameter $v = v_i$). Each line intercepts the ordinate and the values from different lines are then plotted versus the parameter v. This method reduces the information compared to providing surface plots. Fourier analysis is also employed on the errors of isoparametric curves. The sum of the first six Fourier coefficients is plotted versus the parameter v. A preponderance of high frequency components suggests a poor match.

When comparing the curves at the same parameter values, the result is parameter dependent. Re-parameterizing will give different values though the curve stays the same. Fritsch discusses the problem of determining the distance between parametric curves [Frit92]. The major problem is to find a way to correspond curve points on two different curves. This method subdivides both curves into the same number of segments and associates points at the segment joints. Each segment on a curve has the same chord length. Since the curves are implicit, the joints of segments are estimated by linear interpolating curve points of uniformly distributed parameters. Mathematically, the parameter-independent metric is defined as:

Let $P(t)$ $(a \le t \le b)$ represent one of the curves. Divide interval [a,b] into m subintervals of equal length $dt = (b-a)/m$. Let $t_j = a + j \cdot dt$ and evaluate $P_j = P(t_j)$, $j = 0,...,m$. Define

$$\begin{aligned} d_0 &= 0 \\ d_j &= d_{j-1} + \delta(P_{j-1}, P_j), \quad j = 1,...,m \end{aligned} \tag{2-1}$$

where $\delta(P,Q)$ is the Euclidean distance between points P and Q.

Divide interval $[d_0, d_m]$ into n subintervals of equal length $da = (d_m - d_0)/n$. Let $a_k = d_0 + k \cdot da$, $k = 0,...,n$. Compute

$$s_k = t_j + (a_k - d_j)\frac{t_{j+1} - t_j}{d_{j+1} - d_j}, \; k = 1,...,n-1 \tag{2-2}$$

where $d_j \le a_k \le d_{j+1}$ (linear interpolation in t versus d table.) Then evaluate $\tilde{P}_k = P(s_k), \quad k = 0,...,n$. Follow the same procedure on the other curve to obtain $\left\{\tilde{Q}_k\right\}$. The final comparison metric will be some vector norm applied to the sequence $\delta_k = \delta(\tilde{P}_k, \tilde{Q}_k) \quad k = 0,...,n$. When the max-norm is used $dist(P,Q) \approx \max_{0 \le k \le n} \delta_k$. The scheme is shown in Figure 2.4. This method can be extended to measure surface errors. However, no publication has been found.

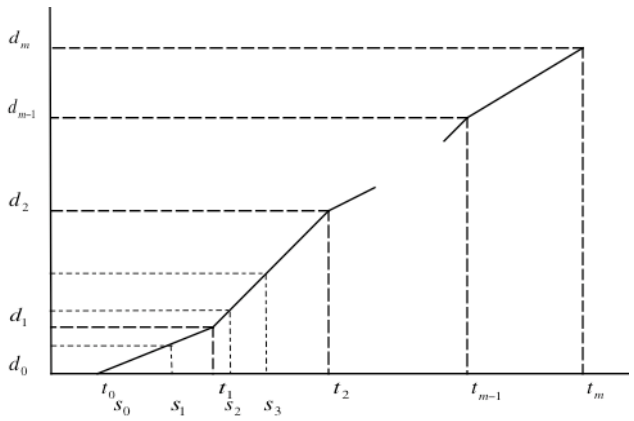


Figure 2.4 Finding distance between two parametric curves

An other method given by Piegl [Pieg97] is to project the original surface points to the trimmed surface. The error metrics can be defined based on the distance between the corresponding points. The projection points are found by Newton's method iteratively. In Peigl's method, the original surface need not to be a parametric surface. It can even be described as discrete surface points.

2.3 Optimization of NURBS Surfaces

Braibant and Fleury showed that B-splines are well suited for shape optimization [Brai84]. The three parameters that can be used to modify a NURBS curve or surface are knot sequences, control points location and weights. Changing knot sequences can hardly give any geometric intuition and is still lacking investigation. Heuristically, the shape of the curve/surface can be improved by using the following hierarchy of parameterization methods: uniform, chord length, and centripetal. Control points and weights are usually modified to change curve/surface shapes. They are variables in optimization processes. The objective is to minimize the error between the target surface and the NURBS surface. The errors between the original surface and the NURBS represented surface are used to construct objective functions.

NURBS surfaces are also suitable for aerodynamic shape optimization in aircraft design. The geometry is linked directly to the flow solver and the objective functions will be based on aerodynamic performance. The other approach is to get target curvature distribution from iterative design and optimize the NURBS surface to achieve the target. Using NURBS surfaces in aerodynamic shape optimization can reduce the number of design variables significantly and ensure good smoothness properties. They are currently being explored. However, this is out of the scope of this dissertation.

Piegl proposed two methods to vary the shape: interactive control points location-based and weight-based modification [Pieg89]. Juhasz provided a weight-based shape

modification method by means of which one can prescribe not only the new position of an arbitrary chosen point of a planar NURBS curve but also the tangent direction [Juha99]. Au and Yuen present shape modifications achieved by the simultaneous modification of control points and weights [AuCK95]. The methods mentioned above need users' interaction. They give more insight into the effects of modifying NURBS control points and weights, but are not suited for an automatic process.

Several authors have addressed the shape modification problem through optimization. Ferguson and Jones proposed methods to control curvature by formulating a constrained nonlinear optimization problem using the coefficients of B-splines in nonrational form as design variables [Ferg86]. Moreton and Sequin studied the application of nonlinear optimization techniques to minimize a fairness functional based on the variation of curvature [More92]. Hu, Li and Zhu provided two methods of modifying the shape of NURBS surfaces by constrained optimization and energy minimization [HuSM01]. Hohenberger and Reuding investigated the possibilities of entering the weights in an automatic fairing process [Hohe95]. Baker et al. applied DIRECT method for parallel global aircraft configuration design space exploration [Bake01]. The Quasi-Newton BFGS method has been applied to optimize NURBS represented wing profiles by Trepanier, et al [Trep00, Lepi00, Lepi01]. Most prior work focuses on curve shape optimization and local optimization methods due to the high dimensionality and complexity of the surface optimization problem.

2.4 Surface Modeling of Aircraft Components

Aircraft conceptual design has directly benefited from advances made in aircraft geometric modeling. A sophisticated aircraft model offers not only a clear visual representation of the design but also a detailed geometry, which can be utilized by higher order analysis methods [Glou96].

Samareh discussed the status and future of geometry modeling in [Sama99]. The modeling techniques used in CAD systems are mainly NURBS, solid modeling (SM) and feature-based solid modeling (FBSM). Traditional CAD systems represent geometry in many mathematical forms, such as Bezier, Coons patches, B-spline curves and surfaces. However, many of the formats can be represented by NURBS without loss of accuracy. Most SM CAD systems use either a boundary representation (B-rep) or constructive solid geometry (CSG) method to represent a physical solid object; FBSM is a technique that adds features to SM. Features are dimension-driven objects, such as holes, slots, bosses, fillets, chamfers, sweep and shell. Both SM and FBSM are solid modeling techniques. NURBS is a surface modeling technique. In comparing solid models with surface models, solid models need more computer resources. In the applications, which only require skin surfaces, NURBS surface representation is a better choice. Samareh also did a detailed survey of shape parameterization techniques for multidisciplinary optimization in [Sama01]. In this paper, eight shape parameterization technique categories have been discussed. They are: basis vector, domain element, partial differential equation, discrete, polynomial and spline, CAD-based, analytical and free-form deformation (FFD). NURBS are included in the polynomial and spline category. The author has mentioned selecting both NURBS control points and weights as variables in the optimization process.

In the conceptual design process, designers cannot afford the time to use a traditional CAD system. Thus parameter controlled geometric modeling tools have evolved in the field of conceptual design. In a nonparametric CAD system, individual points, curves and surfaces must be added and modified to produce a model, while in a parametric system, aircraft components can be added and modified with parameters which are familiar to aircraft designers. The “parametric” in the term “parametric geometric modeling” means using simple parameters such as length, thickness and angle to define shapes. This is different from the term “parametric representation of curves or surfaces”, which refers to presenting curves or surfaces in a parametric mathematical format. One of the popular aircraft design parametric geometry modeling tools is

ACSYNT. Parametric surfaces representing the aircraft components are automatically generated from the aircraft geometry design parameters. B-spline surfaces allow curvature continuity for individual components. By representing component surfaces uniformly with B-splines, the intersections of aircraft components can be calculated with a general algorithm [Jone91, ACSY93, Flem92a, Flem92b]. Rapid Aircraft Modeler (RAM) is another parametric geometry modeler for aircraft conceptual design. In [Glou96], Gloudemans introduced the design and application of RAM.

3 Parameterization of Fuselages and Wings

Some aircraft geometric modeling software is based on constructing surfaces from curves. Different curve functions are provided to add more options to CAGD systems. Some analytic functions and polynomials are used. Curve generation is the backbone of this method. Another method is based on components instead of curves. Component is the basic geometry unit. Only few parameters, which are familiar to aircraft designers, are used; some information needed to fully define a surface is predetermined. This chapter focuses on component-based parameterization of two major components -- fuselage and wing. The parameterization of wing components follows the convention in aircraft design.

3.1 Introduction

The traditional method to define external surfaces of fuselages is by defining cross-sections and longitudinal profiles. The longitudinal profiles determine the characteristic points of cross-sections. For an axisymmetric surface, there is one longitudinal control curve, and the surface can be constructed by rotational sweep. For more complex cross-section types, more longitudinal control curves are involved.

To define these longitudinal or cross-sectional curves is challenging and time-consuming. Some aircraft geometric modeling software adds more options to traditional CAD systems to provide a wide variety of curve types [Sobi97, Trap99]. Curves are defined in a piecewise manner. However, to satisfy continuity, especially C2 continuity at curve joints is very difficult. Some software predefines the curve types and only exposes a few parameters to users. Component is the basic geometry unit. There is always a tradeoff between the design time and the level of detail provided. During the initial

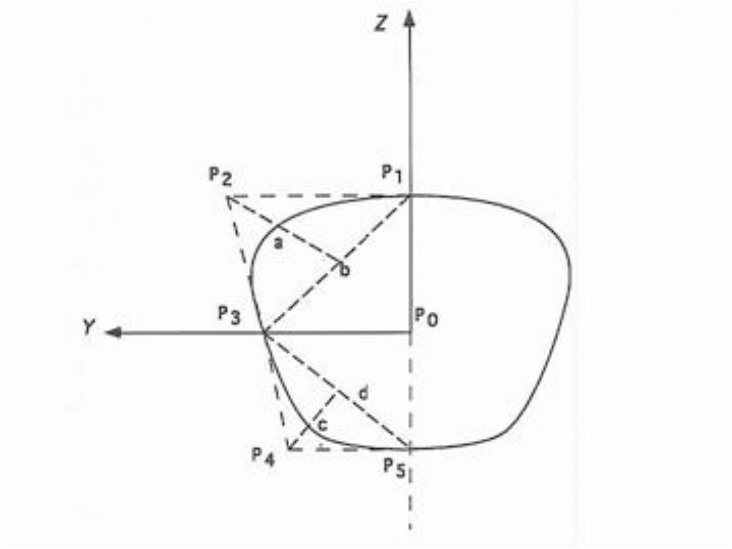
stages of design a small set of parameters is desired, so that design decisions can evolve quickly.

3.2 Parameterization of Fuselages

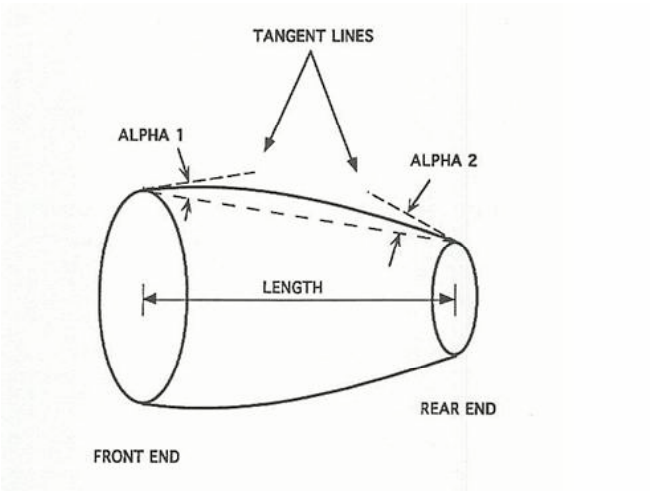
In this section, parameterization of fuselages in selected software is reviewed. Then a control-hull based parameterization is presented.

3.2.1 Longitudinal and Cross-sectional Shapes

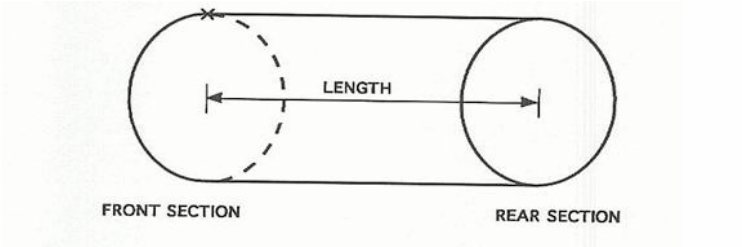
In ACSYNT [Mykl93, Mykl94], a fuselage consists of three components: nose, midsection and afterbody (Figure 3.1). Each of them can have elliptical or conic cross-sections. Elliptical cross-section is a special case of conic cross-section, however, it is more easily defined with fewer parameters and it is listed as a different type. Conic cross-sections are a composition of four curve segments. Due to symmetric characteristics, only two sections need to be defined for each conic cross-section. They are given by P_1, P_2, P_3, P_4, P_5, ρ_1, ρ_2 in Figure 3.1(a), where $\rho_1 = \frac{|ab|}{|P_2 b|}$ and $\rho_2 = \frac{|cd|}{|P_4 d|}$. The nose and afterbody longitudinal profiles are defined by parabolas. A nose tip angle and a shoulder angle define the tangent lines at the two end points. α_1 and α_2 define the two tangents for an afterbody.



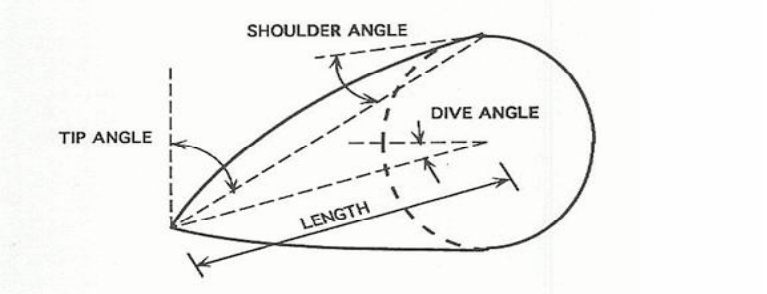
(a) Conic cross-section



(b) Afterbody



(c) MidSection



(d) Nose

Figure 3.1 Fuselage component in ACSYNT

The computation of the component geometry involves basic geometry operations such as calculating the elliptical and conical curve points and tangents, line-line intersection, and curve interpolation. For example, surface points of a Nose component, which is a subcomponent of a fuselage, with elliptical cross-sections are determined as follows and illustrated in Figure 3.2.

1. Determine the two radii of each cross-section.

   On the XZ plane, define the tangent line, AB, by the tip angle, define the tangent line, CD, by the shoulder angle. Calculate the intersection, E, of AB and CD. Define a quadratic Bezier curve by setting the three control points as A,E,C. When the weights of the three control points are 1, it is a nonrational quadratic Bezier curve, or parabola. Given an X value, the Z value of the curve, which is one radius of the cross-section, can be calculated.

   On the XY plane, repeat the steps for the XZ plane and get the other radii of each cross-section.

2. Determine the center of each cross-section.

   If the dive angle is zero, the center of each cross-section is on the X axis. Otherwise calculate the center location on the XZ plane with dive angle and Length.

3. Calculate the points and tangents on each cross-section with the two radii and the center on the X axis. The cross-sections are parallel to the YZ plane.

4. Interpolate the surface with B-splines.

In general, the cross-sections can be arbitrarily located and do not need to be parallel to the XY, YZ or XZ plane, and the centers of the cross-sections do not need to be collinear. In order to be interpolated with tensor product B-spline patches, the number of points on each cross-section must be the same.

By giving the parameters of tip angle, shoulder angle, dive angle, length and two radii of the rear cross-section, the surface points of a Nose component are determined.

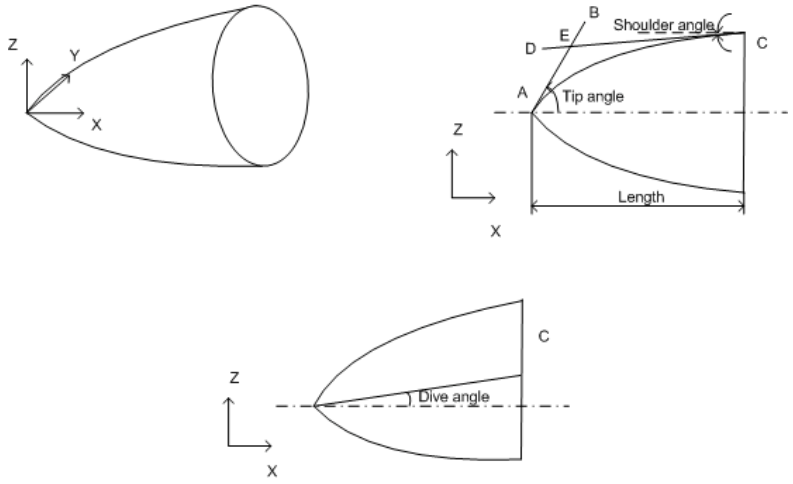


Figure 3.2 Lofting of the nose component

In RAM, a fuselage is a single component [Glou96]. The longitudinal profiles of nose and afterbody are defined by super ellipses. The section between nose and afterbody is interpolated linearly. The cross-sections can be a point, circle, ellipse, or box.

OSCAR also allows rapid production of aircraft shapes by parametric modeling [Char97]. The fuselage in OSCAR is also produced as three separate surfaces: the nose cone, the middle cylinder and the tail cone. The nose cone is half of a biparameter ellipsoid, with poles at top and bottom, and a major axis that varies with a double ellipse

shape. The middle cylinder is a biparameter cylinder locally modified to produce stretched cabs, wing-root fairings and extended wing-boxes. The tail cone is an ellipsoid sheared upwards.

In brief, the common longitudinal profiles of a nose cone include ogive, ellipse, parabolic, super ellipse, etc [Crow, Maso]. The longitudinal profiles of afterbodies are similar to those of nose. Midsections of fuselage are usually represented by cylinders or cut cones. The cross-sectional shapes are usually circular, rounded box, or elliptical. General second-degree curves can be represented by conic sections, such as Figure 3.1(a).

The parameters are defined by the requirements of aerodynamic, structural analysis and surface creation. The more detailed the geometry is, the more parameters will be needed. Very fine modification should be carried out in traditional CAD systems.

3.2.2 Control-Hull Based Parameterization

For both longitudinal and cross-sectional curves, second-degree curves are often used. They are easy to envision and define. However, to have curvature continuity, degree of three or higher is required. Piecewise polynomial/rational surfaces have many advantages and are well suited to interactive shape design. The control polygon approximates the surface shape. Thus, this section uses parameters, which are related to the control hull of bicubic NURBS surfaces, for defining a fuselage component. The whole fuselage surface is represented by a single NURBS surface.

Fuselage cross-sections are usually closed and symmetric. A simple control hull for a fuselage surface is created considering these properties. Not all the control points are free. Some constraints can be added. $P_{i,j}$ are the control points.

The cross-sectional curve is closed:

$$P_{i,0} = P_{i,m-2}, \quad P_{i,1} = P_{i,m-1}, \quad P_{i,2} = P_{i,m} \quad i = 1,...,n$$

The cross-sectional curve is symmetric:

$P_{i,m/2-j}$ is a mirror image of $P_{i,m/2+j}$ with respect to x - z plane $\quad j = 1,...,m/2-1, m$ is even

If a single control point is moved, both the cross-sectional and longitudinal curve shape will change. So the control points, which are not on the upper crown line and lower crown line, are defined by the relative location to the upper and lower crown line. When these parameters are defined, the cross-sectional control points will only be scaled. Using the proportional parameters will reduce the influence of moving upper and lower control points to the cross-sectional shape. The designers can focus on adjusting the upper and lower crown line after defining the cross-sectional shape. A NURBS fuselage surface and its control hull is shown in Figure 3.3.

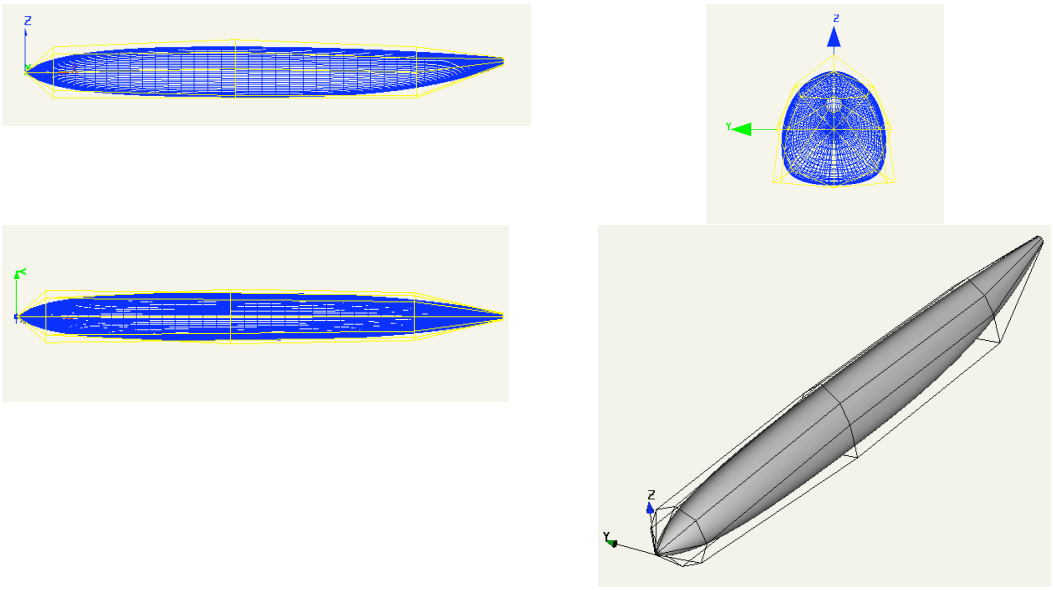


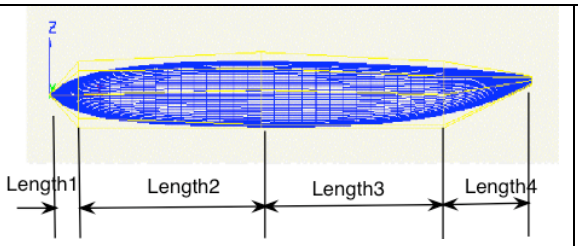
Figure 3.3 B-spline surface of a fuselage and its control hull

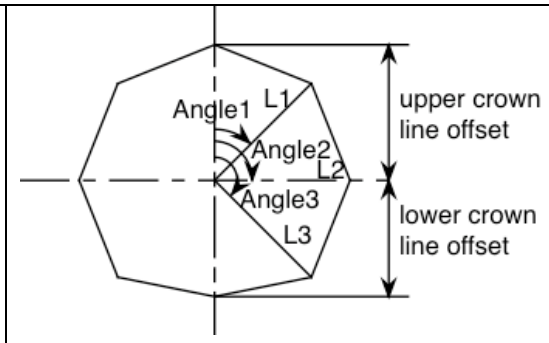
3.2.3 Fuselage Component Parameters

In this dissertation, two types of fuselage components are used.

The parameterization of fuselages with three subcomponents is similar to ACSYNT (Figure 3.1). A wing intersects a fuselage in the midsection. In the remainder of this dissertation, the nose and the afterbody are usually ignored, and due to the symmetry, only half of the midsection is displayed.

The parameterization of fuselages with a single surface is control-hull based. The parameters are listed in Figure 3.4.





L = upper crownline offset + lower crownline offset

$\rho_1 = L_1 / L$

$\rho_2 = L_2 / L$

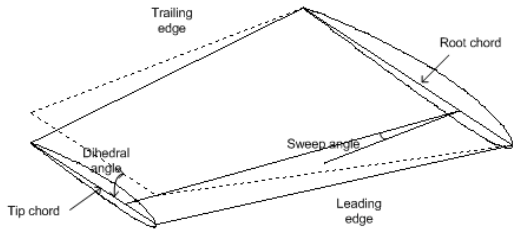
$\rho_3 = L_3 / L$

Figure 3.4 Parameters of a control-hull based fuselage component

3.3 Parameterization of Wings

3.3.1 Reference Wing

Wing geometry is well defined in aircraft design. The basic wing geometry used to begin the layout is the "reference" wing. The reference wing is fictitious and extends through the fuselage to the aircraft centerline. The root airfoil is at the centerline. Aspect ratio, taper ratio and sweep determine the reference wing shape. Figure 3.5 shows the key parameters.



Chord (C) -- distance from leading edge to trailing edge
Span(s) – distance between wing tips
Aspect ratio (A) = b^2 / s
Taper ratio (λ) = C_{tip} / C_{root}
Twist – the change of incidence measured with respect to the root airfoil.
Dihedral – the angle of the wing with respect to the horizontal when seen from the front.
Incidence – the pitch angle of the wing with respect to the fuselage.

Figure 3.5 Wing component layout

3.3.2 Cross-sectional Shapes -- Airfoil

NACA four-digit airfoils are obtained by combining the mean line and thickness distribution [Abbo59]. The first digit integer indicates the maximum value of the mean-line ordinate in percent of the chord. The second integer indicates the distance from the leading edge to the location of the maximum camber in tenths of the chord. The last two integers indicate the section thickness in percent of the chord. The thickness distribution is given by

$$\pm y_t = \frac{t}{0.20}\left(0.29690\sqrt{x} - 0.12600x - 0.35160x^2 + 0.28430x^3 - 0.10150x^4\right) \quad (3\text{-}1)$$

where t is the maximum thickness of the section in fraction of chord.

The leading edge radius is $r_t = 1.1019t^2$.

the mean line is

$$\begin{cases} y_c = \dfrac{m}{p^2}(2px - x^2) & \textit{forward of } \max \textit{ ordinate} \\ y_c = \dfrac{m}{(1-p)^2}[(1-2p) + 2px - x^2] & \textit{aft of } \max \textit{ ordinate} \end{cases} \quad (3\text{-}2)$$

where m is the maximum coordinate of the mean line in fraction of the chord and p is the chordwise position of m.

The upper curve and lower curve coordinates are

$$\begin{cases} x_U = x - y_t \sin\theta \\ y_U = y_c + y_t \cos\theta \end{cases} \tag{3-3}$$

$$\begin{cases} x_L = x + y_t \sin\theta \\ y_L = y_c - y_t \cos\theta \end{cases} \tag{3-4}$$

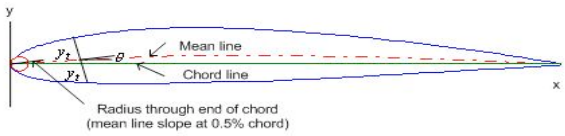


Figure 3.6 Wing profile

By the definition, at the trailing edge, the upper curve and the lower curve do not intersect. So the curve is not closed. When the wing surfaces are used in CFD analysis, with the Navier-Stoke methods, the trailing-edge point is forced to be the average of the upper and lower surface. Although some codes can handle the blunt trailing edge, many applications still require the trailing edge to be closed. When fuselages are being trimmed, the trimming curves, which are intersections of fuselages and wings, also need to be closed. So the point on the trailing edge is forced to be the average of the point computed from equation (3-3) and (3-4).

3.4 Summary

The fuselage parameterization methods are reviewed in this chapter. A control-hull based parameterization of fuselage is given. A small set of control points is used to define a fuselage surface. When applying NURBS surfaces to shape optimization, it gives good baseline geometry. The surface is C2 continuous and the number of control points can be increased by knot insertion to provide more flexibility. The wing parameterization follows the convention in aircraft design. The cross sections are NACA four digit airfoils.

4 Intersection Algorithms

Intersection algorithms for computing B-spline surface intersection have been reviewed in Chapter 2. In this Chapter, an algorithm suitable for geometric trimming is presented. The fundamental idea is to select isoparametric curves based on the characteristics of one surface and intersect these curves with the other surface. The intersection points are then connected to create intersection curves. A subdivision method is used to obtain such curve-surface intersection. The curve is subdivided until it can be approximated by a straight line and the surface is subdivided until the patch can be approximated by a quadrilateral within a given tolerance. Finally the intersection points are computed by a line-quadrilateral intersection algorithm. The number of intersection points depends on the number of isoparametric curves selected to do the intersection, and thus is controllable and independent of the error bound of intersection points. This property is well suited for the geometric trimming. The comparison of this algorithm to the divide-and-conquer method, which subdivides both surfaces, is given in the second section of this chapter.

4.1 The Intersection Algorithm

A NURBS surface is defined as:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \qquad (4\text{-}1)$$

The $P_{i,j}$ are control points, the $w_{i,j}$ are the weights, and the $N_{i,p}(u)$ and $N_{j,q}(v)$ are the B-spline basis functions defined on the knot sequences $U = \{u_0, u_1, ..., u_{n+p+1}\}$ and $V = \{v_0, v_1, ..., v_{m+q+1}\}$, where

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_i} N_{i+1,p-1}(u)$$

The p is the degree in the u direction and q is the degree in v direction. $N_{j,q}(v)$ is defined similarly to $N_{i,p}(u)$. If all the weights are the same, it refers to non-rational B-spline surfaces. Since weights are difficult to visualize and very little is known on setting weights, most often all weights are set to 1. This is a tensor product scheme, or a bidirectional curve scheme. When u or v is a constant, equation (4-1) corresponds to a NURBS curve (4-2), which is called an isoparametric curve.

$$C_{v^*}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i^* P_i^*}{\sum_{i=0}^{n} N_{i,p}(u) w_i^*} \qquad (4\text{-}2)$$

where $w_i^* P_i^* = \sum_{j=0}^{m} N_{j,q}(v) w_{i,j} P_{i,j}$ and $w_i^* = \sum_{j=0}^{m} N_{j,q}(v) w_{i,j}$.

Sometimes, it is convenient to use homogenous coordinates. NURBS are then given by

$$S^w(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j}^w$$

where $P_{i,j}^w = (w_{i,j} x_{i,j}, w_{i,j} y_{i,j}, w_{i,j} z_{i,j}, w_{i,j})$

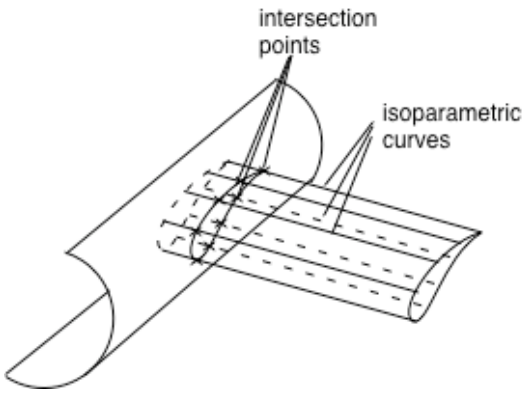


Figure 4.1 Illustration of fuselage-wing intersection

Consider two NURBS surfaces intersecting. Figure 4.1 illustrates a series of intersection points of the isoparametric curves on a wing surface. Connecting these intersection points, the intersection curve are then created. Thus the process starts with finding intersection points of NURBS curves and NURBS surfaces. The subdivision technique provides a good basis to solve the problem in a rather geometrical way and it is also robust. The curve is subdivided repeatedly until it can be approximated by a straight line and the surface is subdivided repeatedly until it can be approximated by a planar polygon. To reduce the computation time, curve segments and surface patches, which do not intersect, will be eliminated from the list immediately. Simple tests of intersection can be based on the intersection of convex hulls. NURBS have strong convex hull property, which is stated as:

Convex hull property for NURBS curves: (weights are >0)

If $u \in [u_i, u_{i+1})$, then $C(u)$ lies within the convex hull of the control points $P_{i-p},...,P_i$.

Convex hull property for NURBS surfaces:

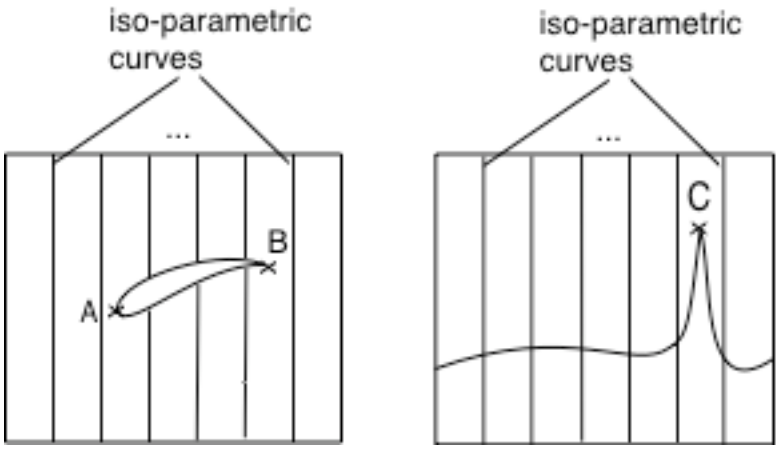
If $(u,v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, then $S(u,v)$ lies within the convex hull of the control points $P_{i,j}$, $i_0 - p \le i \le i_0$ and $j_0 - q \le j \le j_0$.

Thus if the convex hull of a NURBS curve and a convex hull of a NURBS surface do not intersect, the curve and the surface do not intersect. Furthermore, if the curve

convex hull of the control points $P_{i-p},...,P_i$ do not intersect the surface convex hull of the control points $P_{i,j}$, $i_0 - p \le i \le i_0$ and $j_0 - q \le j \le j_0$, then the corresponding curve segment and the surface patch do not intersect. For intersection tests, tight convex hulls are preferred since the non-intersecting curves and patches can be eliminated earlier in the process. A more efficient method is to add knots at each ends of the parametric interval so that the portion of the NURBS curve is redefined in a Bezier form. Rational Bezier curves may be viewed as the backbone of NURBS curves. Every NURBS curve can be broken down into a collection of Bezier curves. The details of transforming a NURBS into Bezier form is given in section 4.1.1. The same holds for surfaces. Every NURBS surface can be broken down into a collection of Bezier surfaces. Since convex hulls are irregular polyhedrons and the intersection test of two irregular polyhedrons is not trivial. Bounding boxes are constructed by using the x, y and z coordinates of the convex hulls. The description of the test can be found in Section 4.1.2. If a bounding box does not intersect with any bounding boxes of the other curve/surface, the corresponding curve/surface will be eliminated from the list immediately. Otherwise, the corresponding curve/surface will be subdivided, and smaller bounding boxes will be created. The detail of the subdivision is given in Section 4.1.3. When the control hull of a curve is close to a straight line or a control hull of a surface is close to a planar quadrilateral, the subdivision stops. The criteria are given in Section 4.1.4. Then an intersection point can be found by a line-quadrilateral intersection. A classical algorithm is used to compute the intersection. Since the intersection curves will be used for trimming in the next chapter, the corresponding intersection points in parametric (*u*-*v*) space are also needed. The Newton iterative method is applied to project points in Euclidean space to parametric spaces. This is described in Section 4.1.5.

When determining isoparametric curves from one surface, it is possible to miss some characteristic points. For example, if the intersection curve is closed, the iso-parametric curves in Figure 4.2(a) will not intersect at point A and B. In Figure 4.2(b), the sudden change of the curvature at point C will not be caught. The isoparametric curves need not be selected uniformly. A bisection scheme to find the characteristic

points is given in 4.1.6. In addition, constructing the intersection curves from intersection points is also a challenging work since the topology information is lost. The method given in 4.1.7 is not a generic one and the problem is solved under some assumptions.



(a) (b)

Figure 4.2 Missed intersection points

4.1.1 Conversion of NURBS Curves and Surfaces to Bezier Form

An *n*-th degree rational Bezier curve is given by

$$b(t) = \frac{\sum_{i=0}^{n} b_i w_i B_i^n(t)}{\sum_{i=0}^{n} w_i B_i^n(t)} \tag{4-3}$$

where b_i are control points, w_i are weights, and B_i^n are the Bernstein polynomials

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad i = 0,...,n$$

They satisfy the recursion

$$B_i^n(t) = (1-t)B_{i-1}^{n-1}(t) + tB_i^{n-1}(t); \quad B_{-1}^n(t) = B_{n+1}^n(t) \equiv 0$$

Every NURBS curve is an ordered collection of rational polynomial curve segments [Fari97]. Each of these may be represented in rational Bezier form and the

corresponding control points and weights are found by knot insertion. If we are interested a particular segment $[u_I, u_{I+1}]$ of a *n*-th degree NURBS curve, insert knot u_I and u_{I+1} until their multiplicities are $n+1$. The weights and control points corresponding to the segment are those used in Bezier form. The corresponding normalized parameter in Bezier form can be found by

$$t = \frac{u_{I+1} - u}{u_{I+1} - u_I}.$$

Bezier curves are usually defined on $[0,1]$, but it can also be defined over any interval $[a,b]$. Using the unnormalized interval avoids the normalization and the mapping of the normalized space to the original space when finding the intersection in parametric space. In our implementation, the Bezier curves/surfaces are defined on the intervals determined by the original intervals of NURBS curves/surfaces.

Knot insertion is a basic operation for a NURBS curve. Insert a knot *u* ($u_k \le u < u_{k+1}$) into the knot sequence. The new control points are given by

$$P_i^{w*} = (1-\alpha_i)P_{i-1}^w + \alpha_i P_i^w$$
$$\text{where} \quad \alpha_i = \begin{cases} 1 & i \le k-p \\ \dfrac{u-u_i}{u_{i+p}-u_i} & k-p+1 \le i \le k \\ 0 & i \ge k+1 \end{cases} \qquad (4\text{-}4)$$

The above equation can be generalized to insert a knot multiple (*r*) times [Pieg97]. Denote the *i*th new control point in the *r*th insertion step by P_{ir}^w

$$P_{ir}^w = (1-\alpha_{ir})P_{i-1\,r-1}^w + \alpha_{ir}P_{ir-1}^w$$
$$\text{where} \quad \alpha_{ir} = \begin{cases} 1 & i \le k-p+r-1 \\ \dfrac{u-u_i}{u_{i+p-r+1}-u_i} & k-p+r \le i \le k-s \\ 0 & i \ge k-s+1 \end{cases} \qquad (4\text{-}5)$$

s is the initial multiplicity.

To transform a NURBS curve segment with $u \in [u_I, u_{I+1}]$ to Bezier form, do knot insertion at u_I, u_{I+1} p times. To transform a NURBS surface patch with

$u \in [u_I, u_{I+1}]$ $v \in [v_J, v_{J+1}]$ to Bezier form, do knot insertion at u_I, u_{I+1} and v_J, v_{J+1} p times. For tensor product surfaces, many algorithms can be extended from the curve algorithms, often by processing the rows of coefficients in one direction and then processing rows in the other direction. So some algorithms are only given for curves in this chapter for the sake of brevity.

4.1.2 Bounding Box Intersection Test

A bounding box is built from the control hull of a transformed Bezier curve or surface. The edge of the bounding box is always parallel to any of the x, y and z axis. The minimum and maximum coordinates of the control hulls in x, y, z are used to set the corner points. The bounding box wraps the control hull. Although it hugs the curve/surface less tightly, it makes the test, otherwise a challenging task, very simple. If the projection of a box in any of the three planes (xy, yz and xz plane) does not intersect with the other box, the curve and the surface do not intersect.

4.1.3 Subdivision of Bezier Curves and Surfaces

Given a Bezier curve with control points $\mathbf{b}_i$ and the parameter over $[a,b]$, the parts of curve, which correspond to $[a,c]$ and $[c,b]$, can also be defined by Bezier curves. The control points of $[a,c]$ are given by $\mathbf{b}_0^j(c),\ j = 0,..,n$ and the control points of part $[c,b]$ are given by $\mathbf{b}_j^{n-j}(c)$, where

$$\mathbf{b}_i^r(c) = \sum_{j=0}^{r} \mathbf{b}_{i+j} B_j^r(c) \quad i = 0,...,n-r \tag{4-6}$$

and $B_j^r(c)$ are Bernstein polynomials.

In the intersection algorithm, the division point in parametric space is set to the middle point of the interval $[a,b]$. A curve is subdivided into two curves, while a surface is subdivided in both *u* and *v* direction and creates four patches.

4.1.4 Subdivision Termination Criteria

To determine whether a curve can be approximated by a line, or a surface can be approximated by a planar quadrilateral, the convex hull property is used again. For Bezier curves and surfaces, the convex hull property, which is similar to that of the B-spline curves/surfaces, states that the curves/surfaces are contained in the convex hulls of their defining control points.

Lane and Riesenfeld [Lane80, Wang91] proposed the convergence test for piecewise linear approximation of the curve,

$$d\left(\mathbf{b}_i, l\left(\mathbf{b}_0, \mathbf{b}_n\right)\right) < Tolerance \quad i = 1,2,...,n-1 \tag{4-7}$$

where $l\left(\mathbf{b}_0, \mathbf{b}_n\right)$ is the line segment from $\mathbf{b}_0$ to $\mathbf{b}_n$ and $d(\cdot,-)$ are Euclidean distances. Measure the distance of control points to the line determined by the start and end control points. If all of the distances are smaller than a tolerance, the control hull is considered to be tight to the line and the curve can be approximated by this line.

For Bezier surfaces, measure the distance of control points to the triangle determined by three corner control points. If all the distances are smaller than a tolerance, the control hull is considered to be tight to the plane and the surface can be approximated by the quadrilateral of the corner control points. That is

$$d\left(\mathbf{b}_{i,j}, p\left(\mathbf{b}_{0,0}, \mathbf{b}_{n,0}, \mathbf{b}_{0,m}\right)\right) < Tolerance \quad i = 1,2,...,n-1,\ j = 1,2,..,m-1 \tag{4-8}$$

where $p\left(\mathbf{b}_{0,0}, \mathbf{b}_{n,0}, \mathbf{b}_{0,m}\right)$ is the plane constructed from $\mathbf{b}_{0,0}, \mathbf{b}_{n,0}, \mathbf{b}_{0,m}$.

The criteria mentioned above terminates the subdivision for the case given in Figure 4.3 and approximates the curve with line $b_1 b_4$ (Figure 4.3 is an exaggerated illustration). Adding control of the overall size of the bounding box will let the subdivision go on but at the same time slows down the intersection process. As most curves/surfaces of airframe components do not have such small curvature radius, this case is ignored.

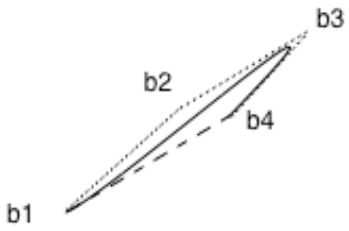
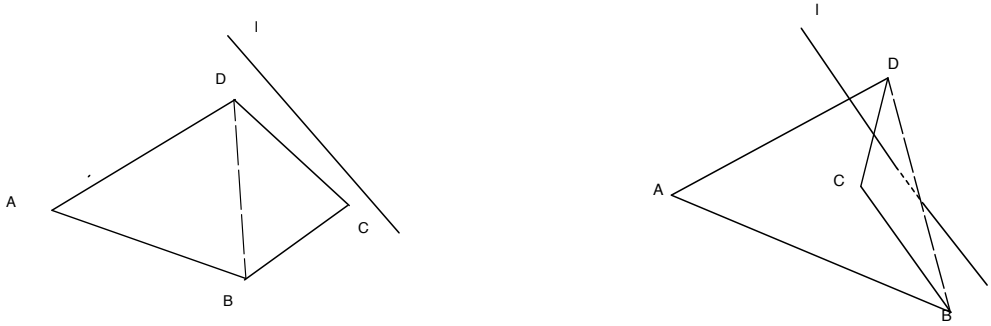


Figure 4.3 Fault termination of curve subdivision

4.1.5 Intersection of Line Segment and Quadrilateral

The quadrilateral is first separated into two triangles. Then the task is to find the intersections of lines and triangles. If the intersection point is outside ABD and BCD or inside both triangles (Figure 4.4), the line does not intersect the quadrilateral. However, the convex hull property is applied. If the corner control points are ABCD, the convex hull of (b) is ABD. Thus as long as the line intersects one of the triangles, the intersection point is accepted. To test whether the intersection point is inside the triangle, represent the intersection point in parameter u and w corresponding to the triangle vertices. If $0 \le u \le 1, 0 \le w \le 1 \; and \; u \ge w$, it is inside the triangle (Figure 4.5); otherwise, it is outside the triangle.



a) the line-plane intersection is outside triangle ABD and BCD

b) the line-plane intersection is inside triangle ABD and BCD

Figure 4.4 Line-plane intersection

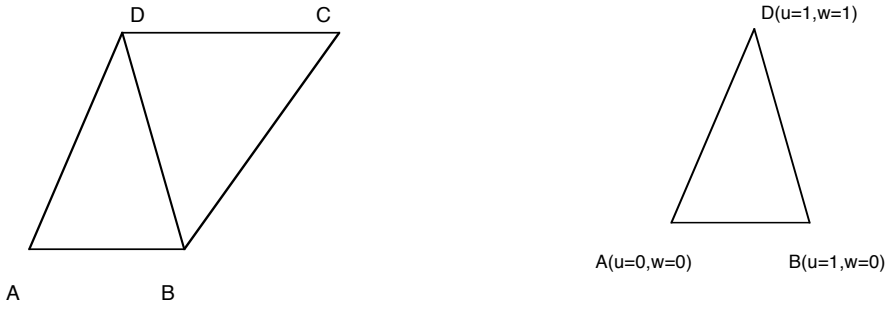


Figure 4.5 Mapping triangle vertices onto parametric space

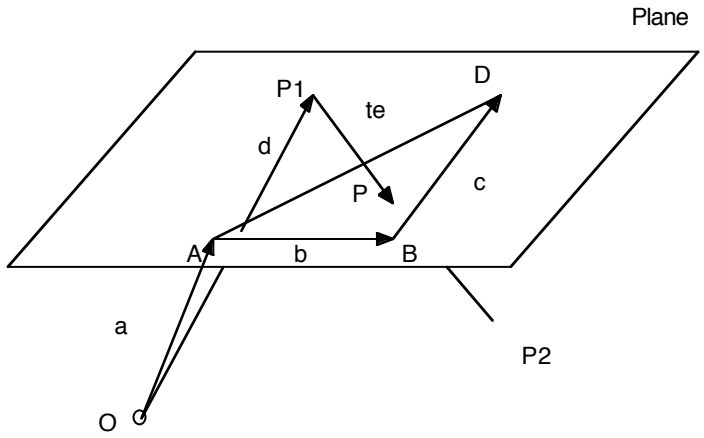


Figure 4.6 Line and triangle intersection

Let a point on line P_1P_2 be $\mathbf{d}+t\mathbf{e}$ and a point in triangle ABD be $\mathbf{a}+u\mathbf{b}+w\mathbf{c}$. If they intersect, then $\mathbf{a}+u\mathbf{b}+w\mathbf{c}=\mathbf{d}+t\mathbf{e}$. Apply some fundamental vector properties to isolate u,w and t:

$$t=\frac{(\mathbf{b}\times\mathbf{c})\cdot\mathbf{a}-(\mathbf{b}\times\mathbf{c})\cdot\mathbf{d}}{(\mathbf{b}\times\mathbf{c})\cdot\mathbf{e}} \tag{4-9}$$

$$u = \frac{(\mathbf{c} \times \mathbf{e}) \cdot \mathbf{d} - (\mathbf{c} \times \mathbf{e}) \cdot \mathbf{a}}{(\mathbf{c} \times \mathbf{e}) \cdot \mathbf{b}} \tag{4-10}$$

$$w = \frac{(\mathbf{b} \times \mathbf{e}) \cdot \mathbf{d} - (\mathbf{b} \times \mathbf{e}) \cdot \mathbf{a}}{(\mathbf{b} \times \mathbf{e}) \cdot \mathbf{c}} \tag{4-11}$$

If $0 \le t \le 1,\ 0 \le u \le 1, 0 \le w \le 1$ *and* $u \ge w$, we find the intersection point $P = \mathbf{d} + t\mathbf{e}$.

4.1.6 Mapping Intersection Points from Euclidean Space to Parametric Space

The points obtained from intersecting lines and quadrilaterals are in Euclidean space. They are sufficient for rendering. However, when sorting and connecting are required or the intersection points are used to trim NURBS surfaces, their corresponding values in parametric space are needed. A classical method is to first convert the curve into a power basis form and solve three polynomial equations for one unknown, or the parameter value to be determined. If the three equations have a common solution, then the point lies on the curve. This method has many disadvantages. The three equations for the curve with degree higher than four do not have a closed form solution. At the same time, we cannot expect the three solutions to be exactly equal. Moreover, the software implementation is rather involved. Thus the Newton's method is applied. This method finds the minimum distance between the point and the curve. The point is considered to be on the curve if the minimum distance is less than a specified tolerance. It is a gradient-based optimization method and the initial value is important. First, evaluate curve points at *n* equally spaced parameter values, choose the parameter value which yields the closest distance to the point as the initial value. The number *n* is generally chosen by heuristic methods. Let the intersection point be *P*. Form the dot product function

$$f(u) = C'(u) \bullet (C(u) - P) \tag{4-12}$$

The distance from *P* to *C(u)* is minimum when $f(u) = 0$. Denote by u_i the parameter obtained at the *i*th Newton iteration. Then

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)} = u_i - \frac{C'(u_i) \bullet (C(u_i) - P)}{C''(u_i) \bullet (C(u_i) - P) + \left|C'(u_i)\right|^2} \tag{4-13}$$

The criteria to check include

- Point coincidence: $|C(u_i) - P| \le \varepsilon_1$
- Zero cosine: $\dfrac{|C'(u_i) \bullet (C(u_i) - P)|}{|C'(u_i)||(C(u_i) - P)|} \le \varepsilon_2$

The point inversion for surfaces is analogous [Peig97]. Since the line-quadrilateral intersection is carried out based on Bezier form and the unnormalized parameters are used, it is more efficient to do the inversion using Bezier form.

In addition, this method requires computation of curve derivatives. Derivatives of nonrational B-spline curves is given by

$$C^{(k)}(u) = \sum_{i=0}^{n} N_{i,p}^{(k)}(u) P_i \qquad (4\text{-}14)$$

Only the basis functions are functions of u. The kth derivative of a basis function is

$$N_{i,p}^{(k)} = p\left(\frac{N_{i,p-1}^{(k-1)}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}(u)}{u_{i+p+1} - u_{i+1}}\right)$$

For nonrational Bezier curves, the kth derivative is given by

$$C^{(k)}(u) = \sum_{i=0}^{n} B_{i,n}^{(k)}(u) b_i \qquad (4\text{-}15)$$

And the kth derivative of nth degree Bernstein polynomials are given by

$$B_{i,p}^{(k)}(u) = \frac{p}{b-a}\left(B_{i,p-1}^{(k-1)}(u) - B_{i+1,p-1}^{(k-1)}(u)\right)$$

However, derivatives of rational functions are complicated, involving denominators to high power. The following method computes the derivatives of rational curves including rational Bezier and NURBS recursively. Let

$$C(u) = \frac{w(u)C(u)}{w(u)} = \frac{A(u)}{w(u)}$$

where $w(u) = \sum_{i=0}^{n} N_{i,p}(u) w_i$

then $A(u)$ is a nonrational curve. The first derivative of $C(u)$ can be computed by

$$C'(u) = \frac{A'(u) - w'(u)C(u)}{w(u)} \tag{4-16}$$

By differentiating $A(u)$ using Leibnitz' rule and rearrange the equation, we obtain

$$C^{(k)}(u) = \frac{A^{(k)}(u) - \sum_{i=1}^{k} \binom{k}{i} w^{(i)}(u) C^{(k-1)}(u)}{w(u)} \tag{4-17}$$

In a similar way, we obtain derivatives of a rational surface:

$$S^{(k,l)} = \frac{1}{w}\left(A^{(k,l)} - \sum_{i=1}^{k} \binom{k}{i} w^{(i,0)} S^{(k-i,l)} - \sum_{j=1}^{l} \binom{l}{j} w^{(0,j)} S^{(k,l-j)} - \sum_{i=1}^{k} \binom{k}{i} \sum_{j=1}^{l} \binom{l}{j} w^{(i,j)} S^{(k-i,l-j)} \right) \tag{4-18}$$

As we see the difficulties in computation of derivatives for rational curve/surface, an important property, curvature, involves the computation of the first and second derivative. A novel and direct method to compute the curvature of rational B-spline curves is provided in the Appendix A.

4.1.7 Connecting Intersection Points

The line-quadrilateral intersection produces a number of discrete intersection points lying on the isoparametric curves. To trace the intersection points, a marching scheme is used. First sort the intersection points on each isoparametric curve by the decreasing parameter *u*. Assume the *i*th isoparametric curve intersects surface S_2 with n_i number of points. Denote the *j*th intersection point from the *i*th isoparametric curves as $Q_{i,j}$. Start with the point of greatest *u* and *v*. March along the *v* direction to connect intersection points. Whenever a point is connected, remove it from the list, so we always connect the top point on every isoparametric curve. In each step there are three choices: $Q_{i,1}$ can be connected to $Q_{i-1,1}$, $Q_{i,2}$ or $Q_{i+1,1}$. To determine which direction to go, test the *u* value of $Q_{i-1,1}$, $Q_{i,2}$ and $Q_{i+1,1}$, and select the greater one. In Figure 4.7, $Q_{i,1}$ will be connected to $Q_{i+1,1}$. When all of them are empty or the curve reaches the surface boundary, the process stops. If there are still points left in the list, repeat this procedure to create another intersection curve.

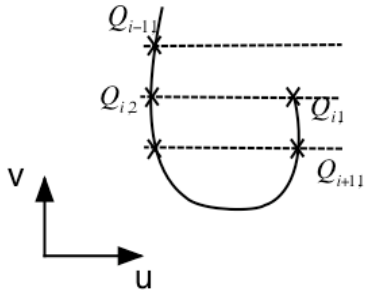


Figure 4.7 Connecting intersection points

4.1.8 Selection of Isoparametric Curves

When determining isoparametric curves from one surface, it is possible to miss some characteristic points if the number of isoparametric curves selected is small (Figure 4.2). In Figure 4.2(a), the isoparametric curves will not intersect at point A and B. However point A and B determine the chord length of the intersection airfoil. So it is important to have them computed. The isoparametric curves need not be selected uniformly.

To obtain the characteristic points, an isoparametric curve is inserted between the two adjacent isoparametric curves if the numbers of intersection points on them are different. This is a bisection approach. This process terminates when the parameter values of the two isoparametric curves are smaller than a tolerance.

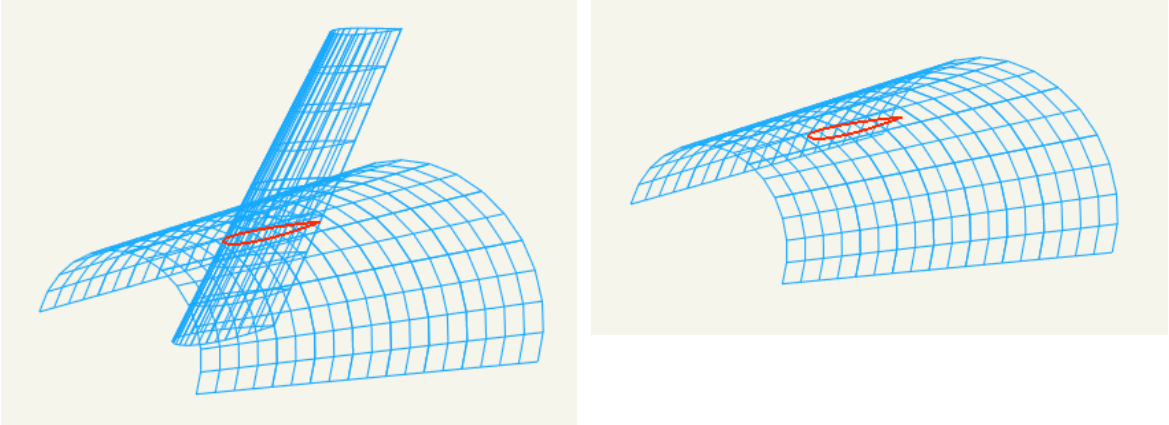
Which parametric variable to fix is also of some interest. We can fix either *u* or *v* of surface S_1 or surface S_2. Perhaps the most obvious criterion is fixing the parametric variable, which has the highest ratio $r = \frac{\text{number of lines intersected}}{\text{total number of lines}}$ when the isoparametric curves are distributed evenly. However, *r* is not a foreseeable parameter. In the execution, if a small number of intersection points are obtained, select the other parameter and try again. In the wing-fuselage intersection, the intersection curve in the wing's parametric space is an open curve, while in the fuselage's parametric space it is a closed curve. $r = 1$

if isoparametric curves are selected on this wing surface. This selection also avoids the missing characteristic points situation mentioned above.

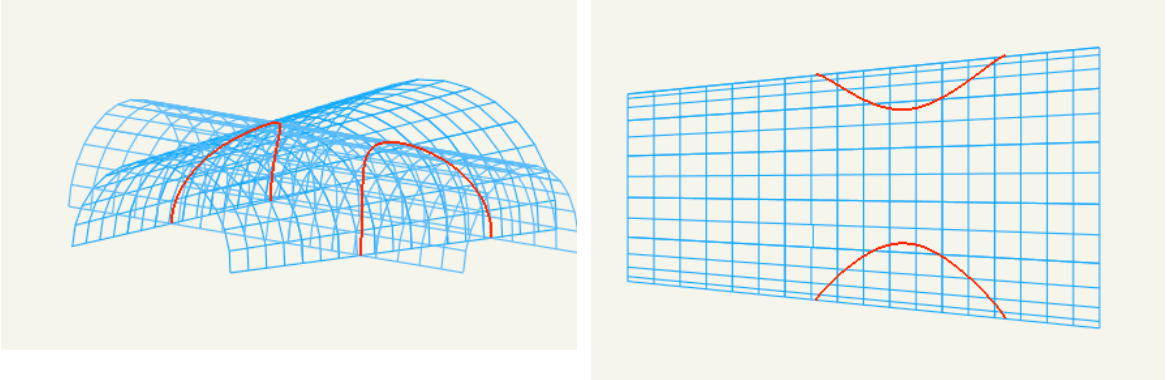
4.2 Examples and Discussions

In this algorithm, the performance can be greatly influenced by implementation choices. We caution that the timings provided in this section should only be taken as a general guide, since the process involves many operations. The efficiency of these operations affects the overall performance. Also implementation in different languages or compilation on a different computer would affect the process somewhat. The test was coded in C++ and performed using Mac OS X on a 400 Mhz PowerBook G4.

The major processes include transformation to Bezier surface patches and Bezier curves, bounding box intersection test, subdivision of Bezier surfaces and curves, line-quadrilateral intersection, finding parametric values of the intersection points, and connecting intersection points to curves. Depending on the surface characteristics, the amount of time spent on each process varies. Table 4.1 gives the number of calls and time for two examples of Figure 4.8. The most expensive operations are Bezier curve and surface subdivision. For a case in which a large number of transformed curves and surfaces do intersect, the time spent on subdivision is considerable.



(a)



(b)

Figure 4.8 Examples of intersections of B-spline surfaces

Table 4.1 Number of calls and time for obtaining the intersection curves for the examples of Figure 4.8

		Figure (a)		Figure (b)	
		Calls	Time (Seconds)	Calls	Time (Seconds)
Transform to Bezier	Curve	50	0.0035	648	0.045
	Surface	63	0.02	63	0.02
Bounding box separability test		6354	0.006	45788	0.045
Subdivision	Curve	564	0.011	0	0
	Surface	346	0.086	1241	0.31
Line-plane Intersection		61	0.0001	972	0.0019
Point inversion	Curve	25	0.005	82	0.016
	Surface	25	0.005	82	0.016
Connecting Points		1	0.0001	1	0.0001

The subdivision scheme can affect the efficiency dramatically. In static division, the division on the curve/surface continues until a termination condition is finally satisfied. In adaptive division, both the curve and the surface are divided until the curve/surface satisfies the termination condition, then only the surface/curve will be continuously divided. Adaptive division can detect the separation in the intersection test earlier and reduce the total number of subdivisions. Table 4.2 gives a comparison of static division and adaptive division for the examples of Figure 4.8. Since in Figure 4.8(b), the isoparametric curves are straight lines, adaptive or static division gives the same results.

Table 4.2 Number of calls of adaptive and static subdivision of Figure 4.8

Subdivision	Figure 4.8(a)	Figure 4.8(b)
	Number of Calls	Number of Calls
Adaptive subdivision		
Curve	564	0
Surface	346	1241
Static subdivision		

Curve	893	0
Surface	1119	1241

Termination criterion can be defined by different means. Using the absolute thickness of control hulls will set the bound for the error (the distance between the obtained intersection to the original surface) of each intersection point due to the convex hull property. A priori decision of the number of times a curve must be subdivided in order for a curve segment to approximate a line segment can also be made [Lass86]. Without setting the constraints on the error bound, thickness ratio and subdivision factor can also be used as a termination criterion.

Table 4.3 Number of intersection points vs. tolerance by subdividing both surfaces

Tolerance	Number of Intersection Points
1.0e-4	72
1.0e-5	121
1.0e-6	183
1.0e-7	352

In the algorithm of subdividing both surfaces given by Peng [Peng84], the number of intersection points increases very fast with a decreased tolerance (Table 4.3). However, by the algorithm given in this chapter, the number of intersection points is independent of the tolerance. If an increased number of intersection points is required, the intersection points calculated can be reused and only new points from different isoparametric curves need to be calculated. Moreover, when a very large number of intersection points is required, this algorithm is also suitable for parallel processing. The intersection points on each isoparametric curve can be calculated separately. Figure 4.9 shows the intersection curves created by Peng's algorithm. The algorithm creates 72 points with a tolerance of 1.0e-4 and 121 points with a tolerance of 1.0e-5. Due to the

difference of error bounds, the two curves do not match well. Figure 4.10 presents the intersection curves created by the algorithm given in this chapter. The tolerance is set to 1.0e-7. The first curve has 25 points and the second has 100 points. The points on the first curve match those on the second.

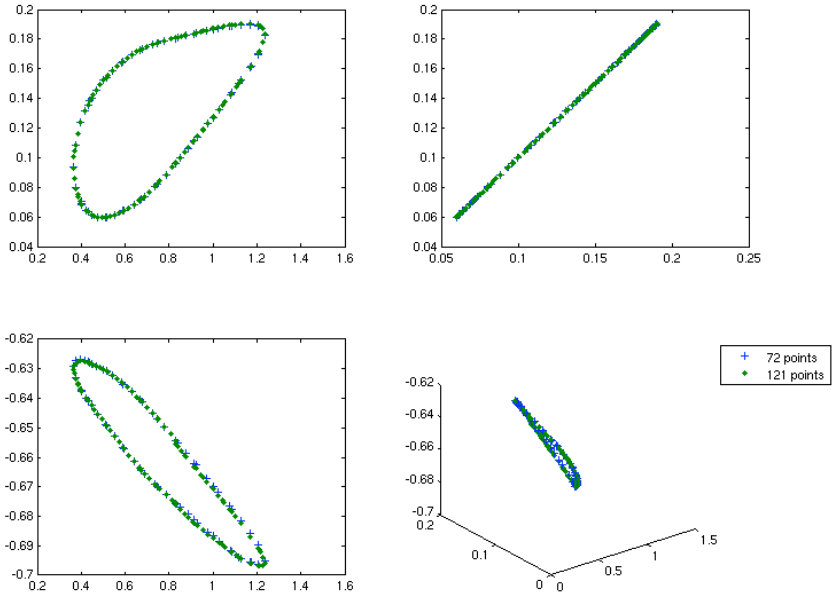


Figure 4.9 Intersection curves of Figure 4.9(a) by subdividing both surfaces The tolerance of the first curve is 1.0e-4 (72 points) and the second is1.0e-5 (121 points).

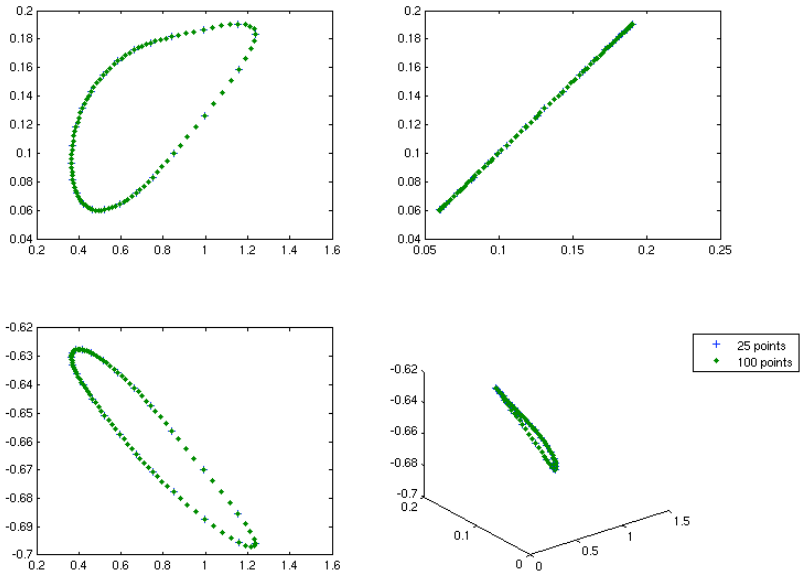


Figure 4.10 Intersection curves of figure 4.9(a) by algorithm given in this chapter with 25 points and 100 points (tolerance 1.0e-7)

In this implementation, each intersection curve contains a list of intersection points. A list allows efficient insertion, removal and sorting. Each intersection point contains three points corresponding to its three representations in Euclidean space and the surface parametric spaces. In the Euclidean space, the point has three coordinates, x, y and z. In parametric spaces, each point has two coordinates, u and v. Intersection of a wing and a control-hull based fuselage component is shown in Figure 4.11.

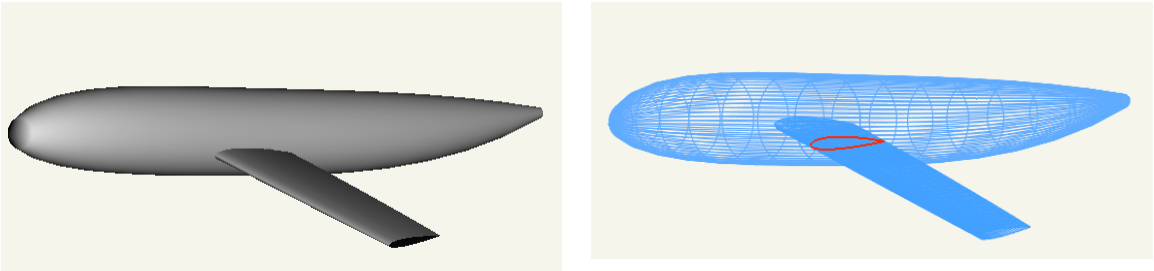


Figure 4.11 Intersection of a wing and a control-hull based fuselage component

4.3 Summary

Most of the existing algorithms focus on providing intersections suitable for rendering. In this chapter, an intersection algorithm suitable for geometric trimming of B-spline surfaces is presented. The number of intersection points is independent of the error bound. Intersection points on different isoparametric curves can be added to refine the intersection curve while the points calculated before can be reused. When a small number of intersection points with high accuracy is required, this algorithm is more efficient than previous methods due to the reduced computation involved. The major operations of this algorithm include determining parametric curves, transforming curve segments and surface patches to Bezier form, convex hull intersection test, line-quadrilateral intersection and connecting intersection points to curves. The algorithm is simple and easy to implement.

5 Geometric Trimming

This chapter presents a geometric trimming method for NURBS surfaces. We start with the comparison of geometric trimming and visual trimming, and then develop and illustrate the method using intersecting fuselage and wing surfaces. The intersection algorithm provided in the last chapter is applied to obtain the trimming curve. In the fuselage's parametric space the trimming curve is closed and the wanted portion is outside the trimming curve loop, while in wing's parametric space the trimming curve is open. We are concerned with surfaces trimmed by an open curve, which is monotone in at least one direction in the parametric space, and a closed curve, which can be subdivided into two such open curves.

5.1 Geometric Trimming vs. Visual Trimming

In most of the literature, the term "trimming" refers to visual trimming, which defines a trimmed surface by using the original surface and a trimming curve. When the trimmed surface is rendered, the original surface is tessellated so that it shows only the wanted portions. In contrast, geometric trimming creates new surface(s). Trimmed surfaces are represented in NURBS form. This is considered as mathematically "clean" since the unwanted portions are removed from the representation and the manipulation of the trimmed surface is the same as untrimmed surfaces. Having a consistent representation also helps data transference among different CAD systems. Figure 5.1 shows a visually trimmed wing surface and a geometrically trimmed wing surface. We notice the difference in their control hulls.

An exact trimming is precluded for both visual and geometric trimming due to the remarkably high degree of their intersections. Three representations of a trimming curve are required: one in the Euclidean space and one in each of the two surface parametric

spaces. The three representations are not exactly the same, which causes gaps and overlaps. For geometrically trimmed surface, errors are also created due to the re-sampling and re-interpolating process of creating new surfaces. The errors will be analyzed in the next chapter. In the remainder of this chapter, we will focus on the development of the algorithms. For the sake of brevity, "trimming" refers to "geometric trimming".

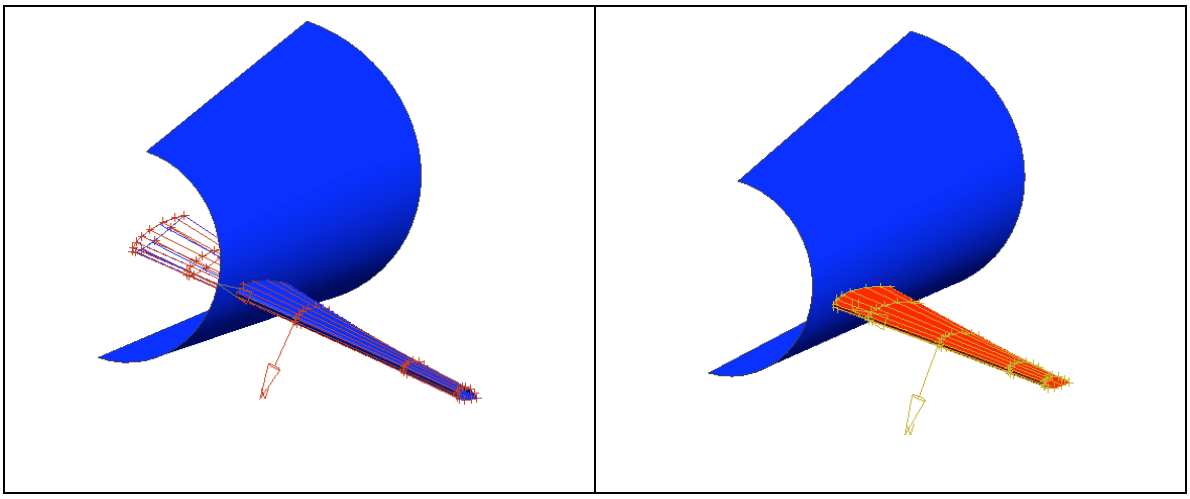
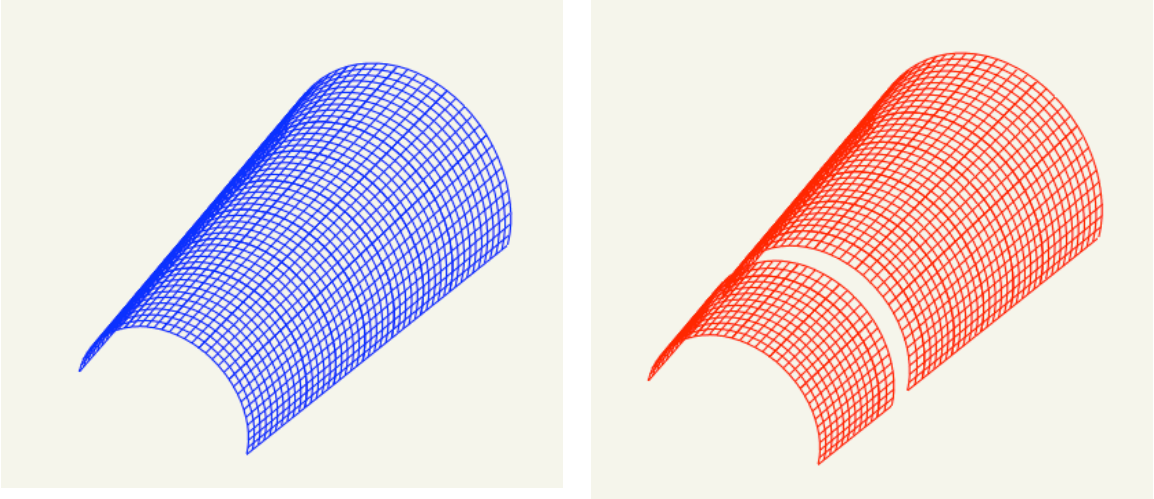


Figure 5.1 Visually trimmed and geometrically trimmed wing surfaces and their control hulls

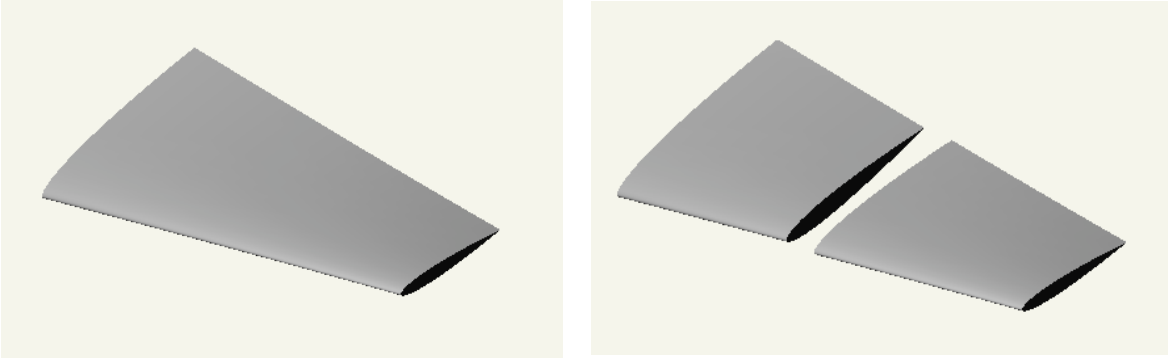
5.2 Trimming of Surfaces Along Isoparametric Curves

In general, exact trimming is precluded. However, in particular cases when the trimming curve is an isoparametric curve, the surface can be subdivided along the curve into two surfaces by knot insertion. Note that knot insertion is really just a change of vector space basis; the surface is not changed, either geometrically or parametrically. Thus the two new surfaces match the corresponding parts of the original surface exactly. The knot insertion has been described in section 4.1.1. Figure 5.2 and 5.3 shows a fuselage surface and a wing surface subdivided along isoparametric curves.



(a) Original surface (b) Subdivided surfaces

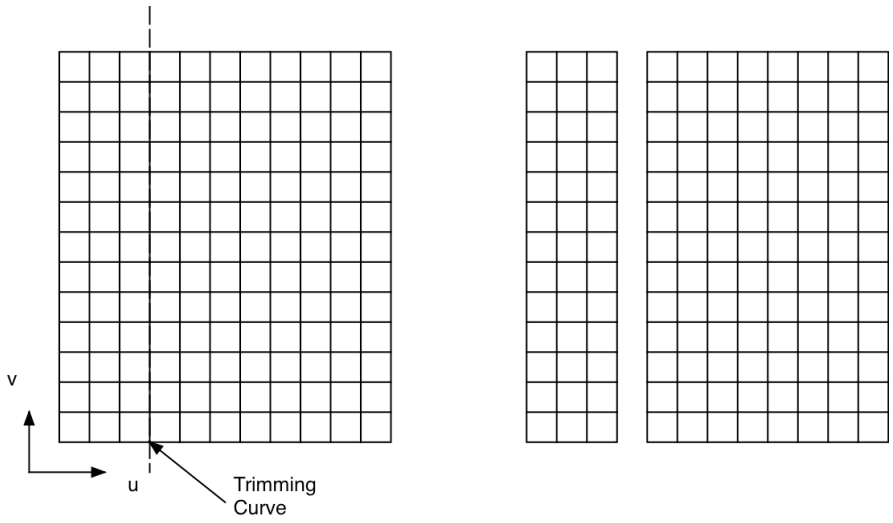
Figure 5.2 Subdividing a fuselage surface along an isoparametric curve



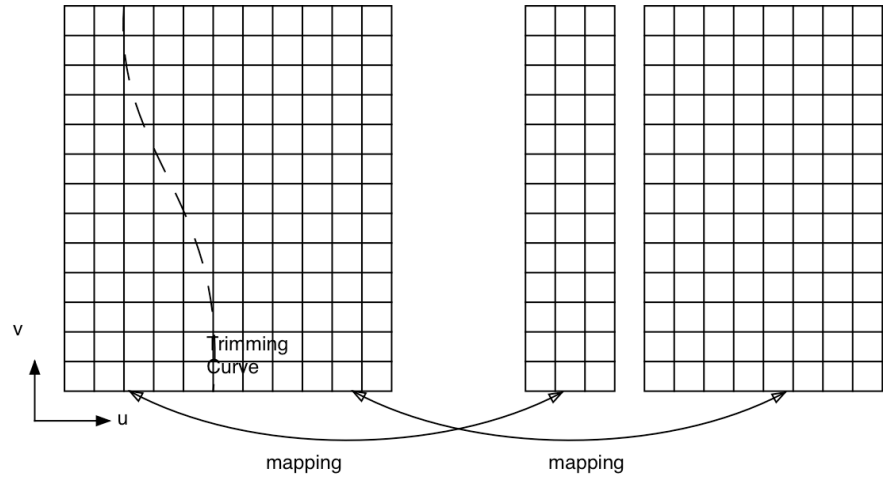
(a) Original surface (b) Subdivided surfaces

Figure 5.3 Subdividing a wing surface along an isoparametric curve

A NURBS surface in its parametric space is a rectangle. Trimming along an isoparametric curve divides the original domain into two rectangles. When a trimming curve is not an isoparametric curve (Figure 5.4), a mapping from the irregular shape to a rectangle is needed.



(a) a *u* isoparametric trimming curve



(b) a general trimming curve

Figure 5.4 Trimming curves in parametric spaces

5.3 Analysis of Trimming Curves

Trimming curves can be categorized into open curves and closed curves in parametric space. An open curve must have its two end points on surface boundaries. They may lie on the opposite sides, adjacent sides or the same side. We denote the *u*

isoparametric curve with the smallest u value $u_{\min}$ boundary; the u isoparametric curve with the greatest u value $u_{\max}$ boundary. The $v_{\min}$ boundary and $v_{\max}$ boundary are the v isoparametric curve with the smallest and greatest v values. An open trimming curve, whose two end points lie on the $u_{\min}$ and $u_{\max}$ boundary respectively, is called a u trimming curve, and an open trimming curve, whose two end points lie on the $v_{\min}$ and $v_{\max}$ boundaries respectively, is called a v trimming curve. We consider the case when the open trimming curve is a $u(v)$ trimming curve and the u (or v) of the trimming curve is monotone.

If the u (or v) of a trimming curve is monotone, any v (u) isoparametric curve on the surface intersects the trimming curve at no more than one point. For u or v trimming curves, the retained portion can be easily mapped from the old parametric space to a rectangle in the new parametric space. This will be discussed in section 5.4. If a u or v trimming curve is not monotone along the trimming direction, mapping from the wanted portion in the original domain to a rectangle might be done by applying potential theory or subdivision, which is discussed in section 5.7.

For trimming curves with two end points on the adjacent sides, the original surface should be first subdivided so that the trimming curve intersects the new surface on the opposite sides (Figure 5.5). So do the trimming curves whose end points lie on the same side of the surface.

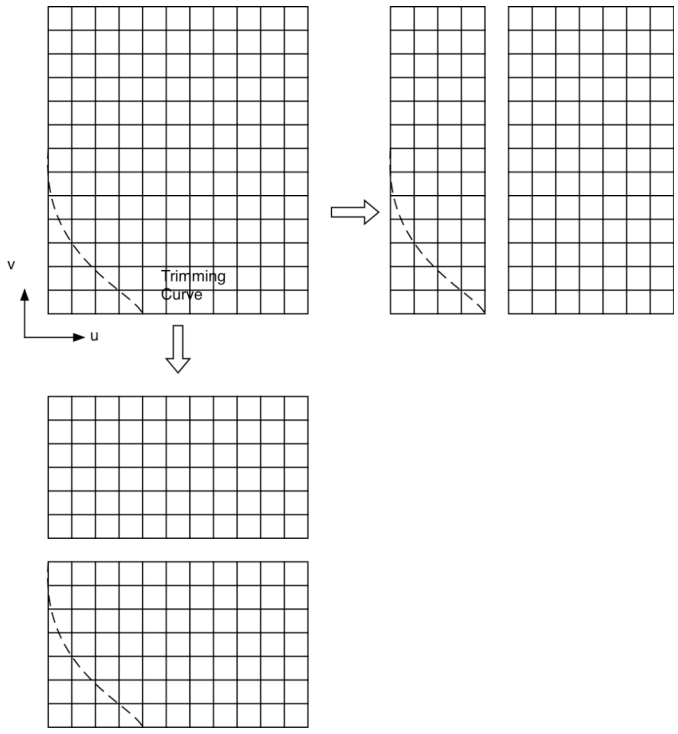
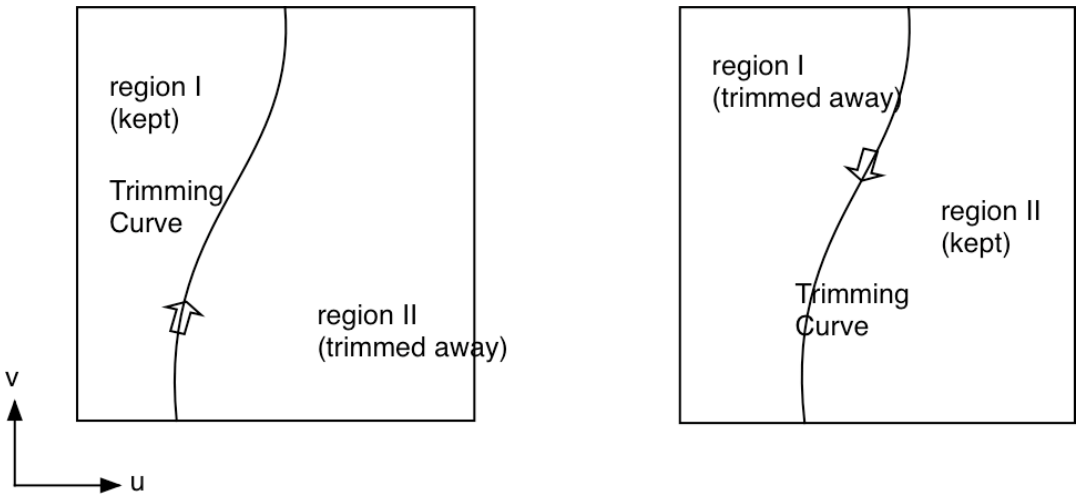


Figure 5.5 Subdivision of a surface and conversion of the curve to a u(v) trimming curve

If a surface is trimmed by a closed trimming curve, the portion outside the trimming curve loop has to be subdivided into patches to be mapped to rectangles. Trimming by a closed curve will be discussed in section 5.5.

5.4 Trimming by an Open Curve

The new surface is created by re-interpolating the surface points in the wanted portion of the original surface. A surface is divided into two regions by a *u* or *v* trimming curve. We set the rule so that the wanted portion is on the left side when walking along the trimming curve. In Figure 5.6(a), the *v* trimming curve starts with the smallest *v*, and region I will be kept, while in figure 5.6(b), the trimming curve starts with the greatest *v* and region II will be kept.



(a) (b)

Figure 5.6 Illustration of trimming regions

Due to the property of tensor-product surfaces, the number of interpolation points on each isoparametric curve must be the same. Let (u_{ltrim}, v_l) be the trimming curve point on the *l*th *v* isoparametric curve. The sampling points are thus selected uniformly in the wanted portion. The sampling points are $(u_k, v_l) = (u_{ltrim} \frac{k}{N^* - 1}, v_l)$ for region I in figure 5.6 and $(u_k, v_l) = \left(u_{ltrim} + (u_{\max} - u_{ltrim}) \frac{k}{N^* - 1}, v_l \right)$ for region II, where N^* is the number of points in the new parametric space on each *v* isoparametric curve. N^* can be any number as long as it satisfies the minimum number of points required by the surface degree and the interpolation method. Increasing the number of points forces a closer adherence to the original shape, however the more interpolation points used, the easier wiggles are created. Determine N^* by the greatest number of knots in the wanted portion. In the following, the re-interpolating process by using the interpolation to point data method is

explained. Other interpolation methods, such as interpolation with specified first derivatives, can be carried out similarly.

A new surface will be constructed by interpolating $N^* \times M^*$ surface points. Assume the surface is trimmed along *u* direction, let $M^* = M$. The superscript * is used for the new surface, which is defined by

$$Q^*_{k,l} = \sum_{i=0}^{N^*-1} \sum_{j=0}^{M^*-1} N_{i,p}(\bar{u}^*_k) N_{j,q}(\bar{v}^*_l) P^*_{i,j} \tag{5-1}$$

which can also be written as

$$Q^*_{k,l} = \sum_{i=0}^{N^*-1} N_{i,p}(\bar{u}^*_k) \left(\sum_{j=0}^{M^*-1} N_{j,q}(\bar{v}^*_l) P^*_{i,j} \right) = \sum_{i=0}^{N^*-1} N_{i,p}(\bar{u}^*_k) R^*_{i,l} \tag{5-2}$$

where $R^*_{i,l}$ are the control points of the isoparametric curves on the surface at a fixed $v^* = \bar{v}^*_l$. So the surface is interpolated by carrying out curve interpolation first in *u* and then in *v* direction, or vise versa due to the symmetric property. The curve interpolation in *u* direction is discussed. The interpolation in *v* direction is analogous. A common method is used to compute the $\bar{u}^{*l}_k$ for each *l*, and then to obtain each $\bar{u}^*_k$ by averaging across all $\bar{u}^{*l}_k, l = 0,...,M^* - 1$, that is

$$\bar{u}^*_k = \frac{1}{M^*} \sum_{l=0}^{M^*-1} \bar{u}^{*l}_k$$

$\bar{u}^{*l}_k$ are chosen by methods such as uniform, chord length and centripetal methods. To avoid a singular system matrix, the knot sequence is computed by using an averaging technique [Pieg97]:

$$u^*_0 = ... = u^*_p = 0, \; u^*_{N^*} = ... = u^*_{N^*+p}$$

$$u^*_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}^*_i$$

For a curve with *p*th degree, at least $p+1$ control points are required. If $\max(u_{ltrim}) \in [u_I, u_{I+1})$ and $\min(u_{ltrim}) \in [u_J, u_{J+1})$, we select $N^* = I + 1$ for region I and $N^* = N_{knot} - J - 1$ for region II. This provides a closer number of control points of the new surface to the wanted portion of the original surface.

The process can be stated as

- Determine N^* and compute the sampling grid in the parametric space by

$$(\bar{u}_k, \bar{v}_l) = (u_{ltrim} \frac{k}{N^* - 1}, v_l), \quad k = 0,...,N^* - 1, l = 0,...,M^* - 1$$

- Re-sampling (evaluating) points by

$$Q^*_{k,l} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} N_{i,p}(\bar{u}_k) N_{j,q}(\bar{v}_l) P_{i,j}$$

- Determine the new parameters $\bar{u}^*_k$, $\bar{v}^*_l$ corresponding to the interpolation points in the new parametric space and the knots sequence u^*_k, v^*_l.

- Solve the linear system $Q^*_{k,l} = \sum_{i=0}^{N^*-1} \sum_{j=0}^{M^*-1} N_{i,p}(\bar{u}^*_k) N_{j,q}(\bar{v}^*_l) P^*_{i,j}$ for the control points $P^*_{i,j}$.

5.5 Trimming by a Closed Curve

A closed curve divides a surface into two regions: one inside and the other outside the curve. In many applications such as wing and fuselage intersection, the region outside the trimming curve (region I) will be kept (Figure 5.7). It can be divided into patches by different schemes.

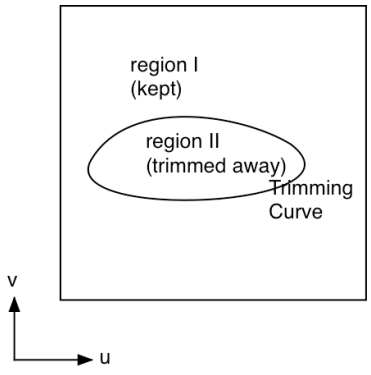


Figure 5.7 A closed trimming curve in parametric space

Applegarth et al [Appl89]. and Rojas [Roja94] divided the surface into four patches. The scheme is shown in Figure 5.8(a). Wang divided the surface into two patches [Wang01]. Each patch has five sides instead of four. When using a tensor-product surface to represent the trimmed surfaces, wiggles are unavoidable around the marked area a and b in the Figure 5.8(b).

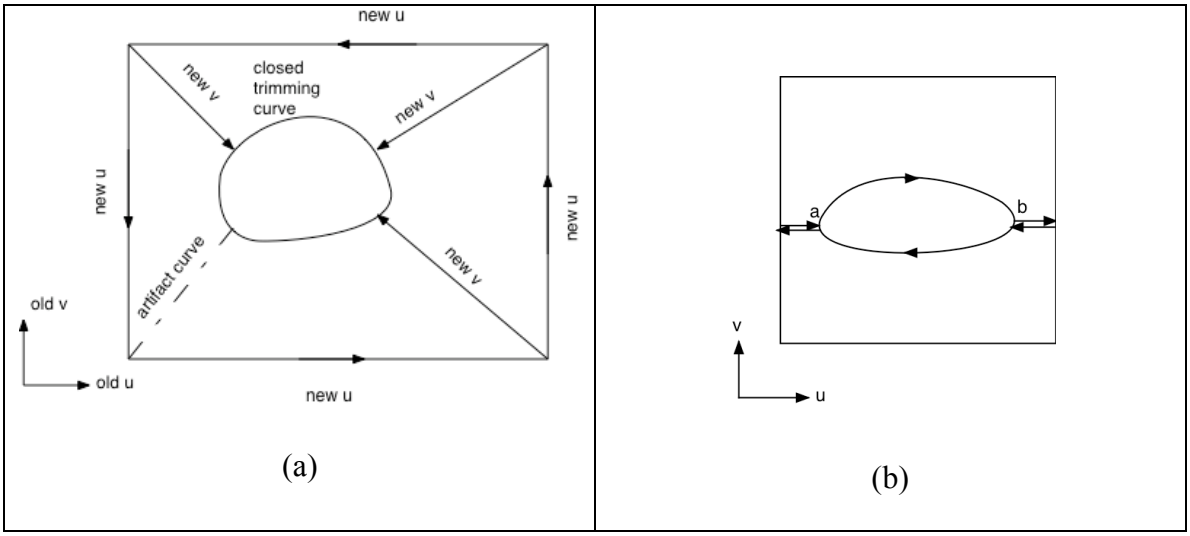


Figure 5.8 Subdivision of a surface trimmed by a closed curve

After subdivision, the continuities at the joint of the new surface can no longer be guaranteed for C2. In scheme (a), there are four break points on the trimming curve. The trimmed surfaces have poor accuracy. In scheme (b), due to the mapping from a five-sided to a four-sided shape, wiggles are unavoidable at location a and b in figure 5.8(b). Inserting more interpolation points around this region can reduce the magnitude of the wiggle. When these points are placed very close, an extreme situation is to insert multiple points at point a and b. If the multiplicity is equal to the surface order in this direction, the surface breaks into two sections along the isoparametric curve.

We notice that parts of the surface can be kept if the original surface is subdivided by isoparametric curves. We thus provide another scheme, which is illustrated in Figure 5.9. Patch A and D are trimmed by isoparametric curves so that they match the corresponding part of the original surface exactly. Patch B and C can be obtained by converting the closed trimming curve into two open curves and applying the method given in last section if the two open trimming curves satisfy the monotone conditions.

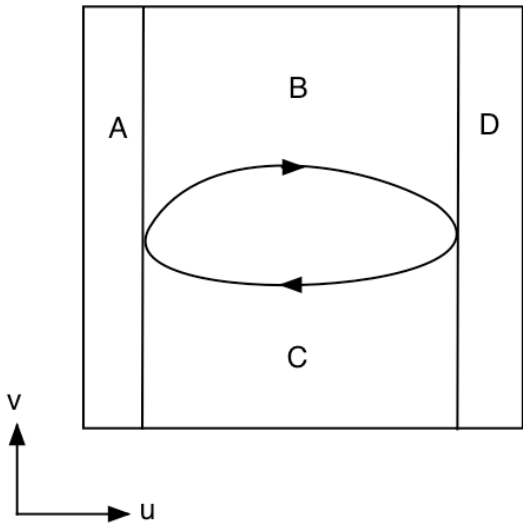


Figure 5.9 Subdivision of a trimmed surface by knot insertion and conversion of a closed trimming curve to two open trimming curves

If a closed trimming curve can be subdivided into two monotone u(or v) curves, the paths from $u_{\min}$ to $u_{\max}$ must be monotone. So we cut the closed curve at $u_{\min}$ and $u_{\max}$. First the $u_{\min}$ and $u_{\max}$ are found and then patch A is obtained by subdividing the original surface along the $u = u_{\min}$ isoparametric curve, and patch D is obtained by subdividing the original surface along $u = u_{\max}$. The remaining surface in the center is trimmed by two open trimming curves.

5.6 Implementation

Below is the pseudo code for the trimming method given in this chapter.

Nomenclature

umin—minimum *u* value of the trimming curve

umax—maximum *u* value of the trimming curve

usurfacemin—minimum *u* value of the surface to be trimmed

usurfacemax—maximum *u* value of the surface to be trimmed

vmin—minimum *v* value of the trimming curve

vmax—maximum *v* value of the trimming curve

vsurfacemin—minimum *v* value of the surface to be trimmed

vsurfacemax—maximum *v* value of the surface to be trimmed

- Trimming by a closed curve

Cut the trimming curve at umin and umax and convert the trimming curve to two open trimming curves;

if (the two trimming curves are monotonic in u direction) uflag=true;

Cut the trimming curve at vmin and vmax and convert the trimming curve to two open trimming curves;

if (the two trimming curves are monotonic in v direction) vflag=true;

if((uflag==true&&vflag==false) || (uflag==true&&vflag==true&& (umax-umin)/(usurfacemax – usurfacemin)> (vmax-vmin)/(vsurfacemax – vsurfacemin)) {

Subdivide the surface along the isoparametric curves u=umin and u=umax;

Trim the center patch by two open trimming curves;

}

else if((uflag==false&&vflag==true) || (uflag==true&&vflag==true&& (umax-umin)/(usurfacemax – usurfacemin) <(vmax-vmin)/(vsurfacemax – vsurfacemin)) {

Subdivide the surface along the isoparametric curves u=umin and u=umax;

Trim the center patch by two open trimming curves;

}

- Trimming by an open curve

if (the curve is not a u or v trimming curve) subdivide the surface and the trimming curve

for(each surface trimmed by a u or v trimming curve) {

if (the curve is monotonic in u direction) uflag=true;

if (the curve is monotonic in v direction) vflag=true;

if((uflag==true&&vflag==false) || (uflag==true&&vflag==true&& (umax-umin)/(usurfacemax – usurfacemin)> (vmax-vmin)/(vsurfacemax – vsurfacemin)) {

Rearrange the curve so that it starts with the smallest u;

if (the left side is wanted){

Sampling N^*=I+1; surface points on selected isoparametric between usurfacemin and utrim;

}else{

Sampling N^*= Nknot–J -1; surface points on between utrim and usurfacemax;

}

Create a new B-spline surface by interpolating the surface points;

} else if((uflag==false&&vflag==true) || (uflag==true&&vflag==true&& (umax-umin)/(usurfacemax – usurfacemin) <(vmax-vmin)/(vsurfacemax – vsurfacemin)){

Rearrange the curve so that it starts with the smallest v;

if (the left side is wanted){

Sampling N^*=I+1 surface between vsurfacemin and vtrim points on each selected isoparametric curve;

}else{

Sampling N^*=Nknot–J -1 surface points between vtrim and vsurfacemax on each selected isoparametric curve;

}

Create a new B-spline surface by interpolating the surface points;

}
}

5.7 Examples and Discussion

Figure 5.10 and 5.11 give two examples of trimming fuselage and wing surfaces. In Figure 5.10, the fuselage is represented by a cylinder and the wing profile is a symmetric airfoil. The dihedral and twist are zero. While in Figure 5.11, the fuselage has elliptical front and rear cross sections. The wing has a 5 degree dihedral and 8 degree of twist. The airfoil is NACA 4-digit 2412. Figure 5.12 is an example of a trimmed control hull based fuselage component.

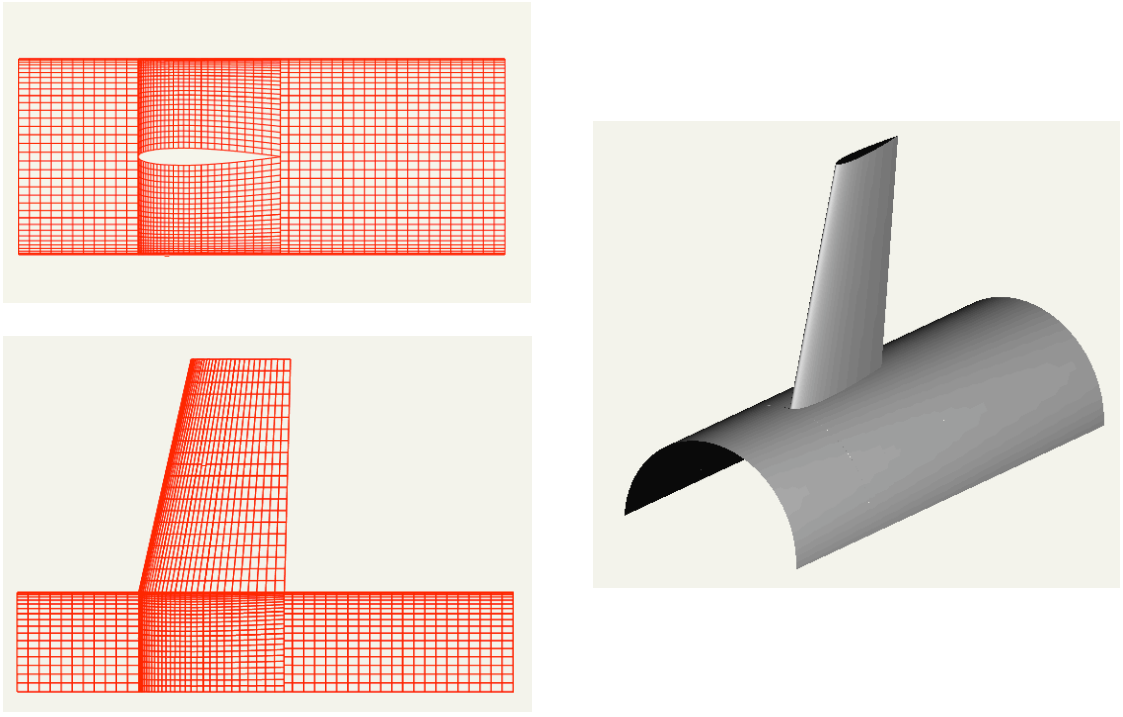


Figure 5.10 Trimming of fuselage and wing (with symmetric airfoil) surfaces

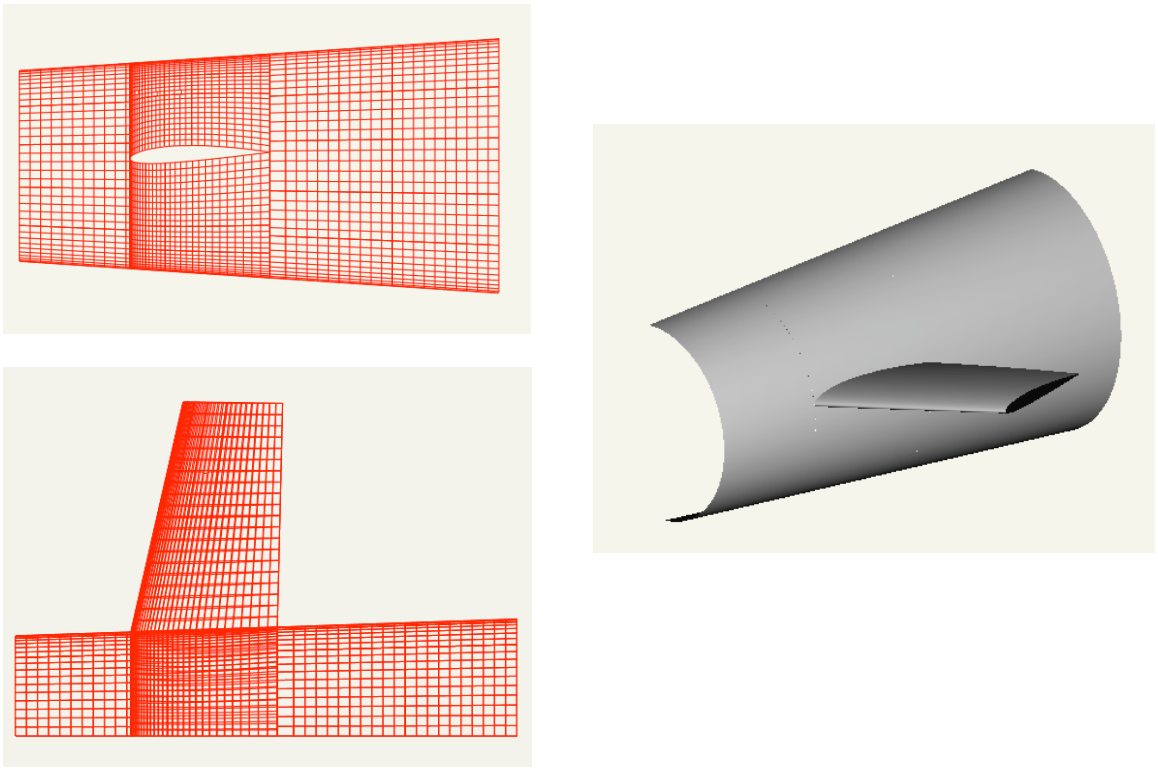


Figure 5.11 Trimming of fuselage and wing (with unsymmetric airfoil) surfaces

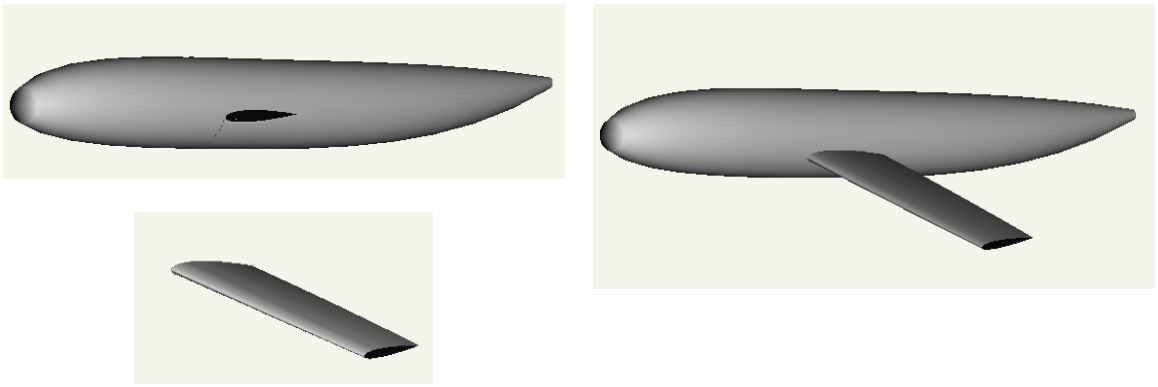


Figure 5.12 Trimming of a wing and a control-hull based fuselage component

We have described the selection of sampling points on each isoparametric curve. The points are distributed evenly in the parametric space. The determination of isoparametric curves also affects the results. If the original surface is created by

interpolation and we have the knowledge of the original interpolation, the knowledge can be applied to trimming sufficiently. Using the parameters of the original interpolation points for sampling the new surface points and computing parameters by the same parameterization method provide high accuracy. The scheme is shown in Figure 5.13, The *v*'s of the original surface are used for sampling trimmed surface points. The number of control points in the v direction of the trimmed surface is the same as the original one. However, in many cases, we do not have the knowledge of how the original surface is created. There are many ways to select isoparametric curves. They can be equidistance, or on the knots. Their influence on trimming errors will be compared in the next chapter.

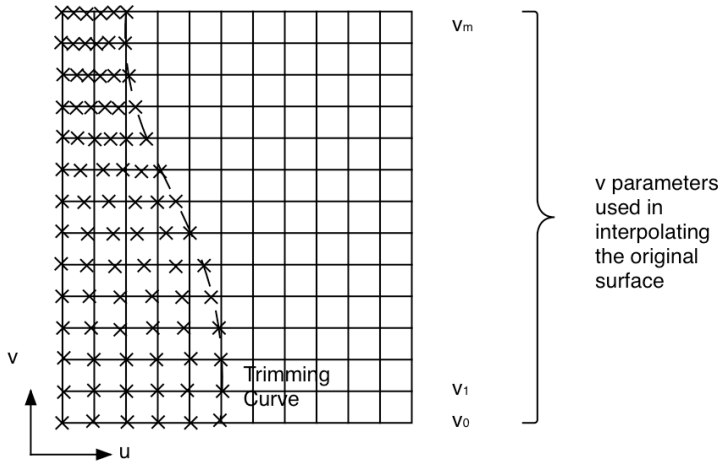


Figure 5.13 Selection of parameters based on the original interpolation points

Intersection curves are found by applying the algorithm provided in last chapter. This method has various advantages for geometric trimming: First, only the intersection points needed in the trimming process are computed. Isoparametric curves, on which the sampling points lie, are determined and intersected with the other surface. Usually, only a small number of isoparametric curves are selected and finding the intersection points is efficient. Second, the accuracy of these points is independent of the number of

isoparametric curves. Third, the interpolation points are connected. We can easily determine the trimming curve types and do trimming curve conversion.

In our method, the trimming curve must satisfy conditions mentioned in section 5.3 and 5.5. If neither the *u* nor *v* of the trimming curve is monotonic, an isoparametric curve might intersect the trimming curve with more than one point. A mapping from the trimmed portion to a rectangular domain cannot be done by mapping isoparametric curves. A possible solution might be applying potential theory. Think of the new surface points on the equipotential lines. Dirichlet boundary conditions are set from the trimming curve. Solve Laplace's equation :

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial u^2} + \frac{\partial^2\Phi}{\partial v^2} = 0$$

and obtain the equipotential lines. Then sample the points on the equipotential lines and construct the new surface by interpolating these points. Further subdividing the trimming curve and surface might also be a solution.

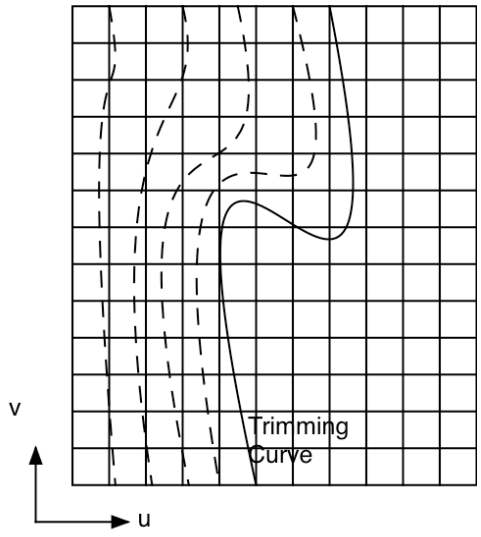


Figure 5.14 Finding interpolation points by solving Laplace's equation

5.8 Summary

In this chapter, a method was presented for geometrical trimming of fuselage and wing surfaces. Certain monotonic conditions for this method were stated. For more generalized problems, possible solutions for future research have been discussed. In this method, the surface trimmed by a closed trimming curve is divided into four patches. Two of the patches are constructed by subdivision along isoparametric curves and thus match the original surface exactly. In next chapter, the errors will be analyzed and the details discussed further.

6 Error Evaluation and Curvature Calculation

Since exact trimming is precluded, a measure of the difference between the original surface and the trimmed one is needed. The difference is referred to as "deviation" or "error". In this chapter, several means of error definition are provided and the distances between corresponding points on the two surfaces are discussed. Not only is the difference of the surface representation considered, but its geometric property, curvature. Curvature is directly linked to many physical meanings, such as strain energy in beam theory and lift and drag in aerodynamics.

6.1 Trimming Errors

Surfaces are treated as discretized points so that the computation of the difference between two surfaces can be converted to the difference between two nets of surface points. The distance between two surface points is the distance in Euclidean space:

$$\left|\mathbf{x}_i - \mathbf{x}_i^*\right| = \sqrt{(x_i - x_i^*)^2 + (y_i - y_i^*)^2 + (z_i - z_i^*)^2} \qquad (6\text{-}1)$$

Let the net of surface points on the original surface be $\mathbf{q}_{i,j}\ i = 0,..,m-1,\ j = 0,...,n-1$ and the trimmed surface $\mathbf{q}_{i,j}^*\ i = 0,..,m-1,\ j = 0,...,n-1$. If we consider them as vectors, and each has $r = m \times n$ points:

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_r \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} \mathbf{x}_0^* \\ \mathbf{x}_1^* \\ \cdots \\ \mathbf{x}_r^* \end{bmatrix}$$

where $\mathbf{q}_{i,j} = \mathbf{x}_{i*m+j}$ and $\mathbf{q}_{i,j}^* = \mathbf{x}_{i*m+j}^*$, the difference is $\Delta\mathbf{X} = \mathbf{X}_1 - \mathbf{X}_2$.

The most common norms for a vector are the L1-norm, L2-norm and L(infinity)-norm, defined by

$$\|\Delta \mathbf{X}\|_1 = \sum_{i=0}^{r-1} |\mathbf{x}_i - \mathbf{x}_i^*| \tag{6-2}$$

$$\|\Delta \mathbf{X}\|_2 = \sqrt{\sum_{i=0}^{r-1} |\mathbf{x}_i - \mathbf{x}_i^*|^2} \tag{6-3}$$

$$\|\Delta \mathbf{X}\|_\infty = \max_i |\mathbf{x}_i| \tag{6-4}$$

When the number of surface points changes, so does the vector size. If we take the discretization into account and change the summations, the values, which are independent of the number of points, can be obtained. We thus use the following equations to evaluate the distance between two surfaces

$$d_1 = \frac{\|\Delta \mathbf{X}\|_1}{r}, \; d_2 = \frac{\|\Delta \mathbf{X}\|_2}{r} \text{ and } d_\infty = \|\Delta \mathbf{X}\|_\infty$$

Finding the corresponding points is performed in Euclidean space since the original and trimmed surfaces are in different parametric spaces. Furthermore we compare the trimmed surface with only part of the original surface. So it is difficult to apply the so-called uniform metric, in which the parameter ranges are divided into $N \times M$ subintervals and one simply compares the surfaces at the resulting points. Thus we first sample the surface points on the retained part of the original surface in the parametric space and then project points to the trimmed surface to find its associate point in Euclidean space. The $\mathbf{q}_{i,j}$ are sampled by uniformly dividing the parameter ranges. To associate $\mathbf{q}_{i,j}$ with its image $\mathbf{q}_{i,j}^*$ on the trimmed surface, $\mathbf{q}_{i,j}$ is projected onto the trimmed surface. It can be projected vertically so that the two points have same x and y coordinates, or it can be projected by finding the closest point on the original surface (Figure 6.1). The vertical projection is orientation dependent. Rotating the two surfaces will give different corresponding points. So the projection by finding the closest point on the surface. The Newton's method, which is a classical method, is applied. It has been described in section 4.1.6 and will not be repeated in this section. The exceptions are the

boundaries, where the points are projected to the boundary curves of the trimmed surface instead of the surface. The point projection does not prevent projecting a point on the upper surface of the original wing surface to the lower surface of the trimmed surface at the airfoil trailing edge due to the small distance between the upper and lower surfaces, which is shown in Figure 6.2. In this case, constraints are added to force the upper surface of the original surface at the trailing edge to project only onto the upper surface of the trimmed surface.

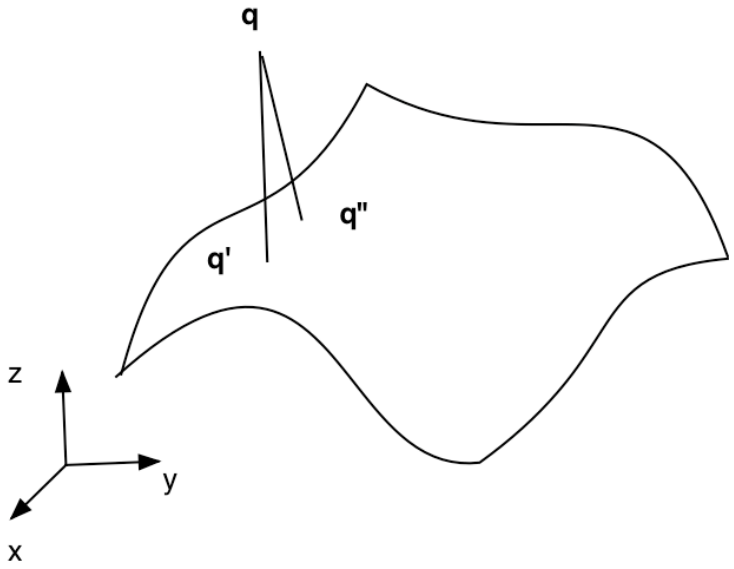


Figure 6.1 Point projection

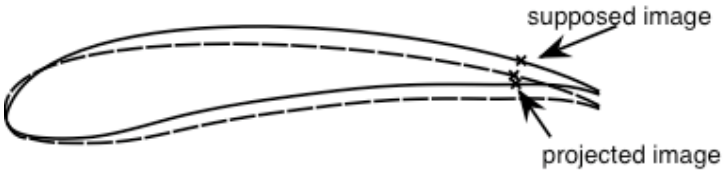


Figure 6.2 Projecting a point on the upper surface of the original surface without constraints

6.2 Evaluating Trimming Errors

Note that a large number of points are required to evaluate the errors. Efficiency thus becomes an important issue in any application with intensive computation of errors. The computation of the Euclidean distance between two points involves computing square roots, which is a very expensive operation.

Let us rewrite the L1-norm and L2-norm as

$$\|\Delta \mathbf{X}\|_1 = \sum_{i=0}^{r-1} \sqrt{(x_i - x_i^*)^2 + (y_i - y_i^*)^2 + (z_i - z_i^*)^2}$$

$$\|\Delta \mathbf{X}\|_2 = \sqrt{\sum_{i=0}^{r-1} \left[(x_i - x_i^*)^2 + (y_i - y_i^*)^2 + (z_i - z_i^*)^2 \right]}$$

From the above, we notice that L1-norm requires computation of square root r times while L2-norm only requires once. So computation of L2-norm is more efficient. In the next chapter, when the errors are computed in the objective functions of the optimization process, they are computed intensively. The L2-norm is a better choice considering the efficiency.

6.3 Curvatures

Curvatures are local surface properties and intrinsic to the surface. An infinite number of curves can be obtained on a surface. Let $u(t)$ be one curve on the surface. The first and second fundamental form in classical differential geometry gives

$$ds^2 = \mathbf{x}_u \mathbf{x}_u du^2 + 2\mathbf{x}_u \mathbf{x}_v du dv + \mathbf{x}_v \mathbf{x}_v dv^2 \tag{6-5}$$

$$\kappa \cos\phi ds^2 = \mathbf{n}\mathbf{x}_{uu} du^2 + 2\mathbf{n}\mathbf{x}_{uv} du dv + \mathbf{n}\mathbf{x}_{vv} dv^2 \tag{6-6}$$

where ϕ is the angle between the main normal of the curve and the **n** is the surface normal at the point **x**, s is the chord length and κ is the curvature. Using the abbreviations

$E = \mathbf{x}_u \mathbf{x}_u,\ F = \mathbf{x}_u \mathbf{x}_v,\ G = \mathbf{x}_v \mathbf{x}_v$

$L = \mathbf{n}\mathbf{x}_{uu},\ M = \mathbf{n}\mathbf{x}_{uv},\ N = \mathbf{n}\mathbf{x}_{vv}$

the equations can be written as

$$ds^2 = Edu^2 + 2Fdudv + Gdu^2 \tag{6-7}$$

$$\kappa \cos\phi ds^2 = Ldu^2 + 2Mdudv + Ndv^2 \tag{6-8}$$

The right-hand side of (6-8) does not contain terms involving ϕ. For $\phi = 0$,

$$\kappa = \frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2} \tag{6-9}$$

The curvature of such a curve is the normal curvature of the surface at **x** in the direction of **t** (defined by du/dv). This is still a function of **t**. By setting $\lambda = dv/du$, the normal curvature is

$$\kappa(\lambda) = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2} \tag{6-10}$$

For a *u* isoparametric curve, $\lambda -> \infty$, thus $\kappa_u = \frac{N}{G}$. Similarly, for a *v* isoparametric curve, $\lambda = 0$ and $\kappa_v = \frac{L}{E}$. The extreme values of $\kappa(\lambda), \kappa_1$ and κ_2 occur at the roots of

$$\det\begin{bmatrix} \kappa E - L & \kappa F - M \\ \kappa F - M & \kappa G - N \end{bmatrix} = 0$$

The κ_1 and κ_2 are principal curvatures of the surface at **x**. Also the above equation yields

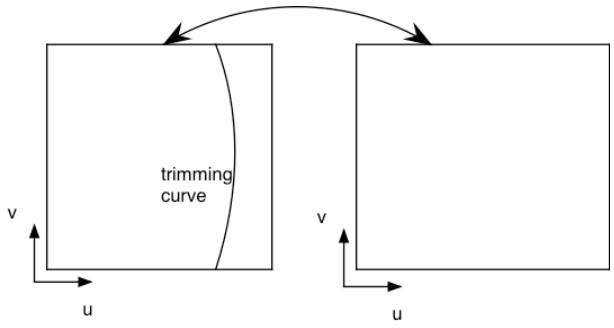
$$\kappa_1 \kappa_2 = \frac{LN - M^2}{EG - F^2}$$

and $\kappa_1 + \kappa_2 = \frac{NE - 2MF + LG}{EG - F^2}$.

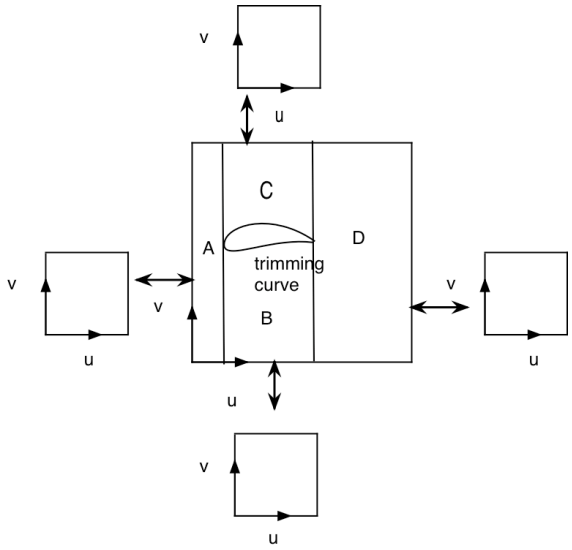
The term $K = \kappa_1 \kappa_2$ is the Gaussian curvature and $H = \frac{1}{2}(\kappa_1 + \kappa_2)$ is the mean curvature. These properties are compared for the original and the trimmed surfaces in the following sections.

6.4 Wing and Fuselage Surface Trimming Errors

The method for geometric trimming of fuselage and wing surfaces was developed in the last chapter. Several factors, which affect the resulting trimmed surfaces, were mentioned. The error evaluation in this chapter provides us comparison metrics. Consider the example given in section 5.7 (Figure 5.11). Figure 6.3 illustrates the mapping of the trimmed and original surfaces in their parametric spaces. The distances of the corresponding points of fuselage region A and D are then analyzed and plotted in Figure 6.4. These two regions are created by subdivision using knot insertion of the original surface and theoretically the errors are zero. We use them to observe the errors caused by the floating-point error in the evaluation and the influence of the tolerance in point projection. In all figures in this chapter, the parameters are in the normalized parametric space, which ranges from 0 to 1. Table 6.1 shows that these errors are small enough to be considered as trivial. So by tuning the tolerances, the error caused by the point projection operation can be ignored and the evaluation is valid. Three tolerances are used in point projection: ε_1 for point distance, ε_2 for cosine and ε_3 for parameter boundary control. In the above figure, ε_1 and ε_2 are set to 1.0e-16. ε_3 is the tolerance in the parametric space, the relation of ε_3 and ε_1 depends on the surface characteristics. In the above examples, the ε_3 is set to 1.0e-20.



(a) Wing surface

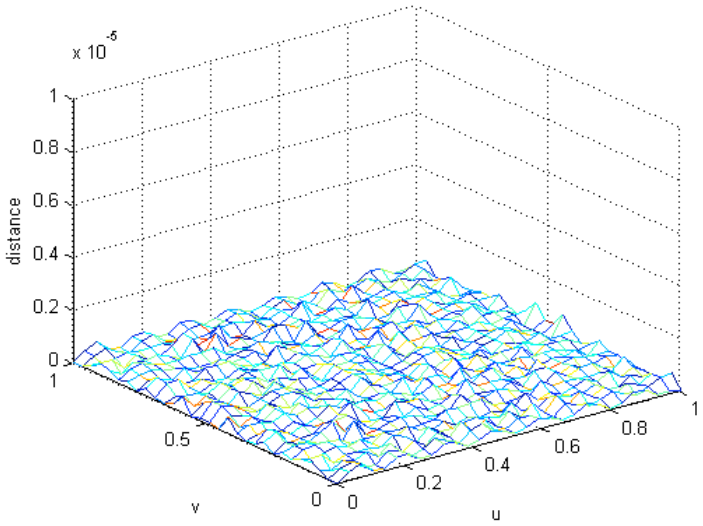


(b) Fuselage surface

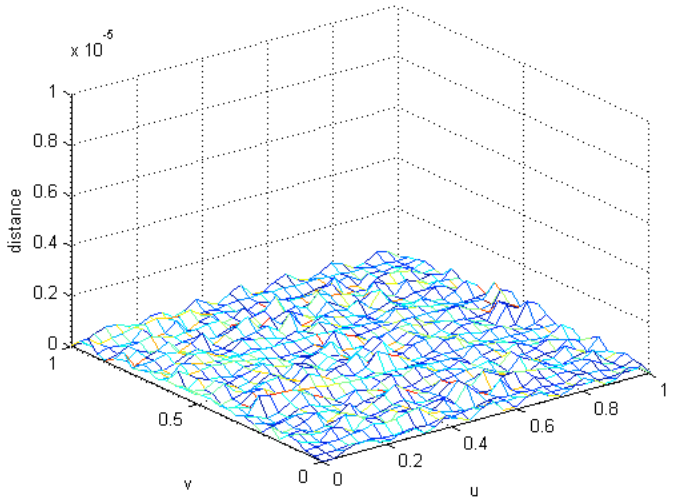
Figure 6.3 Mapping of the trimmed surfaces to the original surfaces in parametric spaces

Table 6.1 Errors of region A and D of a trimmed fuselage surface

	Region A	Region D
d_1	2.4984e-7	2.4806e-7
d_2	9.9750e-9	1.0165e-8
d_∞	8.0885e-7	8.9122e-7



(a) region A



(b) region D

Figure 6.4 Distance plot of trimmed fuselage surface region A and D

The trimmed wing surface and fuselage surface B and C have greater distances around the trimming curve, since the truncation at the trimming curve changes the end conditions. Figure 6.5 and 6.6 show the distance plots of the trimmed wing and fuselage

surfaces. Surface B has the trimming curve on the $v = 1.0$ and surface C has the trimming curve on $v = 0$. At the interpolation points, the distances are trivial.

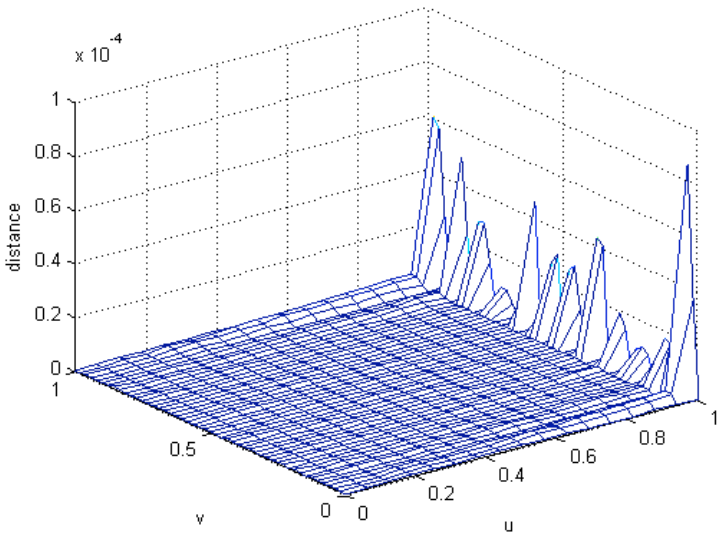
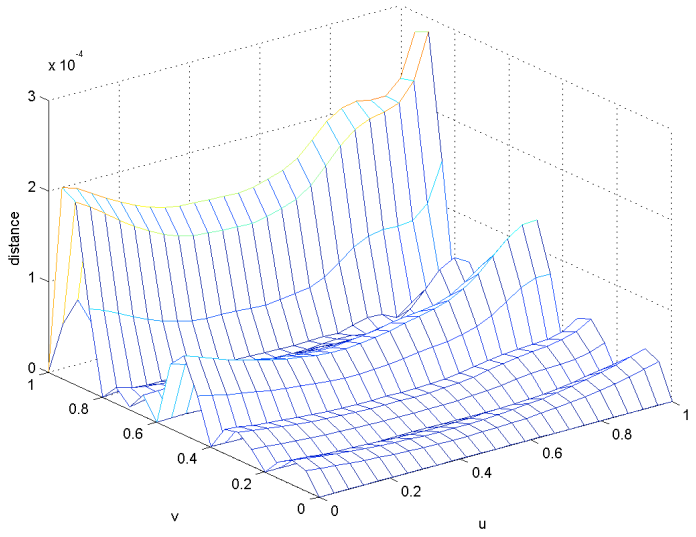
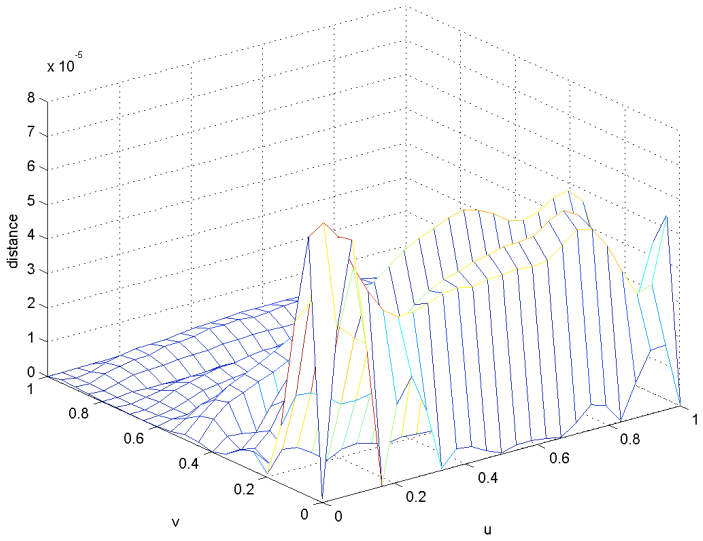


Figure 6.5 Distance plot of the trimmed wing surface



(a) region B



(b) region C

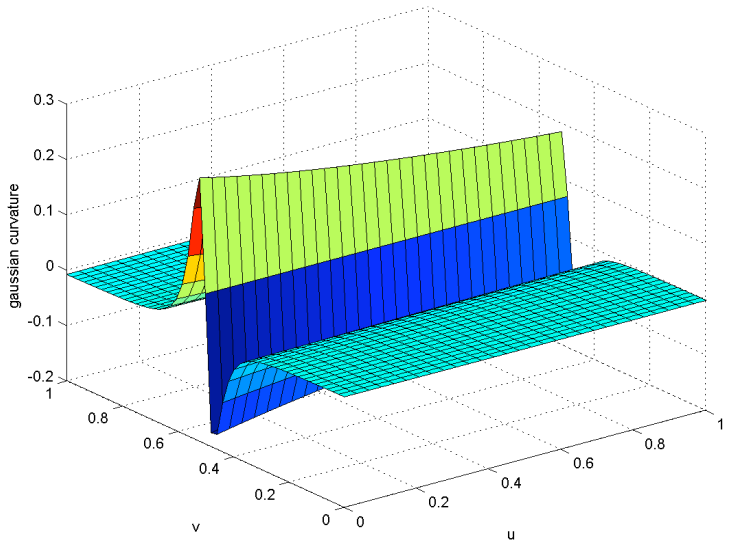
Figure 6.6 Distance plot of the trimmed fuselage surface

Table 6.2 Errors of region B and C of the trimmed fuselage surface and the wing surface

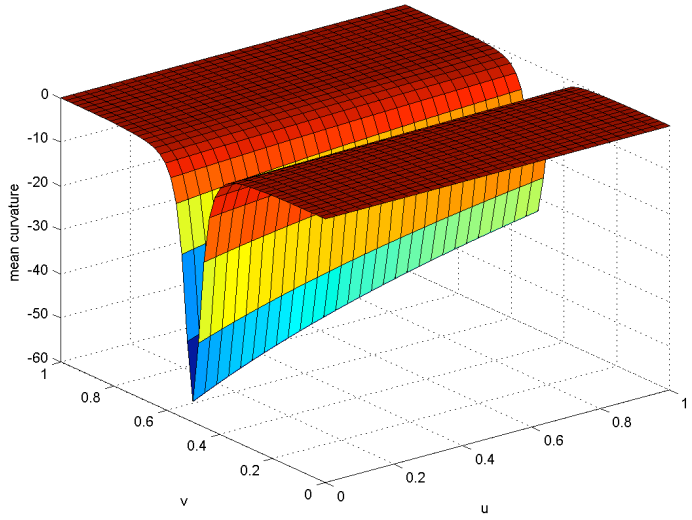
	Region B	Region C	Wing
d_1	3.7176e-5	1.5781e-5	1.4594e-6
d_2	2.8038e-6	1.0351e-6	2.168e-7
d_∞	2.8411e-4	7.8506e-5	8.4740e-5

Studies show the manufacturing tolerance for a normalized airfoil of chord 1 is $2*10^{-4}$[Trep01]. In this example, both the original and the trimmed surfaces are interpolated using uniform parameterization, and errors of the trimmed wing surface are below this tolerance. Figure 6.7 and 6.8 are the curvature plots of the original wing and fuselage surfaces. Figure 6.9 shows the curvatures of the trimmed wing surfaces. Figure 6.10 and Figure 6.11 show the curvatures of the trimmed fuselage region B and C. The curvatures of the trimmed wing are close to those of the original surface, while the

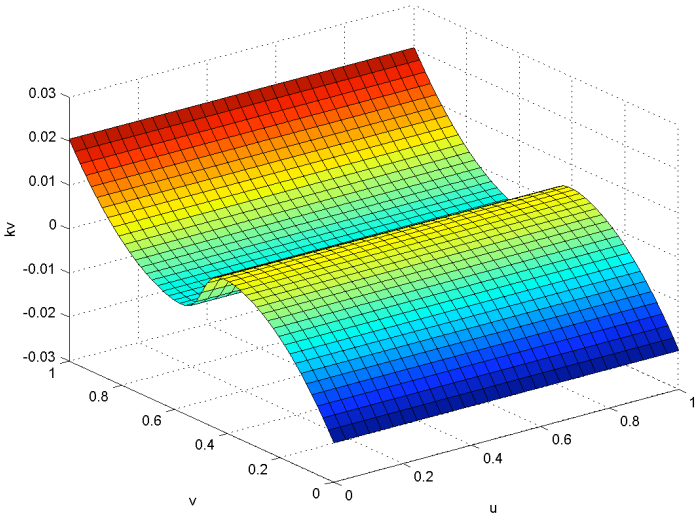
trimmed fuselage region B and C have greater curvature changes. To see the details of the flat portion in Figure 6.11, only part of the surface is displayed in Figure 6.12. The smaller *u* value side corresponds to the leading edge of the intersecting airfoil and the greater *u* to the trailing edge.



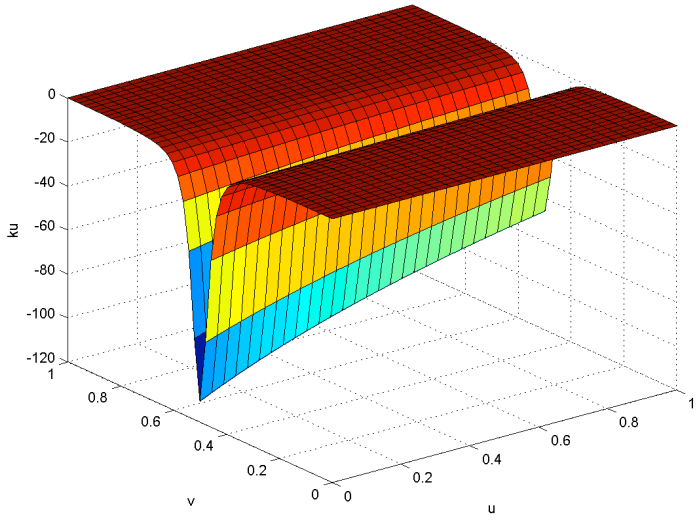
(a) Gaussian curvature plot



(b) mean curvature plot

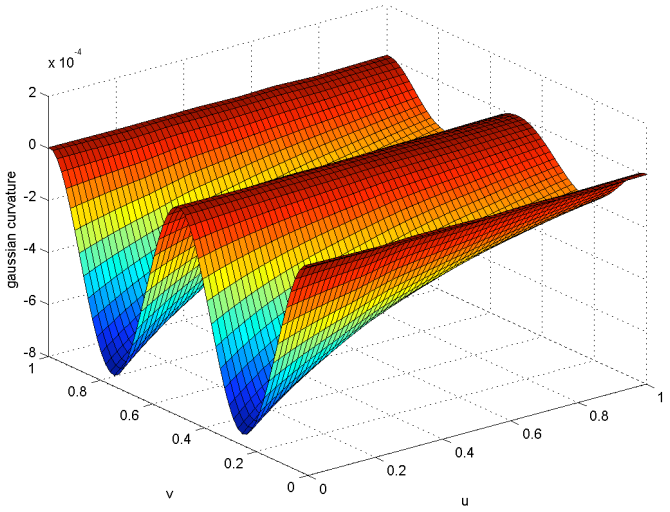


(c) κ_v plot

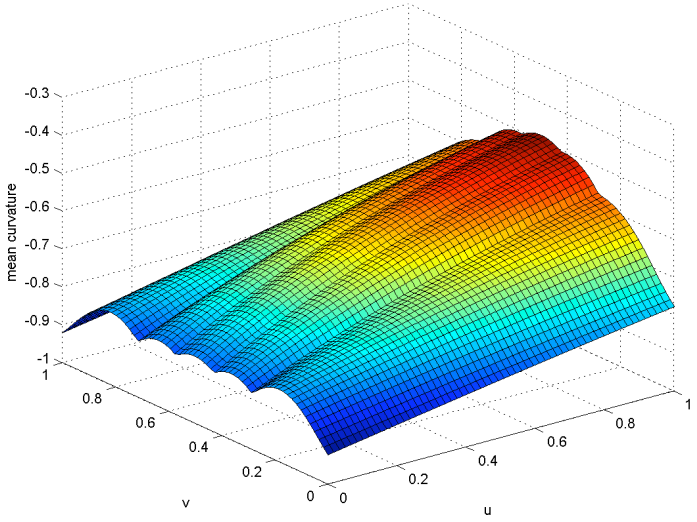


(d) κ_u plot

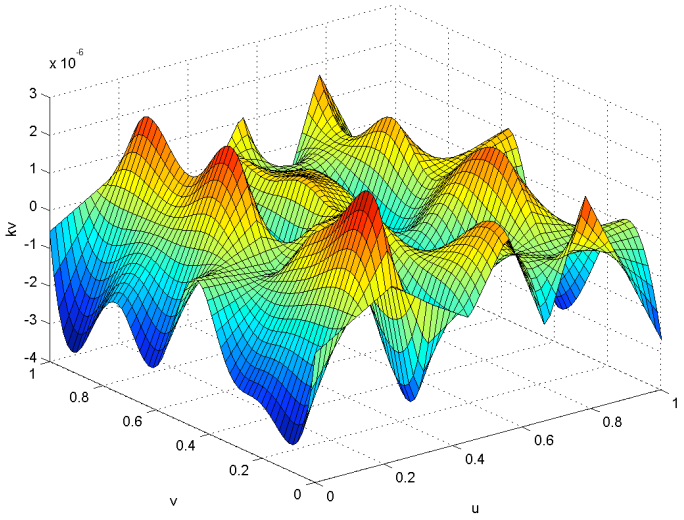
Figure 6.7 Curvature plots of the original wing surface



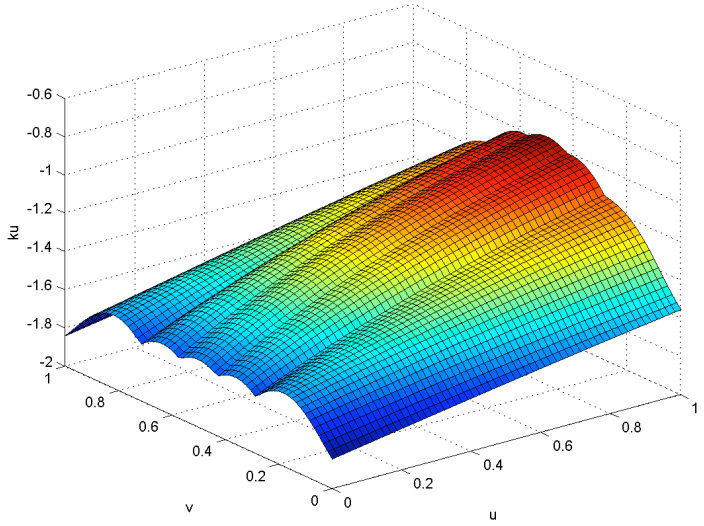
(a) Gaussian curvature plot



(b) mean curvature plot

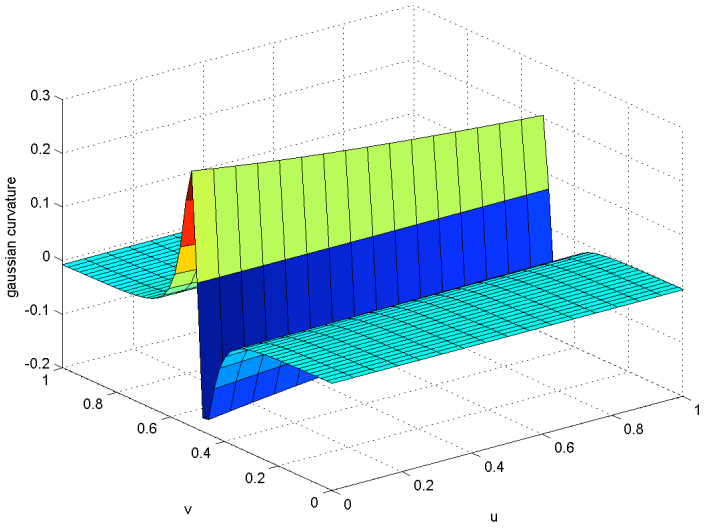


(c) κ_v plot

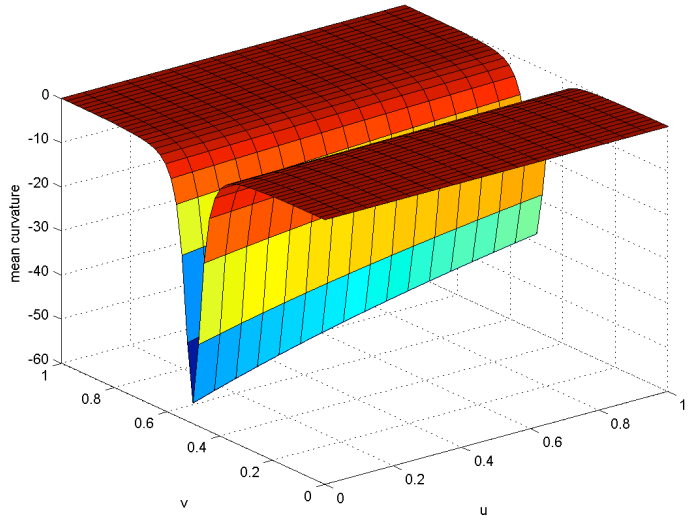


(d) κ_u plot

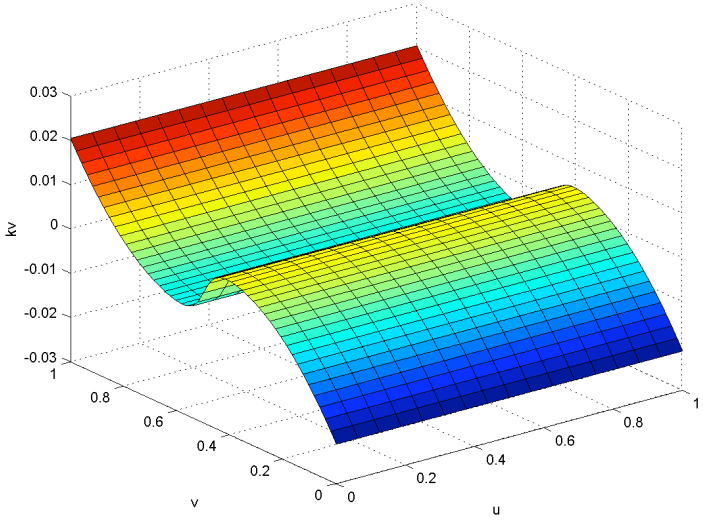
Figure 6.8 Curvature plots of the original fuselage surface



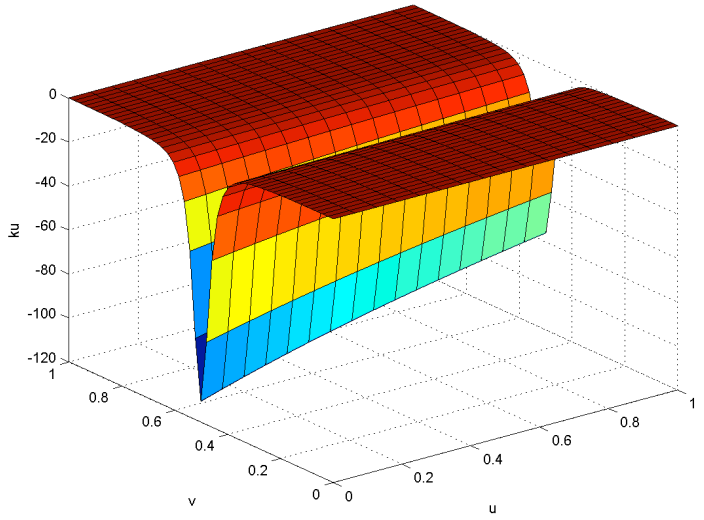
(a) Gaussian curvature plot



(b) mean curvature plot

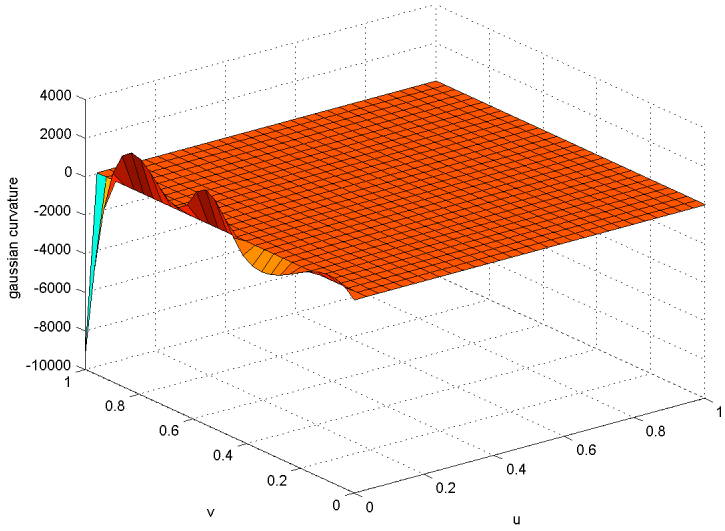


(c) κ_v plot

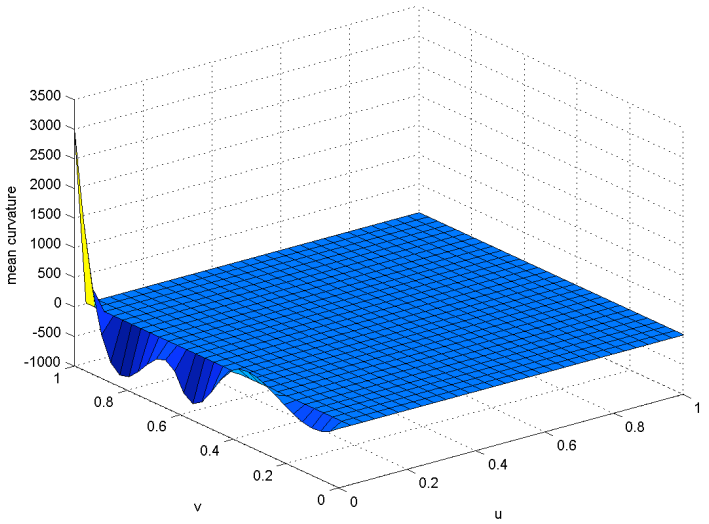


(d) κ_u plot

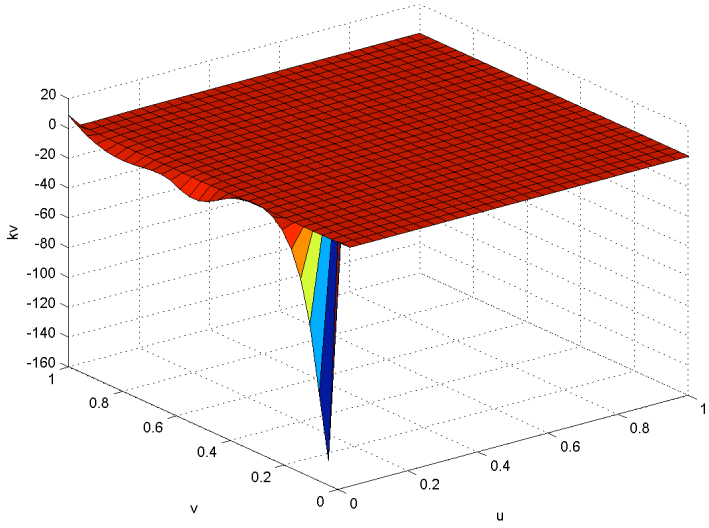
Figure 6.9 Curvature plots of the trimmed wing surface



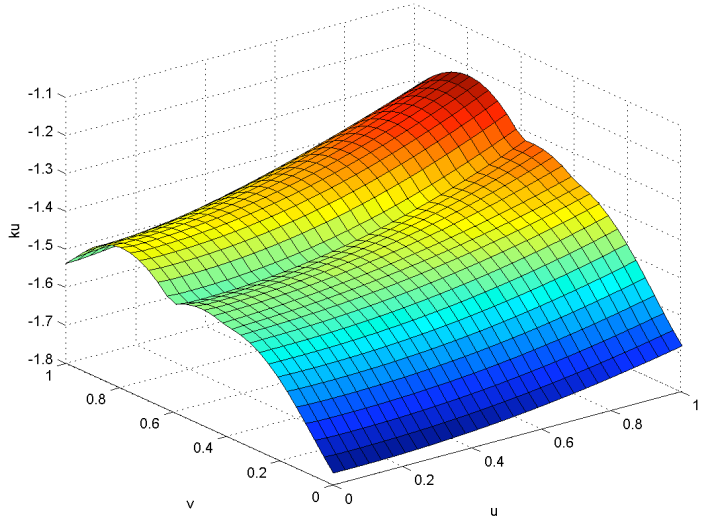
(a) Gaussian curvature plot



(b) mean curvature plot

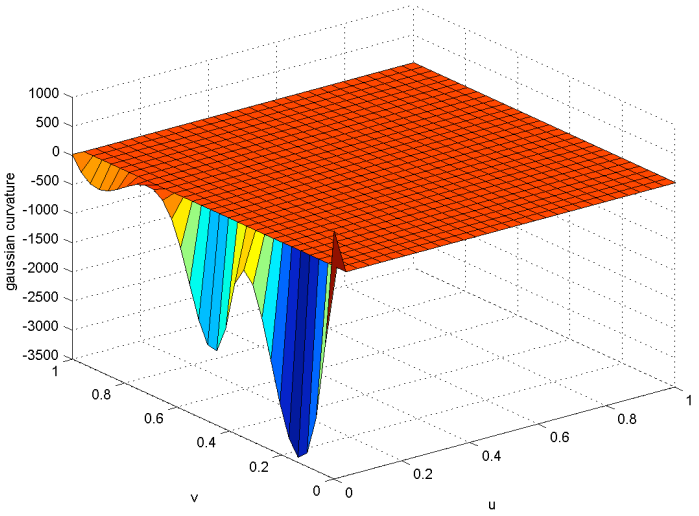


(c) κ_v plot

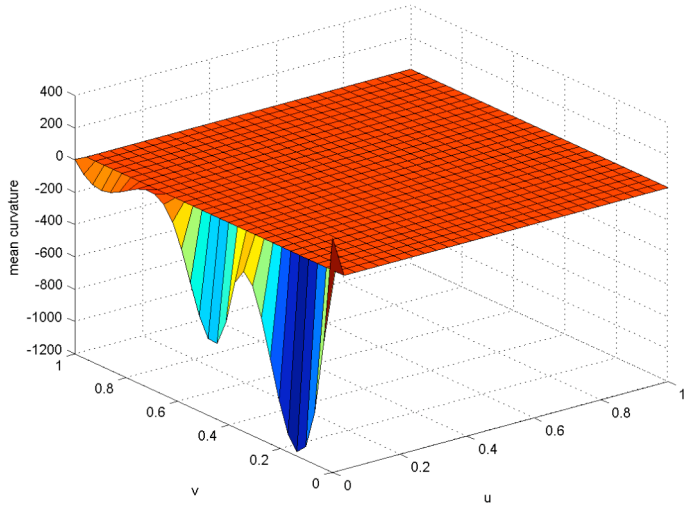


(d) κ_u plot

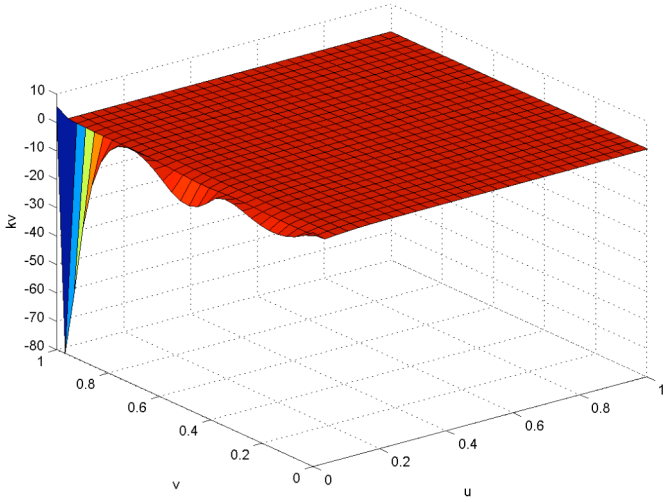
Figure 6.10 Curvature plots of the trimmed fuselage surface region B



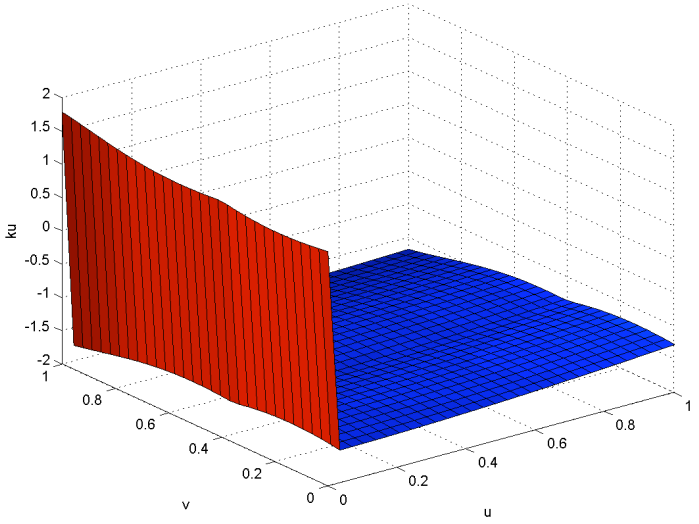
(a) Gaussian curvature plot



(b) mean curvature plot

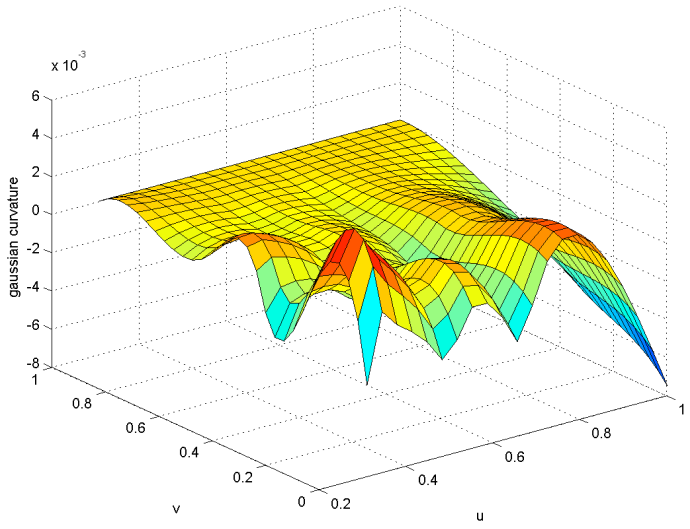


(c) κ_v plot

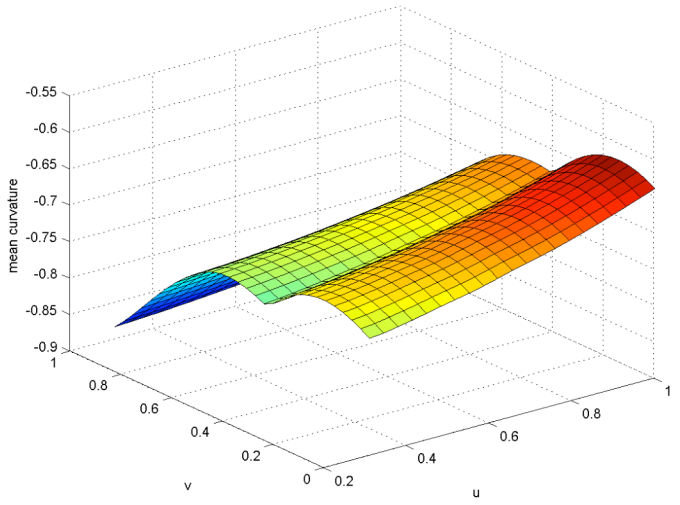


(d) κ_u plot

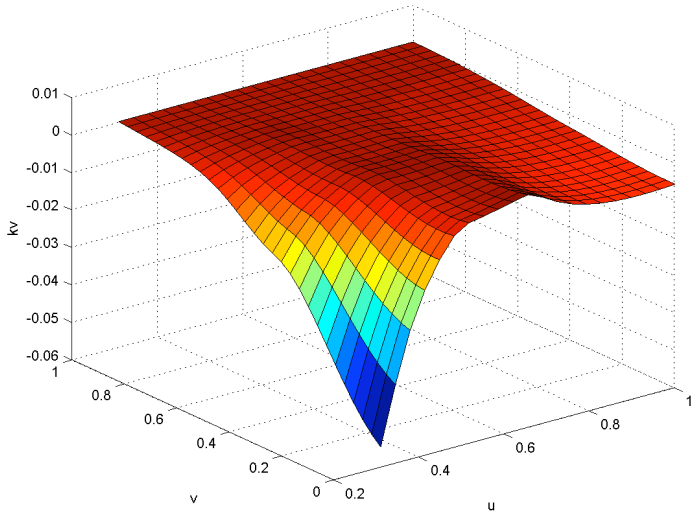
Figure 6.11 Curvature plots of the trimmed fuselage surface region C



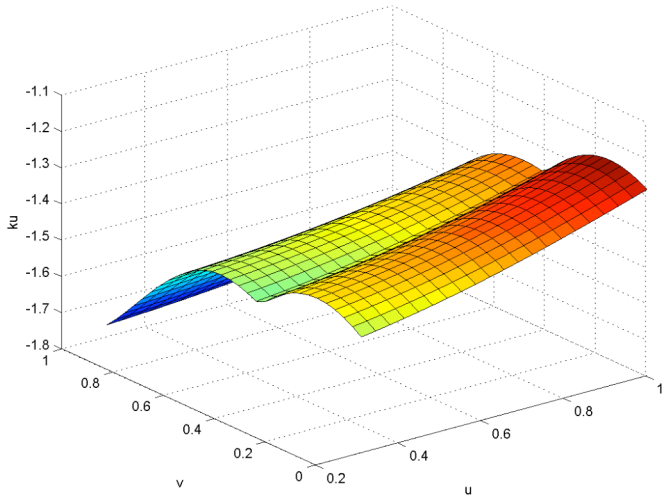
(a) Gaussian curvature plot



(b) mean curvature plot



(c) κ_v plot



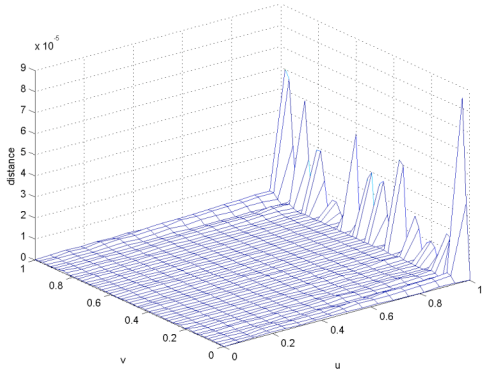
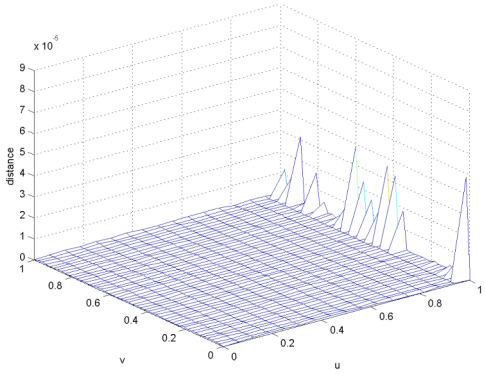
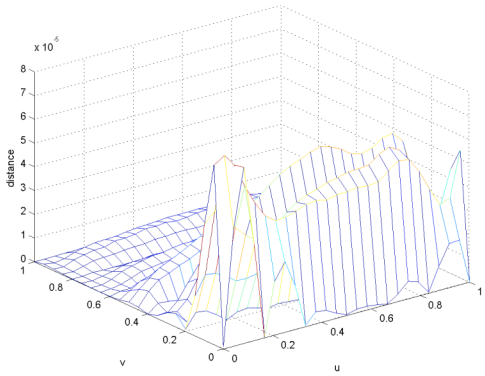
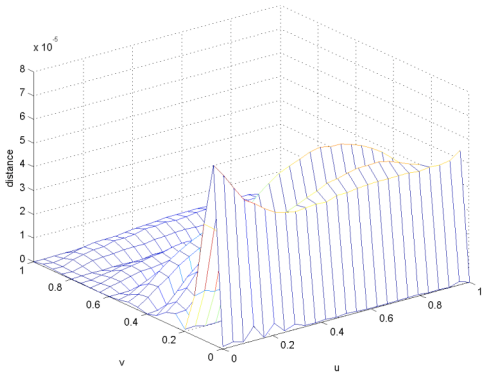
(d) κ_u plot

Figure 6.12 Zoomed curvature plots of the trimmed fuselage surface region C

6.4.1 Number of Interpolation Points

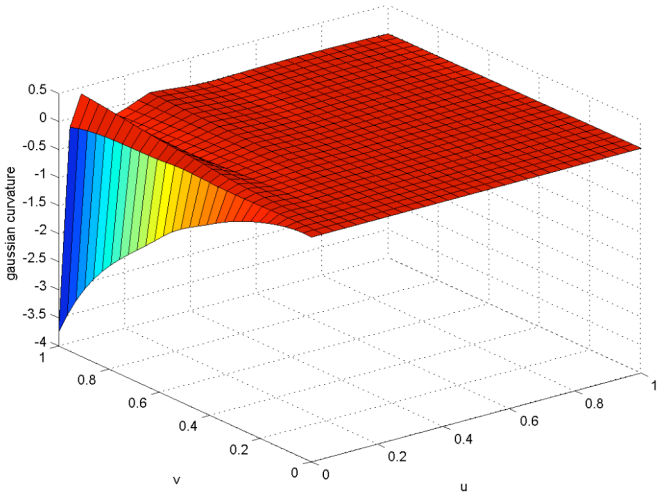
In the above example, the trimming curve is a *u* trimming curve in the wing's parametric space and *v* trimming curves in the fuselage's parametric space. Increasing the number of interpolation along the trimming curve direction decreases the errors of both wing and fuselage surfaces. Table 6.3 is a comparison of the distance plots between using 13 and 25 points on the trimming curve. In general, increasing the number of interpolation points for the trimmed surface decreases the trimming errors. However, in a design optimization process, the geometric definition is directly linked to the number of design variables. Thus reducing the number of parameters involved in the geometric representation is important. The decision has to be made based on the tradeoff between the accuracy and the efficiency.

Table 6.3 Influence of doubling interpolation points on the trimming curve

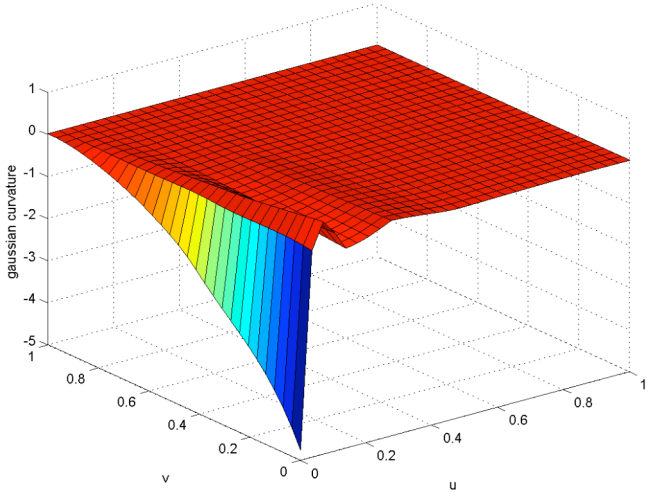
13 points on the trimming curve	25 points on the trimming curve
	
distance plot of the trimmed wing	distance plot of the trimmed wing
d_1 1.4594e-6	d_1 6.7020e-7
d_2 2.1680e-7	d_2 1.1511e-7
d_∞ 8.4740e-5	d_∞ 4.7928e-5
	
distance plot of the trimmed fuselage region C	distance plot of the trimmed fuselage region C
d_1 1.5781e-6	d_1 1.5017e-6
d_2 1.0351e-7	d_2 1.0040e-7
d_∞ 7.8506e-5	d_∞ 7.5396e-5

6.4.2 Parameterization Methods

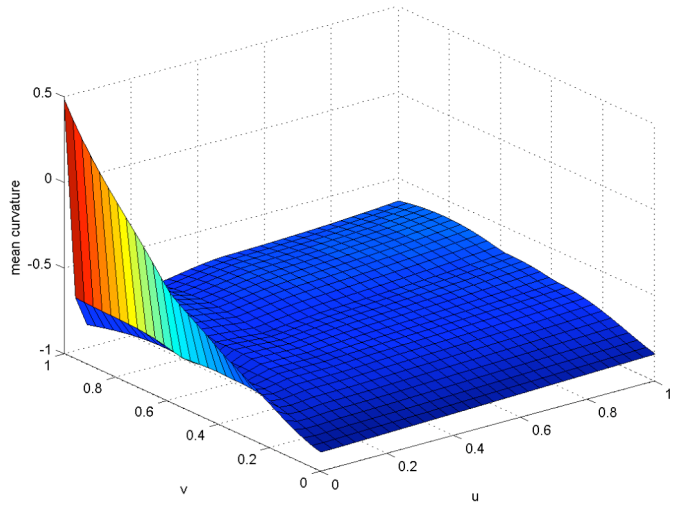
If the chord length method is used to interpolate the trimmed fuselage surface, the curvature change at the leading edge of the intersection airfoil section are reduced (Figure 6.13). However, the trimming error is large. The maximum error is 3.2733e-3 for fuselage region C comparing to that of 7.8506e-5 if the uniform parameterization is used.



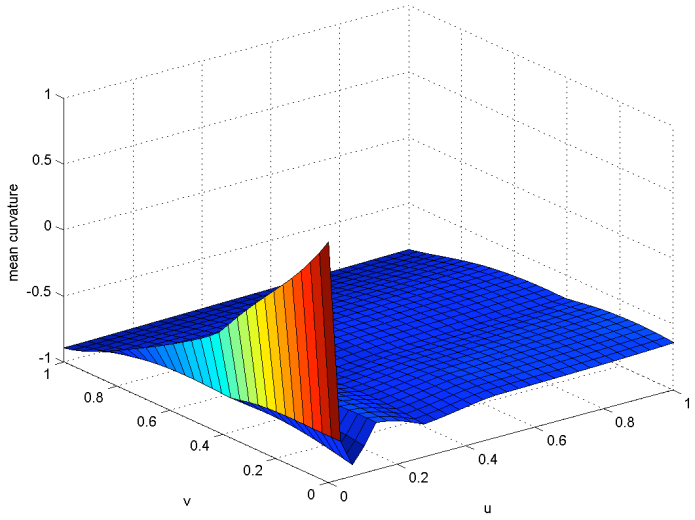
(a) Gaussian curvature of fuselage region B



(b) Gaussian curvature of fuselage region C



(c) Mean curvature of fuselage region B

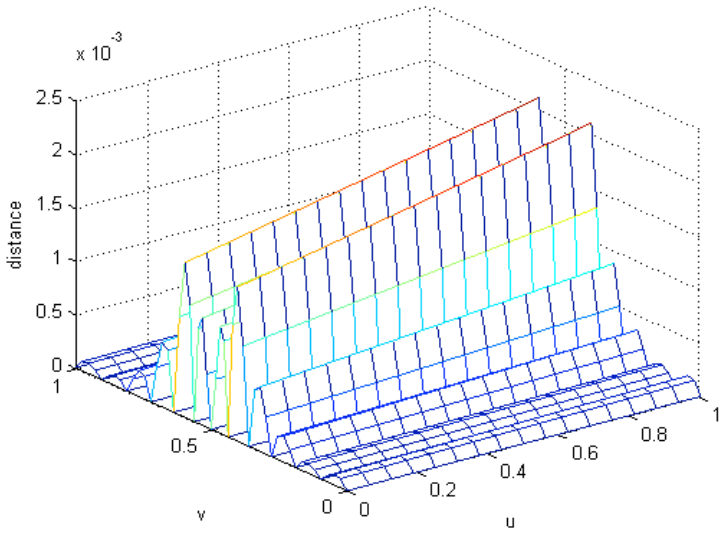


(d) Mean curvature of fuselage region C

Figure 6.13 Curvature plots of trimmed fuselage surface with chord length parameterization

6.4.3 Interpolation Point Locations

The determination of isoparametric curves affects the trimming results. If the parameters used in interpolating the original surface are selected, and the same parameterization method is used, the trimming error is comparatively smaller than that with the same number of isoparametric curves selected from elsewhere. Figure 6.14 compares the results. The errors of (b) are lower than those of (a).

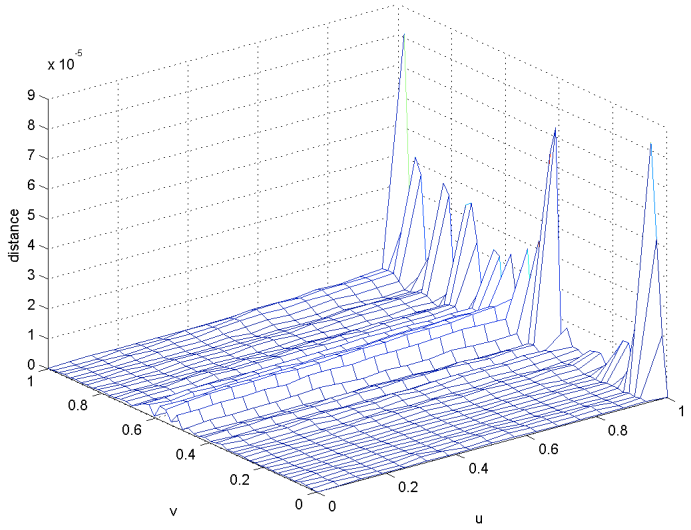


d_1 4.1674e-4

d_2 2.2792e-5

d_∞ 2.1000e-3

(a)



d_1 2.372e-6

d_2 2.5236e-7

d_∞ 8.2433e-5

(b)

Figure 6.14 Distance plots of the surfaces trimmed by (a) not knowing and (b) knowing the interpolation of the original surface

6.4.4 Surface Subdivision Schemes

In the last chapter, a new scheme for subdividing the surface is provided. The defects of the trimmed surface by subdividing the surface into two patches can be easily visualized (Figure 6.15). The defects can be improved by adding interpolation points (Figure 6.16), however, the number and locations of these points are heuristic. The distance plot is provided in Figure 6.17. The parameterization and the number of points on the cross-sectional curve are the same as those of Figure 6.4. The errors are significantly greater in Figure 6.17. Also Figure 6.17(c) is a zoomed graph of Figure 6.17(b), which shows the errors at the region corresponding to that of region D is not trivial, while with the new scheme, the errors are below 1.0e-6.

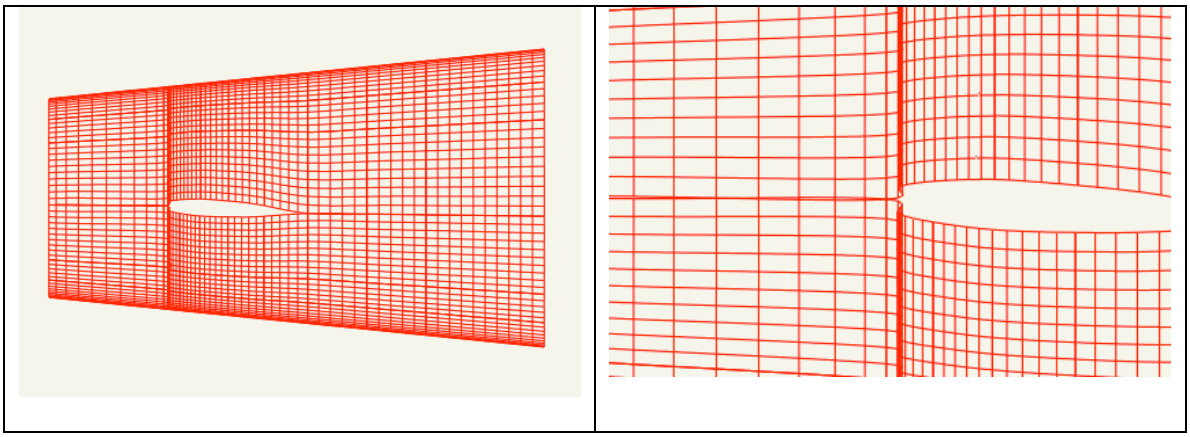


Figure 6.15 Trimming fuselage by subdividing the surface into two patches

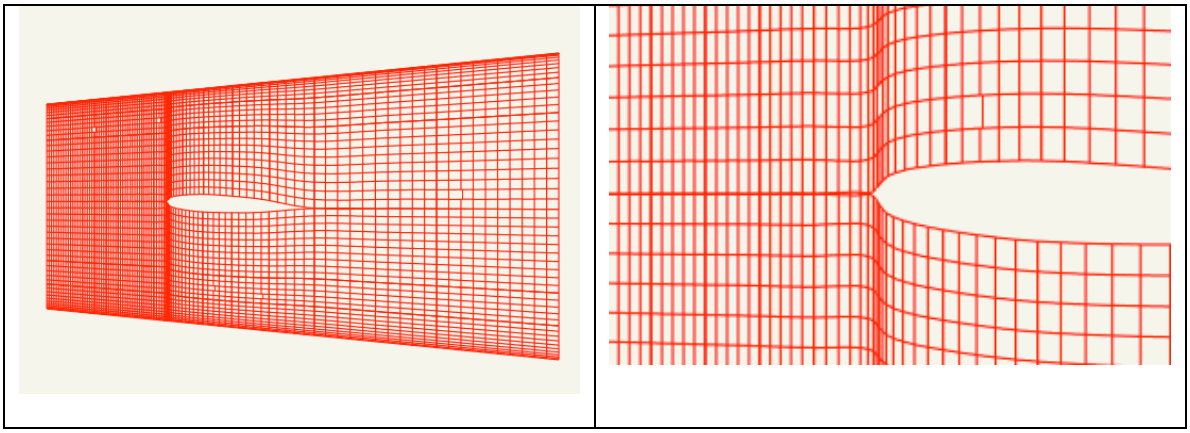
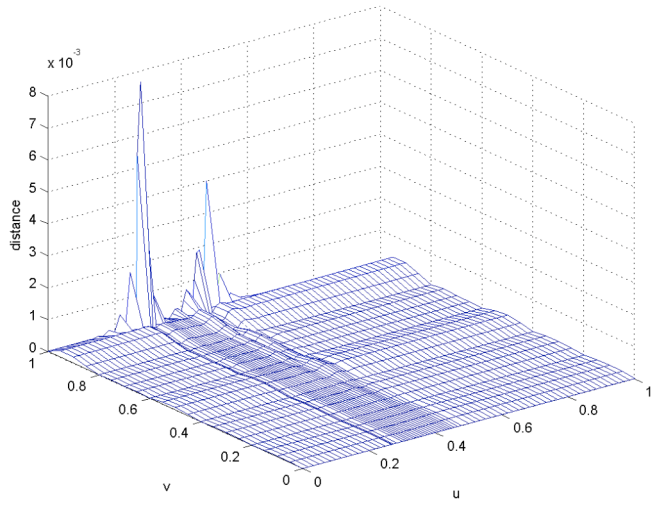
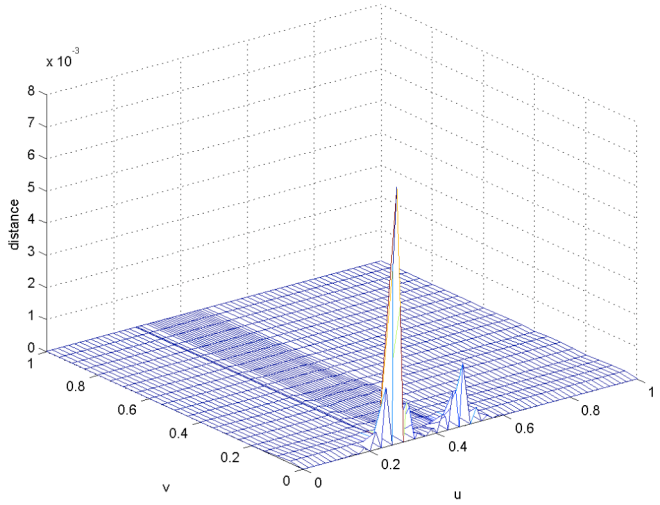


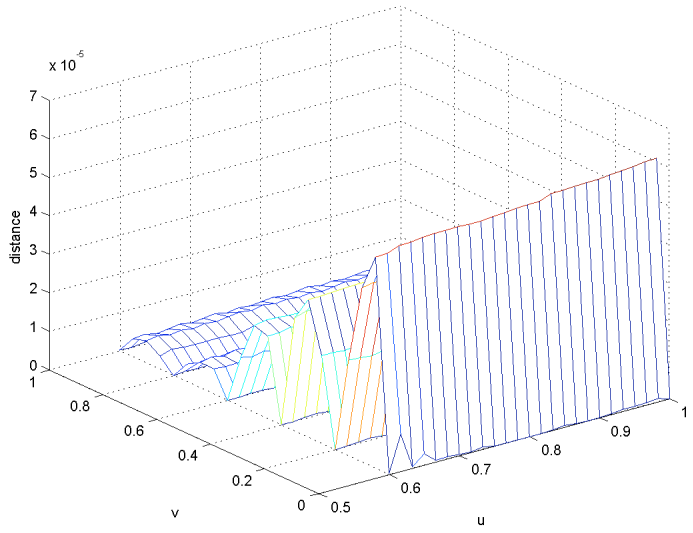
Figure 6.16 Trimming fuselage by subdividing the surface into two patches with double numbers of interpolation points in v direction around the intersection airfoil leading edge



(a) distance plot of the trimmed fuselage lower region



(b) distance plot of the trimmed fuselage upper region



(c) zoomed distance plot of part of the trimmed fuselage upper region

Figure 6.17 Distance plot of a fuselage trimmed by subdividing the surface into two patches

6.5 Conclusions and Summary

In this chapter, the metrics were defined, which are used for measuring the difference between two surfaces. The comparison of trimmed and original surfaces provides us more insight into the geometric trimming process. If the parameters for interpolating the original surface are used in selecting isoparametric curves in the trimming, the trimmed wing surfaces have very small errors. These errors fall into the manufacturing tolerances. The curvatures deviate from those of the original surface slightly. Trimmed fuselage surfaces are divided into four surfaces. Two of them are created by subdividing the original surface through knot insertion and thus match the original surface exactly. The two surfaces in the center have larger errors.

The influence of interpolation points was analyzed. The locations of the interpolation points have great influence on the trimming results. If the parameters for interpolating the original surfaces are used to select isoparametric curves in the trimming

and the same parameterization method is used, the trimming errors may be small. Otherwise, the trimming errors might increase greatly. In many cases, the construction of the original surface is unknown. Increasing the number of interpolation points can be a solution. However, there is a tradeoff between the accuracy and the efficiency. In design optimization, the number of parameters in the geometric representation is directly linked to the design variables. Having a large number of control points will slow the design optimization process and also easily create unusual shapes. Thus the trimmed surface will be optimized in the next chapter to reduce the errors while keeping the number of control points fixed.

7 Optimization of Trimmed B-Spline Surfaces

In general, algorithms for trimming surfaces often incur “gaps” or “overlaps” since the exact trimming is precluded by the remarkably high degree of surface intersections. The possibility of an exact trimming for tensor-product bicubic patches is 324 when regarded as algebraic space curves and 54 in each variable when expressed in the surface parameter domains [Song04]. Unfortunately, the usage of inaccurate models in some high fidelity applications, such as computational fluid dynamics, requires tedious manual “repairing”. Increasing the number of interpolation points can reduce the errors but at the same time increase the complexity for applying the trimmed surfaces in engineering design. Also, the trimming algorithm obtains less accurate results when the interpolation points are selected at “bad” locations. Without the knowledge of the construction of the original surface, selecting the right interpolation points is challenging. Usually, optimization methods provide insights into problems whose domains are complex and unclear. In this chapter, a novel global optimization method, so-called hybrid Parallel Tempering (PT) and Simulated Annealing (SA) method, is employed to reduce the errors of trimmed B-spline surfaces.

Global optimization methods are getting more attention recently. Research has shown that the NURBS shape optimization problems are highly nonlinear with many local optima. The choice of the initial guess is crucial if local optimization methods are used. On the other hand, the convergence rate is critical for applying global optimization methods. The hybrid PT/SA method, which is an effective algorithm to overcome the slow convergence, waiting dilemma, and initial value sensitivity, is a good candidate for optimizing geometrically trimmed B-spline surfaces.

7.1 Optimization Methods

Optimization methods are usually classified into two categories, global and local. Local optimization methods include direct search methods and descent methods. The direct search methods require only objective function evaluations and do not use the partial derivatives of the function. These methods are often suitable for simple problems with a small number of variables. The descent methods, which require computation of partial derivatives, are in general more efficient. Two standard heuristics are widely used to find global minima by using local optimization methods. One is to find local minima starting from widely varying initial values of the independent variables and then pick the minimum; the other is to perturb a local minimum by taking a finite amplitude step away from it and then see if the routine returns a better point. However, in a high dimension problem, finding good initial values or perturbation is difficult if not impossible.

Among a variety of stochastic global search algorithms, we should distinguish the parallel tempering and simulated annealing methods. They are based on the classical stochastic method, Metropolis, and have demonstrated important successes on a variety of global optimization problems, especially those where a desired global minimum is hidden among many poorer local minimums. The hybrid PT/SA method, which is based on the PT and SA method, recently has been applied to molecular biology and has shown promising results in solving complex, high-dimension problems and overcoming slow convergence [Li05]. The following describes these methods.

7.1.1 Metropolis Method and Simulated Annealing Method

The Metropolis method is one of the most popular Markov Chain Monte Carlo methods (MCMC). The MCMC methods create a sequence of points randomly. The probability of jumping from one point to another depends only on the last point and not on the entire previous history.

The Metropolis method is very simple yet applicable to a wide class of problems. It is based on the Boltzmann probability distribution [Metr53]. The basic idea is that a

system in thermal equilibrium at temperature T has its energy probabilistically distributed among all different energy states E: $\text{Prob}(E) \sim \exp(-E/kT)$. The quantity k is a constant of nature that relates temperature to energy. The system sometimes goes uphill as well as downhill, so that it has a chance to escape from a local energy minimum to find a better one. But the lower the temperature, the smaller probability it goes uphill. Other systems can be analogous to the thermodynamic system. We associate our objective function F with the energy E, then the method can be described as: Take a random step from the current point. If $F_{i+1} < F_i$, the new point is accepted. Otherwise, the move will be accepted by a probability of $r = e^{-(F_{i+1}-F_i)/kT}$. The actual implementation of the Metropolis method is sensitive to the length of the steps. If the steps are too small, the Markov process can be easily trapped into a deep local minimum from where it cannot escape in practical time. On the other hand, if the steps are too large, the acceptance rate tends to be low and it may ignore some "local details".

The Simulated Annealing method (SA) is based on the Metropolis method [Kirk83]. Instead of having a fixed temperature, the SA method reduces the temperature during the process. It is analogous to metals cooling and annealing. If the liquid is cooled slowly, the atoms are often able to line themselves up and form a pure crystal that is completely ordered. This crystal is the state of minimum energy for this system. Initially, the Simulated Annealing method raises the initial temperature to a high value, and slowly and gradually reduces it during the procedure to help escape from the local minima. This method can also be used for other systems by introducing a control parameter, analogous to temperature, and an annealing cooling schedule that describes its gradual reduction. In a problem with a rugged energy landscape, such as the surface approximation problem, if the system is still trapped into a local minimum when the temperature is lowered to a certain level, there is a low possibility of escaping from the local minimum.

7.1.2 Parallel Tempering Method

The Parallel Tempering (PT) method, also known as the multiple Markov chain or replica-exchange method, builds up on Markov chains of different temperature [More03].

The replica transition moves between different temperature levels, which enables the system at the low temperature level to escape from local minima and to locate multiple minima by allowing it to switch with the system configuration at higher temperature according to the Metropolis-Hasting rule [Metr53]. The replica moves accelerate the system to reach equilibrium.

7.1.3 Hybrid Parallel Tempering and Simulated Annealing Method

The main idea of the hybrid PT/SA method is to apply the PT moves to the SA scheme to reduce the relaxation time to equilibrium when temperature is changed in SA, which significantly improves the convergence rate of the optimization process. By combining the PT and SA method, a configuration is able to switch between low and high temperature levels in the evolvement of the optimization process and at each level, the temperature will cool down to its target temperature. The moves at high temperature levels intend to explore the energy landscape with greater step length while those at the low temperature levels search the local details.

The hybrid PT/SA method also contains two sets of moves. One is to move on the same Markov chain. The acceptance rate, $\min(1, e^{-(F^{new}-F^{old})/kT})$, is defined in the same way as that of the SA method. The other move is the replica transition, which exchanges the configurations on two adjacent Markov chains:

$$\begin{aligned} x_j^{new} &= x_{j+1}^{old} \\ x_{j+1}^{new} &= x_j^{old} \end{aligned} \tag{7-1}$$

The two adjacent Markov chains are selected randomly every time and the move is acceptable with the probability of $\min(1, e^{-(F_{j+1}/kT_j + F_j/kT_{j+1}) + (F_j/kT_j + F_{j+1}/kT_{j+1})})$, where F_j denotes the objective function evaluated on the current point of the *j*th Markov chain.

7.2 Objective Functions

The objective function, is built as a combination of d_2 and d_∞.

$$F(X) = const_2 \times d_2^2 + const_\infty \times d_\infty^2 \tag{7-2}$$

where X is the vector of variables containing the positions of the NURBS control points, weights or knot sequences. The normalization constants $const_2$ and $const_\infty$ are used to obtain the same order for the two terms and adjust the objective function ranges. Using the square of d_2 and d_∞ is for efficiency. At each iteration the objective function is computed based on a significant number of points. Avoiding the square root operation greatly reduces the computation time, and yet reflects the difference between the two surfaces.

7.3 Variables and Constraints

The approximation of the surface with a fixed degree and fixed number of control points is investigated. A NURBS surface is determined by control points, weights, and knots. The movement of control points has no constraints in Euclidean space while the weights must stay positive and the knot sequence must be non-decreasing. The dimension of variables can easily go beyond one hundred. A NURBS surface with degree (p,q) and $m \times n$ control points has two knot sequences of $p+1+m$ and $q+1+n$. Since the knots are relative values, it can be normalized on $[0,1]$, the first and the last knots can be fixed in the optimization. If clamped knot sequences are used, the first $p+1$ knots are the same, so are the last $p+1$ knots. This further reduces the number of knot variables. The size of the control point location variables is $3mn$ while that of the weight variables is mn. Modifying only the weights has more influence on reducing the maximum distance than reducing the average distance. However, compared to modification of only the control points or knots, the results are insignificant. Furthermore, evaluation of surface points of a rational B-spline surface is more time-consuming. Figure 7.1 compares the results obtained by modifying control points and knots. Modifying control point locations yields better results. Setting both the control point locations and knots as variables increases the

dimension of variables. As a result, tuning the optimization parameters is more difficult. Therefore, only control point locations are set as variables.

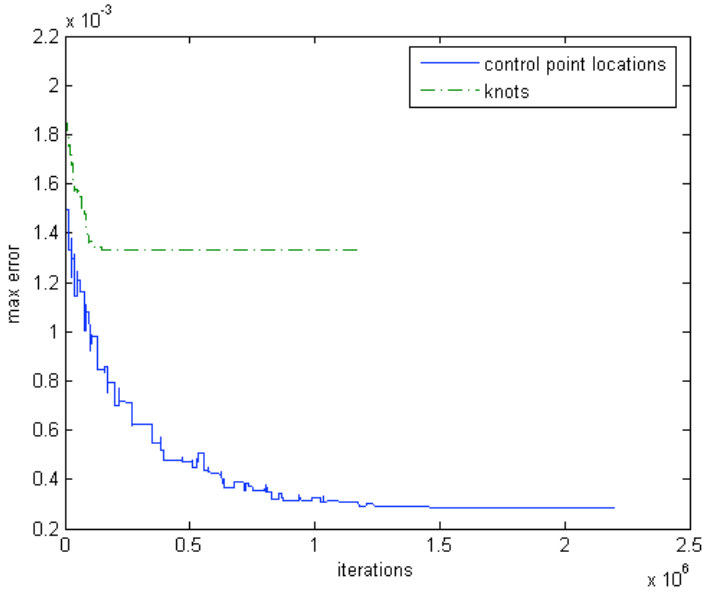


Figure 7.1 Comparison of maximum error convergence of setting control point locations and knots as variables

7.4 Implementation of Hybrid PT/SA Method

At each iteration a proposed point (variable vector) is created by a random move from the current point. The scheme of creating the random move is important. The new point can be located far away from the current point to speed up the search process and to avoid getting trapped at a local minimum, or located nearby to focus on the local area. This can be controlled by a step size procedure. If a single variable is moved, the convergence of the system is slow. Allowing every variable to make a small move from the current value is more efficient. The step size in different coordinates should be adjusted to provide efficient moves. For example, the cross-section (airfoil profile) of a wing surface might have greater curvature changes and greater changes of the distances

from the surface to the target points, and thus applying step sizes, which allow greater movement in this direction, is more efficient.

Point projection is a computationally costly operation, since every point projection is an optimization process per se [Pieg97]. Though good initial values can reduce the convergence time, doing point projection during each iteration slows down the process. However, without doing point projection, the images of target points in the u-v parametric space are fixed and the system will be trapped in local minima. Allowing the images of target points in the *u-v* parametric space to move is critical. An alternative way is to perform point projection every certain number of iterations. To further improve the efficiency, the old values are used as the initial values in point projection since the changes are not significant. Figure 7.2 shows the movement of an image of a target point in the surface's parametric space.

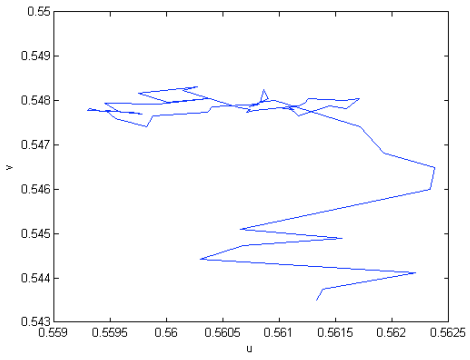


Figure 7.2 The trace of a target point projected in the parametric space

7.5 Results and Discussion

The hybrid PT/SA method was applied to the trimmed wing surfaces. The interpolation points for trimmed surfaces are sampled uniformly in the parametric space of the original surface when the construction of the original surface is unknown. The trimming errors are comparatively larger in these cases. Instead of increasing the number of control points, the configuration of the NURBS surface is modified to provide a better

fitting of the target surface. First the method is validated by using target points on a plane.

7.5.1 Validation for a Plane

If the target points are on a plane, the control points are on the plane. The method should be able to move the control points to the plane. The 10×10 target points are set to be uniformly distributed on the plane $z=0$ bounded by lines $x=0, x=1, y=0, y=1$, and a bicubic NURBS surface has 4×4 control points and clamped end knots. Though the solution is not unique, the control points on the boundary should lie on these lines and the z coordinates of all the control points should move toward 0. For different initial surfaces, the optimization process converges to these solutions. Figure 7.3 gives the distance plots. Figure 7.4 plots initial and optimized control points. Four temperature levels are applied in this example. The smallest step size is 2e-4. The resulting surface has a maximum distance of 3.99e-5 from the target points.

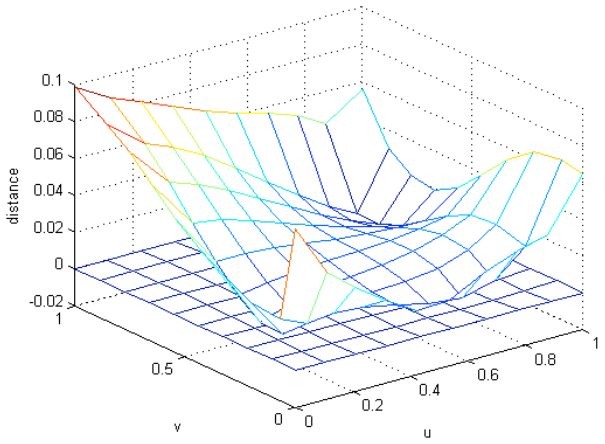
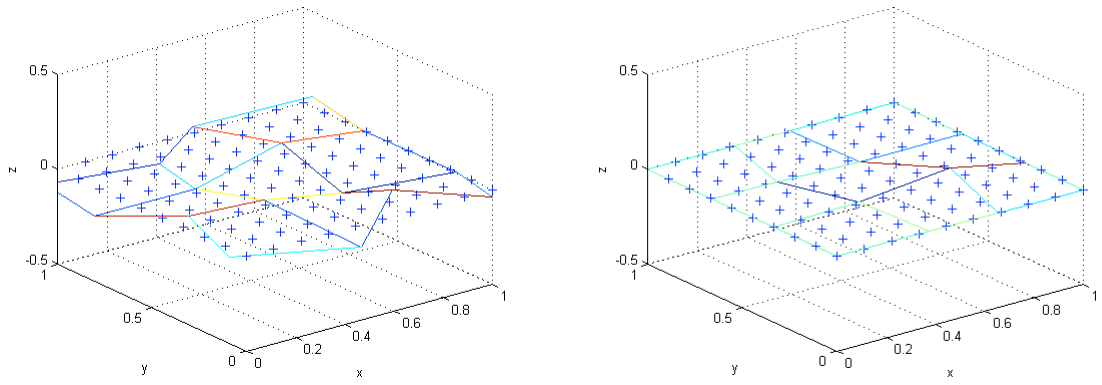


Figure 7.3 Distance plot of the initial and optimized surfaces
(the max error of the initial surface is 9.8887e-2;
the max error of the optimized surface is 3.993e-5)



(a) Initial control hull (b) Optimized control hull

Figure 7.4 Target points and control hulls

7.5.2 Optimization of Trimmed Wing Surfaces

When a wing is trimmed, the isoparametric curves are selected uniformly in the parametric space of the original surface if the construction of the original surface is unknown. Target points of a trimmed wing is shown in Figure 7.5. In this example, the trimmed wing is a bicubic B-spline surface created by interpolating 13×5 surface points (13 along the trimming curve and 5 on each isoparametric curve). Control point locations are set as variables and the problem has a dimension of 195. The target points are 49×17 surface points on the retained part of the original surface.

The important factors controlling the optimization process include the temperature levels, the cooling scheme, the step sizes, and the configuration exchange among temperature levels. The temperatures are lowered proportionally. Four temperature levels start with 4, 16, 64 and 256, and decrease proportionally by a fraction of 0.997 of (T-1.0) every 1000 steps. The step size varies from 2.0.e-6 to 1.28-4 for x coordinates, 0.4e-6 to 4.58e-5 for y and z coordinates. The optimized surface has the maximum error reduced from 2.0259e-3 to 2.1903e-4 and the distances from the target points distributed evenly (Figure 7.6). The average error decreases from 6.5665e-4 to

1.0109e-4. The errors are close to the errors of the surface trimmed with prior knowledge of the original surface. Figure 7.7 shows the initial and optimized control hulls.

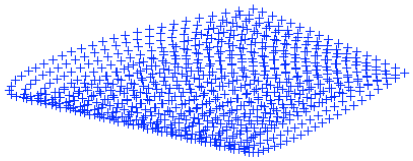


Figure 7.5 Target points of the trimmed wing surface

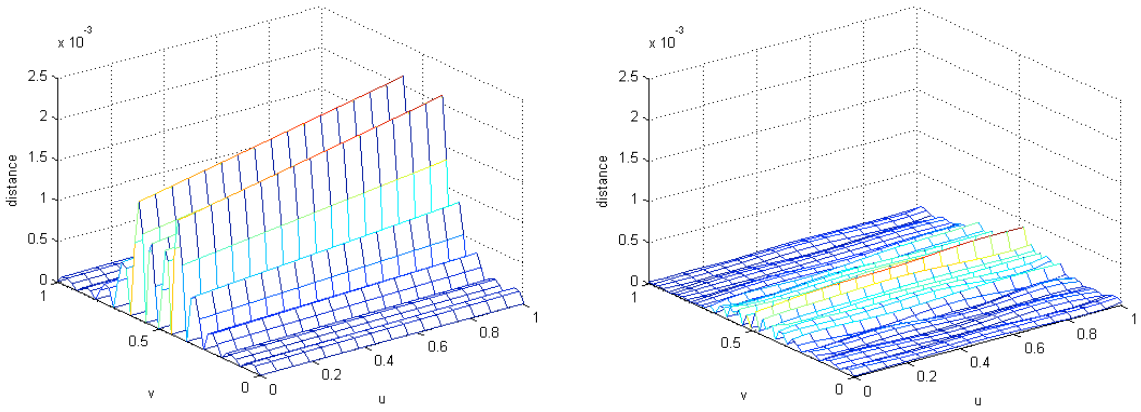


Figure 7.6 Distance plot before and after optimization

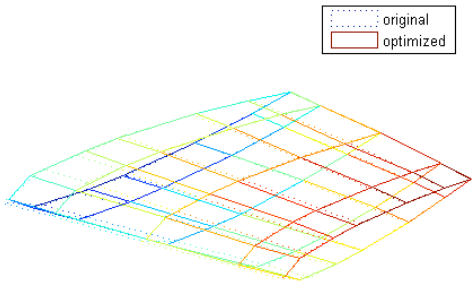


Figure 7.7 Control hulls of the initial and optimized surfaces

The optimization process is faster on a smaller number of targets points since the complexity of the problem reduces and the time for evaluating the objective function is less. An alternative approach to fit a large number of points is to start with a coarser target point net and refine it progressively. The maximum error of fitting the 49×17 target points by the optimized surface obtained from fitting 25×9 points is 4.0203e-4, compared to that of 2.0259e-3 for the original trimmed surface.

7.5.3 Discussion

For high dimension problems with rough energy landscapes, traditional local optimization methods such quasi-Newton experiences great difficulties in finding good initial values, while the hybrid PT/SA method is not sensitive to initial values. A major problem for using the SA method is the setting of the step size. A big step size misses the details and a small step size leads to slow convergence. Figure 7.8 compares the convergence of SA and the hybrid PT/SA for fitting 25×9 points of a trimmed wing surface. Four temperature levels are used in the hybrid PT/SA method. The SA with large step size and small step size employ the configuration of the lowest and highest temperature levels of those of the hybrid PT/SA respectively. The number of iterations of

the hybrid PT/SA method, which converges to the same level of accuracy, is about 2/3 of that of SA method with small step size. In SA with a big step size, the objective function decreases rapidly at beginning, but is trapped in a local minimum of 4.9e3. Overall, the hybrid PT/SA approach yields a better convergence rate than both cases in SA. Though these results are encouraging, studies on setting the parameters are expected to further improve the results.

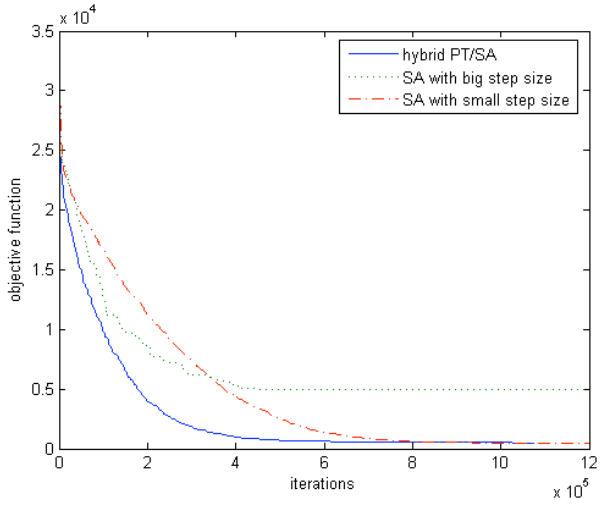


Figure 7.8 Convergence Comparison

The hybrid PT/SA method has been shown to be an appropriate method for reducing trimming errors. In this application, the number of the control points and the surface degree are fixed. Control point locations are selected to be variables. Examples are given to reduce geometrically trimmed wing surfaces. Without the prior knowledge of the construction of original surfaces, errors can be reduced to be close to that of surfaces created by using the parameters and parameterization of interpolating the original surface. Being able to represent a trimmed wing surface more accurately with a small number of control points is important for wing design.

7.6 Summary

A novel optimization method, the hybrid PT/SA method, is applied to optimize the trimmed surfaces and overcomes the initial value and low convergence problems. This chapter introduced the method and discussed the construction of objective functions and the implementation details. The results confirm expectations.

8 Conclusions

Geometric trimming of B-spline surfaces is investigated and an optimal geometric trimming method is provided in this dissertation. Limited research had been carried out previously. The first step in geometric trimming is to find the trimming curve, which is the intersection of two B-spline surfaces. Most existing B-spline surface intersection algorithms focus on creating intersection curves suitable for rendering. Intersection curves in geometric trimming require good connectivity and fewer curve segments. In this dissertation, an efficient intersection algorithm, which is suitable for geometric trimming, is developed. This algorithm selects isoparametric curves based on the characteristics of one surface and intersects these curves with the other surface. The intersection points are then connected to create intersection curves by a marching scheme. A subdivision method is used to obtain such curve-surface intersections. The curve is subdivided until it can be approximated by a straight line and the surface is subdivided until the patch can be approximated by a quadrilateral within a given tolerance. Finally, the intersection points are computed by a line-quadrilateral intersection algorithm. The number of intersection points depends on the number of isoparametric curves selected, and thus is controllable and independent of the error bound of intersection points.

The trimming curves are classified and a new scheme for subdivision of a surface trimmed by a closed trimming curve is presented. The surface trimmed by a closed trimming curve is subdivided into four patches and the trimming curve is converted into two open trimming curves. Two surface patches are created by knot insertion, which match the original surface exactly. The other two surface patches are trimmed by the converted open trimming curves. The scheme provides more accurate trimmed surfaces. In this dissertation, the factors affecting the trimming results are discussed and trimmed

surfaces are optimized by applying the hybrid Parallel Tempering and Simulated Annealing method to reduce the trimming error. Without the prior knowledge of the construction of the original surface, the uniformly distributed isoparametric curves are selected to sample the interpolation points on the retained portion of the original surface. The errors are significantly greater in these cases. The optimization results show that control points of trimmed surfaces can be optimized and the maximum trimming errors are lowered one order of magnitude.

The results of this study provide an effective means to geometrically trim aircraft B-spline surfaces with high accuracy.

This work also has significant meaning in related fields. The intersection algorithm presented in this dissertation provides controllable level of detail. Intersection points can be computed and rendered progressively to refine the curve or in parallel to improve the efficiency. The optimization technique is not restricted to optimizing trimmed surfaces. It can be applied to general surface fitting problems while a B-spline surface is modified to fit a large number of surface points.

This dissertation also lays the foundation for future research. The focus of this study is geometric trimming of wing and fuselage surfaces. However, the method can be extended to other components in the future. In the trimming algorithm provided in this dissertation, an open trimming curve must be monotone along the trimming direction and a closed curve must be able to be converted into two such open curves. Otherwise, the subdivision of the surface might be more complicated and require thorough study of the possible situations. Applying potential theory to resample surface points might also be a solution. The hybrid PT/SA method will be further studied and guidelines for setting optimization parameters will be obtained from either theoretical or experimental methods. Furthermore, the hybrid PT/SA method can be applied to aerodynamic optimization of aircraft components and is expected to explore wider varieties of surface shapes.

Appendix A Computing the Curvature of NURBS Curves

Curvature is an important curve property. Formulae for the derivatives of nonrational B-spline curves and NURBS curves are well known, thus the curvature can be obtained by its definition. However, derivatives of NURBS curves are complicated, involving denominators to high powers. This appendix describes direct methods to compute NURBS curve curvature with respect to the parameters, based on the de Boor algorithm.

A.1 Derivatives of rational curves

Given the control points $P_0, P_1 \ldots P_{L+n-1} \in \mathrm{R}^3$ and the associated nonnegative weights $w_0, w_1, \ldots w_{L+n-1}$, the knot sequence $u_0 \le \ldots \le u_{L+2n-2}$, a NURBS curve of degree n containing L segments is defined to be

$$R(u) = \frac{\sum_{i=0}^{L+n-1} N_{i,n}(u) w_i P_i}{\sum_{i=0}^{L+n-1} N_{i,n}(u) w_i} \tag{A-1}$$

where $N_{i,0} = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{otherwise} \end{cases}$

$$N_{i,k}(u) = \left(\frac{u - u_{i-1}}{u_{i+k-1} - u_{i-1}}\right) N_{i,k-1}(u) + \left(\frac{u_{i+k} - u}{u_{i+k} - u_i}\right) N_{i+1,k-1}(u) \text{ for } k \in \{1, 2 \ldots, n\}$$

Strictly speaking, two more knots u_{-1} and u_{L+2n-1} ought to be included, but these knots have no influence over the curve (for $u_{n-1} \le u \le u_{L+n-1}$). In this appendix, they are omitted.

If we let the numerator in (A-1) be $P(u)$ and the denominator $w(u)$, since $P(u) = R(u)w(u)$,

$$R'(u) = \frac{1}{w(u)}[P'(u) - w'(u)R(u)] \tag{A-2}$$

The second derivative is

$$R''(u) = \frac{1}{w(u)}[P''(u) - 2R'(u)w'(u) - w''(u)R(u)] \quad \text{(A-3)}$$

$P'(u), P''(u), w'(u), w''(u)$ can be calculated with the formula for nonrational B-spline curves . The curvature formula for *u* equals 0 is

$$\kappa_R(0) = \frac{n-3}{n-2}\frac{u_n}{u_{n+1}}\frac{w_0 w_2}{w_1^2}\frac{b}{a^3} \quad \text{(A-4)}$$

where $U = (0,...,0,u_n,u_{n+1},...,u_{n+L-2},1,...,1)$ is the knot sequence, $D = (d_0,...,d_L)$ is the control polygon, w_0, w_1, w_2 are the first three weights and $a = \|d_1 - d_0\|$, $b = \|(d_1 - d_0) \times (d_2 - d_0)\|$. Curvature at any point can be obtained by applying the subdivision property.

The curve itself can also be defined recursively by the de Boor algorithm [Fari95, Fari97]. Suppose that $u_I \le u < u_{I+1}$, the only control points that influence the curve in this interval are $P_{I-n+1},...,P_{I+1}$, Let $P_{i,0}(u) = P_i$ for $i = I-n+1,...,I+1$, and let

$$\alpha_{i,k} = \frac{u - u_{i-1}}{u_{i+n-k} - u_{i-1}} \quad \text{(A-5)}$$

$$w_{i,k} = (1-\alpha_{i,k})w_{i-1,k-1} + \alpha_{i,k}w_{i,k-1} \quad \text{(A-6)}$$

$$w_{i,k}P_{i,k} = (1-\alpha_{i,k})w_{i-1,k-1}P_{i-1,k-1} + \alpha_{i,k}w_{i,k-1}P_{i,k-1} \quad \text{(A-7)}$$

for $k = 1,...,n$ and $i = I-n+k+1,...,I+1$, then $R(u) = P_{I+1,n}$ is the NURBS curve, see Figure A-1.

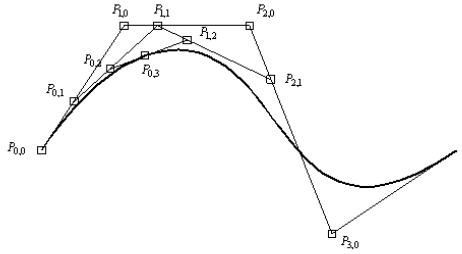


Figure A-1. Recursive scheme of calculating $P(u)$

The first derivative of $P_{i,k}(u)$ is given by

$$P'_{i,k}(u) = k\frac{w_{i-1,k-1}w_{i,k-1}}{w^2_{i,k}}\left(\frac{P_{i,k-1} - P_{i-1,k-1}}{u_{i+n-k} - u_{i-1}}\right) \qquad \text{(A-8)}$$

Thus the first derivative of a NURBS curve $R(u)$ is

$$R'(u) = n\frac{w_{I,n-1}(u)w_{I+1,n-1}(u)}{w^2_{I+1,n}(u)}\left(\frac{P_{I+1,n-1}(u) - P_{I,n-1}(u)}{u_{I+1} - u_I}\right) \qquad \text{(A-9)}$$

The proof of (A-8) and (A-9) is in [Floa92].

Throughout the rest of this section, u is dropped without ambiguity. Also, the assumption is made that R is differentiable at u.

A.2 The Curvature of a NURBS Curve

Curvature calculation using equation (A-3) and (A-4) can be simplified by multiplying them out and applying the cross product of a vector and itself is zero. Then

$$R' \times R'' = \frac{1}{w^3}[wP'' \times P' - w''P \times P' - w'P'' \times P]$$

and $$\kappa(u) = \frac{w^3\left\|(wP'' - w''P) \times P' - w'P'' \times P\right\|}{\left\|wP' - w'P\right\|^3} \qquad \text{(A-10)}$$

Formula (A-10) does not need to calculate the first and second derivatives of the rational curve. However, it is difficult to provide a geometrical insight. Next we will derive a curvature formula in terms of intermediate control points of the de Boor algorithm.

First we will prove a lemma. It is similar to the lemma in [Floa92] for rational Bezier curves.

Lemma Given the NURBS curve of degree n containing L segments defined by equations (A-5) (A-6) (A-7), and given any $k \in \{1,...,n-1\}$ and $i \in \{0,...,n-k\}$, the following expression holds:

$$P_{i+1,k} - P_{i,k} = \frac{\alpha_{i+1,k} w_{i+1,k-1}}{w_{i+1,k}} (P_{i+1,k-1} - P_{i,k-1}) + \frac{(1-\alpha_{i,k}) w_{i-1,k-1}}{w_{i,k}} (P_{i,k-1} - P_{i-1,k-1}) \tag{A-11}$$

Proof. Substitute (A-5) and (A-6) into (A-7), we get:

$$P_{i+1,k} - P_{i,k} = \frac{(1-\alpha_{i+1,k}) w_{i,k-1}}{w_{i+1,k}} P_{i,k-1} + \frac{\alpha_{i+1,k} w_{i+1,k-1}}{w_{i+1,k}} P_{i+1,k-1} - \frac{(1-\alpha_{i,k}) w_{i-1,k-1}}{w_{i,k}} P_{i-1,k-1} - \frac{\alpha_{i,k} w_{i,k-1}}{w_{i,k}} P_{i,k-1}$$

To prove (A-11) is equivalent to prove

$$\frac{(1-\alpha_{i+1,k}) w_{i,k-1}}{w_{i+1,k}} - \frac{\alpha_{i,k} w_{i,k-1}}{w_{i,k}} = \frac{(1-\alpha_{i,k}) w_{i-1,k-1}}{w_{i,k}} - \frac{\alpha_{i+1,k} w_{i+1,k-1}}{w_{i+1,k}}$$

that is, to prove:

$$(1-\alpha_{i+1,k}) w_{i,k-1} w_{i,k} - \alpha_{i,k} w_{i,k-1} w_{i+1,k} = (1-\alpha_{i,k}) w_{i-1,k-1} w_{i+1,k} - \alpha_{i+1,k} w_{i+1,k-1} w_{i,k}$$

by applying equation (A-6), the left hand side is

$$(1-\alpha_{i+1,k}) w_{i,k-1} \left[(1-\alpha_{i,k}) w_{i-1,k-1} + \alpha_{i,k} w_{i,k-1} \right] - \alpha_{i,k} w_{i,k-1} \left[(1-\alpha_{i+1,k}) w_{i,k-1} + \alpha_{i+1,k} w_{i+1,k-1} \right]$$
$$= (1-\alpha_{i+1,k})(1-\alpha_{i,k}) w_{i-1,k-1} w_{i,k-1} - \alpha_{i,k} \alpha_{i+1,k} w_{i+1,k-1} w_{i,k-1}$$

the right hand side is

$$(1-\alpha_{i,k}) w_{i-1,k-1} \left[(1-\alpha_{i+1,k}) w_{i,k-1} + \alpha_{i+1,k} w_{i+1,k-1} \right] - \alpha_{i+1,k} w_{i+1,k-1} \left[(1-\alpha_{i,k}) w_{i-1,k-1} + \alpha_{i,k} w_{i,k-1} \right]$$
$$= (1-\alpha_{i+1,k})(1-\alpha_{i,k}) w_{i-1,k-1} w_{i,k-1} - \alpha_{i,k} \alpha_{i+1,k} w_{i+1,k-1} w_{i,k-1} \quad \square$$

The expression for the curvature of $R(u)$ is defined as

$$\kappa(u) = \frac{\left\| R' \times R'' \right\|}{\left\| R' \right\|^3} \tag{A-12}$$

This can be computed by using the lemma and the first derivative formula (A-8). Rearranging (A-9), we have

$$w_{I+1,n}^2(u) R'(u) = \frac{n}{u_{I+1} - u_I} w_{I,n-1}(u) w_{I+1,n-1}(u) \left(P_{I+1,n-1}(u) - P_{I,n-1}(u) \right)$$

take derivatives on both sides of the equation

$$2 w_{I+1,n} w'_{I+1,n} R' + w_{I+1,n}^2 R''$$
$$= \frac{n}{u_{I+1} - u_I} [((w'_{I,n-1} w_{I+1,n-1} + w_{I,n-1} w'_{I+1,n-1}) \left(P_{I+1,n-1} - P_{I,n-1} \right) + w_{I,n-1} w_{I+1,n-1} (P'_{I+1,n-1} - P'_{I,n-1})] \tag{A-13}$$

By applying (A-9) we have

$$R' \times [(w'_{I,n-1} w_{I+1,n-1} + w_{I,n-1} w'_{I+1,n-1})(P_{I+1,n-1} - P_{I,n-1})] = 0$$

$$R' \times R'' = \frac{1}{w_{I+1,n}^2} \frac{n}{u_{I+1} - u_I} [w_{I,n-1} w_{I+1,n-1} R' \times (P'_{I+1,n-1} - P'_{I,n-1})]$$

$$= \frac{n^2 w_{I,n-1}^2 w_{I+1,n-1}^2}{w_{I+1,n}^4 (u_{I+1} - u_I)^2} [(P_{I+1,n-1} - P_{I,n-1}) \times (P'_{I+1,n-1} - P'_{I,n-1})]$$

Now we will focus on $(P_{I+1,n-1} - P_{I,n-1}) \times (P'_{I+1,n-1} - P'_{I,n-1})$. By applying (A-8) and lemma, we have

$$(P_{I+1,n-1} - P_{I,n-1}) \times (P'_{I+1,n-1} - P'_{I,n-1}) = A(P_{I+1,n-2} - P_{I,n-2}) \times (P_{I,n-2} - P_{I-1,n-2})$$

$$A = \frac{\alpha_{I+1,n-1} w_{I+1,n-2}}{w_{I+1,n-1}} \left[-(n-1) \frac{w_{I-1,n-2} w_{I,n-2}}{w_{I,n-1}^2 (u_{I+1} - u_{I-1})} \right] - (n-1) \frac{(1 - \alpha_{I,n-1}) w_{I-1,n-2}}{w_{I,n-1}} \frac{w_{I,n-2} w_{I+1,n-2}}{w_{I+1,n-1}^2 (u_{I+2} - u_I)}$$

$$= -(n-1) \frac{w_{I-1,n-2} w_{I,n-2} w_{I+1,n-2}}{w_{I+1,n-1} w_{I,n-1}} [\frac{\alpha_{I+1,n-1}}{w_{I,n-1} (u_{I+1} - u_{I-1})} + \frac{1 - \alpha_{I,n-1}}{w_{I+1,n-1} (u_{I+2} - u_I)}]$$

$$= -(n-1) \frac{w_{I-1,n-2} w_{I,n-2} w_{I+1,n-2}}{w_{I+1,n-1}^2 w_{I,n-1}^2} [\frac{\alpha_{I+1,n-1} w_{I+1,n-1}}{(u_{I+1} - u_{I-1})} + \frac{(1 - \alpha_{I,n-1}) w_{I,n-1}}{(u_{I+2} - u_I)}]$$

$$= -(n-1) \frac{w_{I-1,n-2} w_{I,n-2} w_{I+1,n-2}}{w_{I+1,n-1}^2 w_{I,n-1}^2} [\frac{(u - u_I) w_{I+1,n-1}}{(u_{I+2} - u_I)(u_{I+1} - u_{I-1})} + \frac{(u_{I+1} - u) w_{I,n-1}}{(u_{I+1} - u_{I-1})(u_{I+2} - u_I)}]$$

$$= -(n-1) \frac{w_{I-1,n-2} w_{I,n-2} w_{I+1,n-2}}{w_{I+1,n-1}^2 w_{I,n-1}^2} \frac{(u_{I+1} - u_I)}{(u_{I+2} - u_I)(u_{I+1} - u_{I-1})} [\frac{(u - u_I) w_{I+1,n-1}}{u_{I+1} - u_I} + \frac{(u_{I+1} - u) w_{I,n-1}}{u_{I+1} - u_I}]$$

$$= -(n-1) \frac{w_{I-1,n-2} w_{I,n-2} w_{I+1,n-2} w_{I+1,n}}{w_{I+1,n-1}^2 w_{I,n-1}^2} \frac{(u_{I+1} - u_I)}{(u_{I+2} - u_I)(u_{I+1} - u_{I-1})}$$

Then, we have the formula of curvature:

$$\kappa(u) = \frac{(n-1)(u_{I+1} - u_I)^2}{n(u_{I+2} - u_I)(u_{I+1} - u_{I-1})} \frac{w_{I-1,n-2} w_{I,n-2} w_{I+1,n-2} w_{I+1,n}^3}{w_{I,n-1}^3 w_{I+1,n-1}^3} \frac{\left\| (P_{I+1,n-2} - P_{I,n-2}) \times (P_{I,n-2} - P_{I-1,n-2}) \right\|}{\left\| P_{I+1,n-1} - P_{I,n-1} \right\|^3}$$

(A-14)

Appendix B Data

This appendix presents the surfaces used in Chapter 6 and 7. A B-Spline surface is described as:

UOrder, VOrder,

Number of UKnots, Number of VKnots

Uknots,

VKnots,

Control points X, Y, Z, W, (w=1.0 if only X, Y, Z are presented)

B.1 Data Used in Figure 6.5

The original wing surface:

4,4

9,17

0.000000,0.000000,0.000000,0.000000,2.000000,4.000000,4.000000,4.000000,4.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,12.000000,12.000000,12.000000,12.000000,

1.598275,-1.986931,0.236701,

1.463602,-1.986913,0.236908,

1.296392,-1.987592,0.229145,

1.141984,-1.989252,0.210178,

1.060431,-1.990864,0.191750,

1.012731,-1.992594,0.171969,

0.998612,-1.994276,0.152745,

1.017503,-1.995554,0.138141,

1.068829,-1.996071,0.132226,

1.152321,-1.995662,0.136904,

1.305996,-1.993548,0.161071,
1.468876,-1.990184,0.199523,
1.598275,-1.986931,0.236701,

1.590470,-1.655163,0.204252,
1.440838,-1.654870,0.207604,
1.254869,-1.655285,0.202853,
1.082860,-1.656816,0.185359,
0.991816,-1.658442,0.166773,
0.938353,-1.660268,0.145901,
0.922217,-1.662108,0.124867,
0.942866,-1.663566,0.108203,
0.999756,-1.664246,0.100441,
1.092635,-1.663960,0.103704,
1.263949,-1.661922,0.126994,
1.445825,-1.658515,0.165942,
1.590470,-1.655163,0.204252,

1.575108,-0.992240,0.132355,
1.395781,-0.991214,0.144080,
1.172442,-0.990875,0.147957,
0.965190,-0.991937,0.135827,
0.855007,-0.993477,0.118219,
0.789789,-0.995427,0.095933,
0.769341,-0.997562,0.071531,
0.793247,-0.999404,0.050478,
0.861091,-1.000475,0.038236,
0.972679,-1.000551,0.037368,
1.179399,-0.998878,0.056487,
1.399602,-0.995610,0.093845,
1.575108,-0.992240,0.132354,

1.558776,-0.330544,0.046440,
1.350633,-0.328426,0.070651,
1.090771,-0.326880,0.088314,
0.848693,-0.327049,0.086388,
0.719334,-0.328275,0.072369,
0.642061,-0.330210,0.050254,
0.616800,-0.332591,0.023041,
0.643391,-0.334859,-0.002884,
0.721674,-0.336456,-0.021134,
0.851574,-0.337119,-0.028714,
1.093447,-0.336236,-0.018618,
1.352103,-0.333563,0.011925,
1.558776,-0.330544,0.046440,

1.550000,0.000000,0.000000,
1.327778,0.002752,0.031461,
1.050000,0.005012,0.057285,
0.790741,0.005395,0.061662,
0.651852,0.004382,0.050091,
0.568519,0.002491,0.028467,
0.540741,0.000000,0.000000,
0.568519,-0.002491,-0.028467,
0.651852,-0.004382,-0.050091,
0.790741,-0.005395,-0.061662,
1.050000,-0.005012,-0.057285,
1.327778,-0.002752,-0.031461,
1.550000,-0.000000,-0.000000,

The trimmed wing surface:

4,4

9,17

0.000000,0.000000,0.000000,0.000000,2.000000,4.000000,4.000000,4.000000,4.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,12.000000,12.000000,12.000000,12.000000,

1.598275,-1.986931,0.236701,

1.463602,-1.986913,0.236908,

1.296392,-1.987592,0.229144,

1.141984,-1.989252,0.210178,

1.060431,-1.990863,0.191750,

1.012731,-1.992595,0.171969,

0.998612,-1.994276,0.152745,

1.017503,-1.995554,0.138141,

1.068829,-1.996071,0.132226,

1.152321,-1.995662,0.136904,

1.305996,-1.993547,0.161071,

1.468876,-1.990184,0.199523,

1.598275,-1.986931,0.236701,

1.593219,-1.772030,0.215682,

1.448801,-1.769596,0.217710,

1.268931,-1.767257,0.211705,

1.102356,-1.766342,0.193536,

1.014182,-1.766770,0.174913,

0.962438,-1.767872,0.154342,

0.946890,-1.769389,0.133871,

0.967001,-1.770915,0.117883,

1.022229,-1.772174,0.110777,

1.112300,-1.773146,0.114630,
1.278214,-1.773809,0.138489,
1.453904,-1.773368,0.177576,
1.593219,-1.772030,0.215682,

1.583212,-1.342485,0.170708,
1.419400,-1.335144,0.177190,
1.214282,-1.326682,0.175796,
1.023361,-1.320519,0.160295,
0.921877,-1.318526,0.141874,
0.861939,-1.318344,0.120080,
0.843407,-1.319516,0.097232,
0.865838,-1.321551,0.078351,
0.928793,-1.324325,0.068501,
1.031996,-1.328113,0.070109,
1.222450,-1.334424,0.092277,
1.423911,-1.339923,0.131557,
1.583212,-1.342485,0.170708,

1.573014,-0.913454,0.119855,
1.390105,-0.901068,0.132404,
1.160033,-0.886287,0.137792,
0.944890,-0.874692,0.127107,
0.830045,-0.870173,0.110076,
0.761756,-0.868642,0.087786,
0.740001,-0.869449,0.062814,
0.764501,-0.872013,0.040807,
0.834971,-0.876366,0.027497,
0.951172,-0.883071,0.025677,
1.166142,-0.895218,0.044002,
1.393528,-0.906847,0.081298,

1.573014,-0.913454,0.119855,

1.567768,-0.699067,0.092959,
1.375412,-0.684123,0.108940,
1.132973,-0.666137,0.118258,
0.905787,-0.651778,0.110518,
0.784269,-0.645971,0.094479,
0.711778,-0.643749,0.072126,
0.688358,-0.644367,0.046160,
0.713823,-0.647200,0.022537,
0.787986,-0.652357,0.007321,
0.910630,-0.660548,0.003492,
1.137817,-0.675660,0.019357,
1.378166,-0.690402,0.055113,
1.567768,-0.699067,0.092959,

B.2 Data Used in Figure 6.6

The original fuselage surface:

4,4

14,12

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,9.000000,9.000000,9.000000,9.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,7.000000,7.000000,7.000000,7.000000,

0.000000,-0.000000,-0.600000,
0.000000,-0.180744,-0.602621,
0.000000,-0.445283,-0.470036,
0.000000,-0.604910,-0.138015,
0.000000,-0.604910,0.138015,
0.000000,-0.445283,0.470036,

0.000000,-0.180744,0.602621,
0.000000,0.000000,0.600000,

0.222222,-0.000000,-0.622222,
0.222222,-0.185207,-0.624941,
0.222222,-0.456278,-0.487444,
0.222222,-0.619846,-0.143127,
0.222222,-0.619846,0.143127,
0.222222,-0.456278,0.487444,
0.222222,-0.185207,0.624941,
0.222222,0.000000,0.622222,

0.555555,-0.000000,-0.655555,
0.555556,-0.191901,-0.658419,
0.555556,-0.472769,-0.513558,
0.555555,-0.642250,-0.150794,
0.555556,-0.642250,0.150794,
0.555555,-0.472770,0.513558,
0.555555,-0.191901,0.658419,
0.555555,0.000000,0.655555,

1.000000,-0.000000,-0.700000,
1.000000,-0.200827,-0.703059,
1.000000,-0.494759,-0.548375,
1.000000,-0.672122,-0.161018,
1.000000,-0.672122,0.161017,
1.000000,-0.494759,0.548375,
1.000000,-0.200827,0.703058,
1.000000,0.000000,0.700000,

1.333333,-0.000000,-0.733333,

1.333334,-0.207521,-0.736537,
1.333333,-0.511251,-0.574488,
1.333333,-0.694526,-0.168685,
1.333333,-0.694526,0.168685,
1.333333,-0.511251,0.574488,
1.333333,-0.207521,0.736537,
1.333333,0.000000,0.733333,

1.666666,-0.000000,-0.766667,
1.666666,-0.214215,-0.770016,
1.666666,-0.527743,-0.600601,
1.666667,-0.716931,-0.176352,
1.666667,-0.716931,0.176352,
1.666666,-0.527743,0.600601,
1.666666,-0.214215,0.770016,
1.666666,0.000000,0.766667,

2.000000,-0.000000,-0.800000,
2.000001,-0.220910,-0.803495,
2.000000,-0.544235,-0.626714,
2.000000,-0.739335,-0.184020,
2.000000,-0.739335,0.184020,
2.000000,-0.544235,0.626714,
2.000000,-0.220910,0.803495,
2.000000,0.000000,0.800000,

2.444444,-0.000000,-0.844444,
2.444444,-0.229835,-0.848134,
2.444444,-0.566224,-0.661532,
2.444444,-0.769207,-0.194243,
2.444444,-0.769207,0.194243,

2.444444,-0.566224,0.661532,

2.444444,-0.229835,0.848134,

2.444444,0.000000,0.844444,

2.777778,-0.000000,-0.877778,

2.777779,-0.236530,-0.881613,

2.777778,-0.582716,-0.687645,

2.777777,-0.791611,-0.201911,

2.777779,-0.791611,0.201911,

2.777777,-0.582716,0.687645,

2.777778,-0.236529,0.881613,

2.777778,0.000000,0.877778,

3.000000,-0.000000,-0.900000,

3.000000,-0.240992,-0.903932,

3.000000,-0.593711,-0.705054,

3.000000,-0.806547,-0.207022,

3.000000,-0.806547,0.207022,

3.000000,-0.593711,0.705054,

3.000000,-0.240992,0.903932,

3.000000,0.000000,0.900000,

The trimmed fuselage surface region B:

4,4

11,10

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,6.000000,6.000000,6.000000,6.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,5.000000,5.000000,5.000000,5.000000,

0.696501,-0.000000,-0.669650,

0.696502,-0.142280,-0.671703,

0.696501,-0.352229,-0.593748,
0.696501,-0.597227,-0.323508,
0.696501,-0.655506,-0.099573,
0.696501,-0.644799,0.046555,

0.697181,-0.000000,-0.669718,
0.697181,-0.140106,-0.671757,
0.697181,-0.346848,-0.596203,
0.697181,-0.591335,-0.333466,
0.697181,-0.652910,-0.114497,
0.697181,-0.645992,0.029653,

0.747100,-0.000000,-0.674710,
0.747100,-0.138088,-0.676743,
0.747100,-0.341847,-0.603563,
0.747100,-0.586566,-0.348141,
0.747100,-0.652163,-0.133906,
0.747100,-0.649701,0.008566,

0.910626,-0.000000,-0.691062,
0.910626,-0.139737,-0.693140,
0.910626,-0.345930,-0.618906,
0.910626,-0.594476,-0.359572,
0.910626,-0.662044,-0.141728,
0.910626,-0.660612,0.003499,

1.137818,-0.000000,-0.713782,
1.137818,-0.144941,-0.715943,
1.137818,-0.358814,-0.637113,
1.137817,-0.613865,-0.362437,
1.137818,-0.680340,-0.132715,

1.137818,-0.675643,0.019353,

1.378160,-0.000000,-0.737816,
1.378159,-0.152784,-0.740082,
1.378160,-0.378240,-0.653713,
1.378159,-0.640798,-0.354395,
1.378160,-0.702674,-0.106488,
1.378160,-0.690492,0.055119,

1.567764,-0.000000,-0.756776,
1.567764,-0.160100,-0.759132,
1.567764,-0.396336,-0.665175,
1.567764,-0.664234,-0.341588,
1.567764,-0.719759,-0.076686,
1.567764,-0.699133,0.092966,

The trimmed fuselage surface region C:

4,4

11,10

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,6.000000,6.000000,6.000000,6.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,5.000000,5.000000,5.000000,5.000000,

0.696501,-0.644799,0.046555,
0.696501,-0.636317,0.180329,
0.696501,-0.562432,0.376310,
0.696501,-0.322514,0.606355,
0.696501,-0.130275,0.671605,
0.696501,0.000000,0.669650,

0.695792,-0.643675,0.063479,

0.695792,-0.631994,0.194773,
0.695792,-0.555680,0.385566,
0.695792,-0.317079,0.608493,
0.695792,-0.128076,0.671512,
0.695792,0.000000,0.669579,

0.743762,-0.644006,0.089162,
0.743763,-0.627853,0.217495,
0.743762,-0.547966,0.401863,
0.743762,-0.310557,0.616044,
0.743762,-0.125428,0.676284,
0.743762,0.000000,0.674376,

0.905782,-0.651861,0.110524,
0.905782,-0.632318,0.238848,
0.905781,-0.548933,0.421666,
0.905782,-0.309599,0.633228,
0.905782,-0.125027,0.692498,
0.905782,0.000000,0.690578,

1.132972,-0.666145,0.118263,
1.132972,-0.645515,0.250137,
1.132973,-0.559781,0.437690,
1.132972,-0.315406,0.654566,
1.132973,-0.127369,0.715273,
1.132972,0.000000,0.713297,

1.375406,-0.684220,0.108952,
1.375407,-0.665193,0.247483,
1.375405,-0.578842,0.445567,
1.375406,-0.327170,0.675171,

1.375406,-0.132130,0.739608,

1.375406,0.000000,0.737541,

1.567764,-0.699133,0.092966,

1.567764,-0.682856,0.238056,

1.567764,-0.597127,0.447133,

1.567764,-0.339031,0.690411,

1.567764,-0.136931,0.758928,

1.567764,0.000000,0.756776,

B.3 Data Used in Table 6.3 with 25 points on the trimming curve

The original surfaces are given in section B.1 and B.2.

The trimmed wing surface:

4,4

9,29

0.000000,0.000000,0.000000,0.000000,2.000000,4.000000,4.000000,4.000000,4.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,11.000000,12.000000,13.000000,14.000000,15.000000,16.000000,17.000000,18.000000,19.000000,20.000000,21.000000,22.000000,24.000000,24.000000,24.000000,24.000000,

1.598275,-1.986931,0.236701,

1.530939,-1.986922,0.236805,

1.438633,-1.987085,0.234941,

1.332828,-1.987573,0.229362,

1.262074,-1.988054,0.223865,

1.199887,-1.988629,0.217291,

1.146265,-1.989298,0.209653,

1.101208,-1.990058,0.200964,

1.064663,-1.990878,0.191581,

1.036581,-1.991729,0.181859,

1.016928,-1.992588,0.172039,
1.005671,-1.993435,0.162357,
1.002738,-1.994226,0.153322,
1.008058,-1.994915,0.145443,
1.021557,-1.995459,0.139227,
1.043166,-1.995812,0.135184,
1.072850,-1.995955,0.133550,
1.110575,-1.995867,0.134565,
1.156291,-1.995515,0.138585,
1.209949,-1.994869,0.145967,
1.271546,-1.993928,0.156723,
1.341080,-1.992691,0.170868,
1.444331,-1.990618,0.194557,
1.533575,-1.988557,0.218112,
1.598275,-1.986931,0.236701,

1.593219,-1.772030,0.215682,
1.521014,-1.770843,0.216698,
1.421885,-1.769311,0.215955,
1.308066,-1.767919,0.211435,
1.231865,-1.767166,0.206528,
1.164822,-1.766688,0.200349,
1.106951,-1.766482,0.192911,
1.058268,-1.766550,0.184224,
1.018737,-1.766858,0.174670,
0.988307,-1.767310,0.164627,
0.966963,-1.767926,0.154355,
0.954665,-1.768638,0.144107,
0.951346,-1.769385,0.134431,
0.956947,-1.770160,0.125878,
0.971390,-1.770878,0.118993,

0.994616,-1.771549,0.114331,
1.026584,-1.772137,0.112147,
1.067265,-1.772662,0.112704,
1.116596,-1.773089,0.116386,
1.174516,-1.773386,0.123576,
1.241008,-1.773570,0.134291,
1.316037,-1.773566,0.148534,
1.427397,-1.773316,0.172568,
1.523559,-1.772715,0.196631,
1.593219,-1.772030,0.215682,

1.583212,-1.342485,0.170708,
1.501314,-1.338906,0.173955,
1.388597,-1.333934,0.176030,
1.258786,-1.328722,0.174323,
1.171713,-1.325462,0.171068,
1.094957,-1.322837,0.166108,
1.028576,-1.320850,0.159445,
0.972614,-1.319505,0.151084,
0.927073,-1.318765,0.141453,
0.891901,-1.318400,0.130974,
0.867115,-1.318516,0.119948,
0.852678,-1.318953,0.108658,
0.848525,-1.319612,0.097727,
0.854628,-1.320558,0.087794,
0.870902,-1.321633,0.079479,
0.897318,-1.322954,0.073427,
0.933823,-1.324447,0.069933,
0.980395,-1.326225,0.069305,
1.036952,-1.328236,0.071987,
1.103411,-1.330451,0.078419,

1.179723,-1.332928,0.088625,
1.265794,-1.335425,0.102594,
1.393460,-1.338884,0.126633,
1.503557,-1.341250,0.151140,
1.583212,-1.342485,0.170708,

1.573014,-0.913454,0.119855,
1.481574,-0.907414,0.126140,
1.355450,-0.898901,0.132180,
1.209827,-0.889747,0.134661,
1.111974,-0.883897,0.134003,
1.025570,-0.879052,0.131115,
0.950705,-0.875219,0.125980,
0.887460,-0.872403,0.118590,
0.835869,-0.870569,0.109420,
0.795887,-0.869349,0.098927,
0.767574,-0.868939,0.087451,
0.750887,-0.869085,0.075303,
0.745779,-0.869648,0.063183,
0.752259,-0.870772,0.051814,
0.770244,-0.872220,0.041893,
0.799740,-0.874214,0.034155,
0.840686,-0.876651,0.028933,
0.893071,-0.879728,0.026588,
0.956802,-0.883381,0.027623,
1.031777,-0.887581,0.032544,
1.117916,-0.892420,0.041387,
1.215061,-0.897507,0.054134,
1.359122,-0.904791,0.076796,
1.483263,-0.910229,0.100591,
1.573014,-0.913454,0.119855,

1.567768,-0.699067,0.092959,
1.471607,-0.691779,0.100963,
1.338846,-0.681470,0.109269,
1.185383,-0.670317,0.114185,
1.082182,-0.663150,0.115062,
0.990983,-0.657176,0.113422,
0.911898,-0.652404,0.109239,
0.845019,-0.648839,0.102497,
0.790403,-0.646445,0.093691,
0.748008,-0.644790,0.083298,
0.717914,-0.644109,0.071675,
0.700078,-0.644105,0.059147,
0.694464,-0.644620,0.046451,
0.701101,-0.645833,0.034352,
0.719906,-0.647471,0.023586,
0.750909,-0.649809,0.014932,
0.794045,-0.652726,0.008744,
0.849308,-0.656464,0.005407,
0.916601,-0.660951,0.005459,
0.995815,-0.666160,0.009436,
1.086854,-0.672201,0.017383,
1.189528,-0.678602,0.029281,
1.341785,-0.687831,0.050908,
1.472957,-0.694828,0.074054,
1.567768,-0.699067,0.092959,

The trimmed fuselage surface B

4,4
17,10

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,12.000000,12.000000,12.000000,12.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,5.000000,5.000000,5.000000,5.000000,

0.696501,-0.000000,-0.669650,

0.696502,-0.142280,-0.671703,

0.696501,-0.352229,-0.593748,

0.696501,-0.597227,-0.323508,

0.696501,-0.655506,-0.099573,

0.696501,-0.644799,0.046555,

0.696860,-0.000000,-0.669686,

0.696860,-0.141218,-0.671732,

0.696860,-0.349601,-0.594951,

0.696860,-0.594348,-0.328377,

0.696860,-0.654237,-0.106869,

0.696860,-0.645380,0.038292,

0.709568,-0.000000,-0.670957,

0.709568,-0.139871,-0.672996,

0.709568,-0.346264,-0.597738,

0.709568,-0.590872,-0.335901,

0.709568,-0.653044,-0.117501,

0.709568,-0.646772,0.026474,

0.750908,-0.000000,-0.675091,

0.750908,-0.138963,-0.677131,

0.750908,-0.344015,-0.603032,

0.750907,-0.589134,-0.344695,

0.750908,-0.653643,-0.128440,

0.750908,-0.649831,0.014932,

0.794040,-0.000000,-0.679404,
0.794040,-0.138772,-0.681451,
0.794040,-0.343541,-0.607744,
0.794040,-0.589438,-0.350487,
0.794040,-0.655315,-0.134729,
0.794040,-0.652803,0.008745,

0.849305,-0.000000,-0.684931,
0.849305,-0.139119,-0.686991,
0.849305,-0.344401,-0.613151,
0.849305,-0.591510,-0.355279,
0.849305,-0.658336,-0.138786,
0.849305,-0.656515,0.005408,

0.916598,-0.000000,-0.691660,
0.916597,-0.140070,-0.693741,
0.916598,-0.346755,-0.619175,
0.916597,-0.595553,-0.358770,
0.916598,-0.662835,-0.140151,
0.916598,-0.661000,0.005459,

0.995809,-0.000000,-0.699581,
0.995809,-0.141687,-0.701690,
0.995809,-0.350758,-0.625733,
0.995809,-0.601743,-0.360650,
0.995809,-0.668903,-0.138354,
0.995809,-0.666247,0.009437,

1.086855,-0.000000,-0.708685,
1.086855,-0.143969,-0.710829,

1.086855,-0.356408,-0.632807,
1.086854,-0.610054,-0.360872,
1.086855,-0.676489,-0.133345,
1.086855,-0.672189,0.017383,

1.189522,-0.000000,-0.718952,
1.189522,-0.146904,-0.721138,
1.189522,-0.363676,-0.640368,
1.189522,-0.620409,-0.359395,
1.189522,-0.685480,-0.125101,
1.189522,-0.678693,0.029284,

1.341783,-0.000000,-0.734178,
1.341782,-0.151739,-0.736429,
1.341783,-0.375646,-0.650996,
1.341782,-0.636995,-0.354803,
1.341783,-0.699226,-0.109317,
1.341783,-0.687858,0.050911,

1.472953,-0.000000,-0.747295,
1.472954,-0.156443,-0.749606,
1.472953,-0.387291,-0.659451,
1.472954,-0.652543,-0.348010,
1.472953,-0.711274,-0.091596,
1.472953,-0.694880,0.074058,

1.567764,-0.000000,-0.756776,
1.567764,-0.160100,-0.759132,
1.567764,-0.396336,-0.665175,
1.567764,-0.664234,-0.341588,
1.567764,-0.719759,-0.076686,

1.567764,-0.699133,0.092966,

The trimmed fuselage surface region C:

4,4

17,10

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,12.000000,12.000000,12.000000,12.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,5.000000,5.000000,5.000000,5.000000,

0.696501,-0.644799,0.046555,

0.696501,-0.636317,0.180329,

0.696501,-0.562432,0.376310,

0.696501,-0.322514,0.606355,

0.696501,-0.130275,0.671605,

0.696501,0.000000,0.669650,

0.696171,-0.644232,0.054822,

0.696171,-0.634194,0.187382,

0.696171,-0.559129,0.380829,

0.696171,-0.319860,0.607399,

0.696171,-0.129201,0.671562,

0.696171,0.000000,0.669617,

0.707882,-0.643899,0.067801,

0.707882,-0.631496,0.198663,

0.707882,-0.554590,0.388493,

0.707882,-0.316123,0.610116,

0.707882,-0.127687,0.672718,

0.707882,0.000000,0.670788,

0.748002,-0.644877,0.083309,
0.748002,-0.629783,0.212622,
0.748002,-0.550643,0.398926,
0.748002,-0.312599,0.615678,
0.748002,-0.126255,0.676718,
0.748002,0.000000,0.674800,

0.790404,-0.646433,0.093690,
0.790404,-0.629596,0.222268,
0.790404,-0.548915,0.406689,
0.790404,-0.310804,0.620784,
0.790404,-0.125524,0.680954,
0.790404,0.000000,0.679040,

0.845014,-0.648913,0.102508,
0.845013,-0.630643,0.230835,
0.845013,-0.548571,0.414235,
0.845013,-0.309959,0.626782,
0.845014,-0.125178,0.686417,
0.845014,0.000000,0.684501,

0.911894,-0.652464,0.109248,
0.911894,-0.633137,0.237905,
0.911893,-0.549860,0.421312,
0.911894,-0.310233,0.633618,
0.911894,-0.125284,0.693114,
0.911894,0.000000,0.691189,

0.990980,-0.657216,0.113428,
0.990980,-0.637265,0.243082,
0.990980,-0.553001,0.427675,

0.990980,-0.311776,0.641230,
0.990980,-0.125905,0.701039,
0.990980,0.000000,0.699098,

1.082179,-0.663191,0.115068,
1.082179,-0.643033,0.246388,
1.082179,-0.557981,0.433339,
1.082179,-0.314573,0.649615,
1.082179,-0.127034,0.710184,
1.082179,0.000000,0.708218,

1.185380,-0.670374,0.114193,
1.185380,-0.650415,0.247840,
1.185379,-0.564773,0.438311,
1.185380,-0.318600,0.658766,
1.185379,-0.128662,0.720537,
1.185380,0.000000,0.718538,

1.338840,-0.681564,0.109282,
1.338841,-0.662480,0.246968,
1.338840,-0.576362,0.443793,
1.338840,-0.325712,0.671924,
1.338840,-0.131540,0.735939,
1.338840,0.000000,0.733884,

1.471611,-0.691716,0.100955,
1.471611,-0.674056,0.242773,
1.471611,-0.588006,0.446359,
1.471610,-0.333107,0.682795,
1.471611,-0.134533,0.749270,
1.471611,0.000000,0.747161,

1.567764,-0.699133,0.092966,

1.567764,-0.682856,0.238056,

1.567764,-0.597127,0.447133,

1.567764,-0.339031,0.690411,

1.567764,-0.136931,0.758928,

1.567764,0.000000,0.756776,

B.4 Data Used in Figure 6.14

The original wing surface:

4,4

9,17

0.000000,0.000000,0.000000,0.000000,2.000039,4.000207,4.000207,4.000207,4.000207,

0.000000,0.000000,0.000000,0.000000,1.866621,2.656655,3.322027,3.841158,4.254750,4.668343,5.187473,5.852845,6.642879,8.509500,8.509500,8.509500,8.509500,

1.598275,-1.986931,0.236701,

1.478421,-1.986896,0.237107,

1.319581,-1.987453,0.230733,

1.156186,-1.989031,0.212699,

1.073500,-1.990621,0.194528,

1.023010,-1.992007,0.178683,

0.994353,-1.994328,0.152148,

1.029242,-1.995871,0.134511,

1.082160,-1.995991,0.133144,

1.166671,-1.995533,0.138381,

1.328726,-1.993124,0.165912,

1.483176,-1.989845,0.203400,

1.598275,-1.986931,0.236701,

1.590469,-1.655153,0.204251,
1.457308,-1.654871,0.207481,
1.280670,-1.655169,0.204081,
1.098698,-1.656590,0.187829,
1.006400,-1.658189,0.169556,
0.949930,-1.659627,0.153121,
0.917468,-1.662148,0.124302,
0.955822,-1.663933,0.103897,
1.014588,-1.664175,0.101151,
1.108612,-1.663836,0.105011,
1.289317,-1.661489,0.131844,
1.461805,-1.658157,0.169918,
1.590469,-1.655154,0.204251,

1.575107,-0.992200,0.132350,
1.415531,-0.991261,0.143081,
1.203454,-0.990822,0.148098,
0.984313,-0.991708,0.137973,
0.872641,-0.993210,0.120804,
0.804045,-0.994681,0.103996,
0.763608,-0.997559,0.071096,
0.808560,-0.999874,0.044642,
0.878914,-1.000425,0.038322,
0.991908,-1.000445,0.038111,
1.210079,-0.998442,0.060996,
1.418975,-0.995225,0.097788,
1.575107,-0.992200,0.132350,

1.558775,-0.330523,0.046437,
1.373576,-0.328608,0.068328,
1.126898,-0.327005,0.086654,

0.871094,-0.326908,0.087763,
0.740028,-0.328082,0.074343,
0.659148,-0.329441,0.058807,
0.610105,-0.332585,0.022874,
0.660884,-0.335511,-0.010580,
0.742442,-0.336518,-0.022082,
0.874016,-0.337121,-0.028978,
1.129447,-0.335914,-0.015171,
1.374901,-0.333240,0.015379,
1.558775,-0.330523,0.046437,

1.550000,0.000000,0.000000,
1.352283,0.002482,0.028365,
1.088642,0.004772,0.054549,
0.814768,0.005472,0.062540,
0.674068,0.004519,0.051656,
0.587048,0.003252,0.037171,
0.533572,0.000000,0.000000,
0.587048,-0.003252,-0.037171,
0.674068,-0.004519,-0.051656,
0.814768,-0.005472,-0.062540,
1.088642,-0.004772,-0.054549,
1.352283,-0.002482,-0.028365,
1.550000,-0.000000,-0.000000,

The wing surface trimmed by not knowing the interpolation of the original surface:

4,4
9,17

0.000000,0.000000,0.000000,0.000000,2.000013,4.000071,4.000071,4.000071,4.000071,

0.000000,0.000000,0.000000,0.000000,1.953999,2.877143,3.743359,4.516473,5.213567,5.910782,6.684058,7.550644,8.474067,10.428185,10.428185,10.428185,10.428185,

1.598275,-1.986931,0.236701,

1.510783,-1.986876,0.237332,

1.384158,-1.987256,0.232987,

1.232852,-1.988186,0.222358,

1.132067,-1.989571,0.206528,

1.058775,-1.990562,0.195204,

0.975953,-1.994553,0.149580,

1.068237,-1.996403,0.128439,

1.141780,-1.995593,0.137697,

1.243028,-1.994504,0.150144,

1.391388,-1.991763,0.181470,

1.514337,-1.989087,0.212066,

1.598275,-1.986931,0.236701,

1.593218,-1.772022,0.215682,

1.499411,-1.770414,0.217532,

1.363352,-1.768489,0.214603,

1.200405,-1.766773,0.205321,

1.091610,-1.766517,0.189754,

1.012455,-1.766403,0.178652,

0.922412,-1.769361,0.130800,

1.021521,-1.772451,0.106705,

1.100954,-1.772912,0.115633,

1.210216,-1.773603,0.127608,

1.370345,-1.773554,0.159240,

1.502843,-1.772965,0.190411,

1.593219,-1.772022,0.215682,

1.583211,-1.342461,0.170706,
1.476833,-1.337700,0.175522,
1.321974,-1.331094,0.176228,
1.135784,-1.324002,0.170658,
1.010936,-1.320399,0.156325,
0.920034,-1.318031,0.146190,
0.815284,-1.318866,0.094513,
0.927825,-1.324495,0.063865,
1.019053,-1.327541,0.071607,
1.144358,-1.331856,0.081928,
1.328143,-1.337276,0.113162,
1.479854,-1.340935,0.144689,
1.583211,-1.342462,0.170706,

1.573014,-0.913444,0.119854,
1.454268,-0.905439,0.128682,
1.280842,-0.894014,0.134615,
1.071634,-0.881366,0.134781,
0.930766,-0.874294,0.123099,
0.828145,-0.869581,0.114983,
0.708255,-0.868179,0.060780,
0.833696,-0.876455,0.022318,
0.936675,-0.882182,0.027818,
1.077957,-0.890243,0.035070,
1.285482,-0.901311,0.063875,
1.456542,-0.909355,0.094157,
1.573014,-0.913444,0.119854,

1.567769,-0.699073,0.092960,
1.442906,-0.689424,0.104052,

1.260282,-0.675554,0.112993,

1.039656,-0.660085,0.116531,

0.890811,-0.651248,0.106529,

0.782357,-0.645338,0.099685,

0.654811,-0.642790,0.044552,

0.786547,-0.652416,0.001876,

0.895370,-0.659506,0.005992,

1.044600,-0.669470,0.011356,

1.263982,-0.683406,0.038436,

1.444723,-0.693678,0.067692,

1.567769,-0.699073,0.092960,

The wing surface trimmed by knowing the interpolation of the original surface:

4,4

9,17

0.000000,0.000000,0.000000,0.000000,2.000010,4.000049,4.000049,4.000049,4.000049,

0.000000,0.000000,0.000000,0.000000,1.869033,2.661856,3.330223,3.851776,4.267088,4.682408,5.203986,5.872609,6.665743,8.535005,8.535005,8.535005,8.535005,

1.598275,-1.986931,0.236701,

1.478103,-1.986895,0.237106,

1.319234,-1.987456,0.230701,

1.156117,-1.989032,0.212688,

1.073474,-1.990622,0.194517,

1.023054,-1.992006,0.178701,

0.994332,-1.994328,0.152156,

1.029303,-1.995874,0.134490,

1.082157,-1.995990,0.133155,

1.166597,-1.995534,0.138370,

1.328334,-1.993130,0.165840,

1.482843,-1.989854,0.203305,

1.598275,-1.986931,0.236701,

1.593218,-1.772022,0.215682,
1.464362,-1.769832,0.217725,
1.293535,-1.767518,0.213053,
1.117619,-1.766338,0.196028,
1.028281,-1.766703,0.177696,
0.973677,-1.767408,0.161430,
0.942264,-1.769375,0.133300,
0.979656,-1.771389,0.113782,
1.036611,-1.772299,0.111587,
1.127705,-1.773263,0.116007,
1.302332,-1.773798,0.143308,
1.468942,-1.773252,0.181443,
1.593219,-1.772022,0.215682,

1.583212,-1.342465,0.170706,
1.437073,-1.335900,0.176752,
1.242403,-1.327749,0.176572,
1.040892,-1.320957,0.162654,
0.938096,-1.318821,0.144595,
0.875054,-1.318134,0.127801,
0.838090,-1.319375,0.096729,
0.880171,-1.322343,0.073268,
0.945280,-1.324877,0.068976,
1.049659,-1.328729,0.071207,
1.250143,-1.335243,0.097045,
1.441112,-1.340247,0.135506,
1.583211,-1.342465,0.170706,

1.573014,-0.913446,0.119854,

1.409859,-0.902383,0.131338,
1.191643,-0.888214,0.137690,
0.964691,-0.875616,0.129134,
0.848398,-0.870873,0.112545,
0.776820,-0.868731,0.095977,
0.734003,-0.869201,0.062443,
0.780434,-0.873164,0.034587,
0.853551,-0.877383,0.027450,
0.971085,-0.884232,0.026298,
1.197431,-0.896919,0.048414,
1.412906,-0.907654,0.085155,
1.573014,-0.913446,0.119854,

1.567769,-0.699073,0.092960,
1.396198,-0.685729,0.107518,
1.166312,-0.668509,0.117641,
0.926722,-0.652957,0.112329,
0.803688,-0.646886,0.096776,
0.727834,-0.643998,0.080511,
0.682022,-0.644073,0.045870,
0.730536,-0.648542,0.015709,
0.807606,-0.653622,0.006966,
0.931663,-0.661994,0.003825,
1.170900,-0.677817,0.023514,
1.398638,-0.691460,0.058880,
1.567769,-0.699073,0.092960,

B.5 Data Used in Figure 6.17

The original fuselage surface is given in section B2.

The trimmed fuselage upper surface:

4,4

25,10

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,11.000000,12.000000,13.000000,14.000000,15.000000,16.000000,17.000000,18.000000,20.000000,20.000000,20.000000,20.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,5.000000,5.000000,5.000000,5.000000,

0.000000,-0.000000,-0.600000,

-0.000000,-0.132060,-0.601840,

0.000000,-0.326929,-0.531993,

-0.000000,-0.554328,-0.289860,

-0.000000,-0.608421,-0.089217,

0.000000,-0.598483,0.041713,

0.046433,-0.000000,-0.604643,

0.046433,-0.132741,-0.606497,

0.046433,-0.328615,-0.536110,

0.046433,-0.557188,-0.292103,

0.046433,-0.611560,-0.089907,

0.046433,-0.601571,0.042035,

0.116085,-0.000000,-0.611608,

0.116085,-0.133763,-0.613483,

0.116085,-0.331146,-0.542285,

0.116085,-0.561478,-0.295468,

0.116085,-0.616268,-0.090942,

0.116085,-0.606202,0.042520,

0.208949,-0.000000,-0.620895,

0.208949,-0.135126,-0.622799,

0.208949,-0.334519,-0.550519,

0.208949,-0.567197,-0.299955,
0.208949,-0.622546,-0.092324,
0.208949,-0.612378,0.043164,

0.278606,-0.000000,-0.627861,
0.278606,-0.136148,-0.629786,
0.278606,-0.337050,-0.556695,
0.278606,-0.571489,-0.303318,
0.278607,-0.627256,-0.093357,
0.278606,-0.617010,0.043652,

0.348229,-0.000000,-0.634823,
0.348229,-0.137168,-0.636769,
0.348229,-0.339575,-0.562870,
0.348229,-0.575773,-0.306688,
0.348229,-0.631960,-0.094403,
0.348229,-0.621640,0.044124,

0.417982,-0.000000,-0.641798,
0.417982,-0.138198,-0.643766,
0.417982,-0.342124,-0.569048,
0.417982,-0.580085,-0.310032,
0.417982,-0.636683,-0.095400,
0.417982,-0.626276,0.044654,

0.487248,-0.000000,-0.648725,
0.487249,-0.139192,-0.650714,
0.487248,-0.344585,-0.575214,
0.487248,-0.584290,-0.313477,
0.487248,-0.641338,-0.096580,
0.487248,-0.630892,0.044965,

0.558331,-0.000000,-0.655833,
0.558331,-0.140317,-0.657844,
0.558331,-0.347371,-0.581427,
0.558331,-0.588895,-0.316540,
0.558331,-0.646248,-0.097077,
0.558331,-0.635583,0.046097,

0.622636,-0.000000,-0.662264,
0.622636,-0.140954,-0.664292,
0.622636,-0.348947,-0.587462,
0.622636,-0.592012,-0.321033,
0.622636,-0.650202,-0.100124,
0.622636,-0.639991,0.044166,

0.712232,-0.000000,-0.671223,
0.712232,-0.143413,-0.673287,
0.712232,-0.355036,-0.594160,
0.712232,-0.600679,-0.320185,
0.712232,-0.657727,-0.093653,
0.712232,-0.645458,0.053664,

0.707444,-0.000000,-0.670744,
0.707444,-0.139072,-0.672778,
0.707444,-0.344286,-0.598387,
0.707444,-0.588633,-0.339274,
0.707444,-0.651924,-0.122706,
0.707444,-0.646970,0.020507,

0.789606,-0.000000,-0.678961,
0.789606,-0.138606,-0.681006,

0.789606,-0.343130,-0.607451,
0.789606,-0.588862,-0.350700,
0.789606,-0.654834,-0.135322,
0.789606,-0.652483,0.007947,

0.910504,-0.000000,-0.691050,
0.910504,-0.139675,-0.693127,
0.910504,-0.345777,-0.618964,
0.910504,-0.594313,-0.359851,
0.910504,-0.661979,-0.142150,
0.910504,-0.660656,0.003024,

1.079880,-0.000000,-0.707988,
1.079880,-0.143781,-0.710129,
1.079881,-0.355940,-0.632275,
1.079880,-0.609350,-0.360904,
1.079881,-0.675826,-0.133824,
1.079880,-0.671657,0.016640,

1.302723,-0.000000,-0.730272,
1.302723,-0.149669,-0.732501,
1.302724,-0.370527,-0.649269,
1.302723,-0.630653,-0.360065,
1.302724,-0.695056,-0.119389,
1.302723,-0.686412,0.038668,

1.542848,-0.000000,-0.754285,
1.542847,-0.161441,-0.756644,
1.542848,-0.399652,-0.661006,
1.542848,-0.667200,-0.332310,
1.542848,-0.719843,-0.064276,

1.542848,-0.696195,0.106333,

1.932467,-0.000000,-0.793247,
1.932468,-0.165167,-0.795712,
1.932467,-0.408880,-0.697759,
1.932467,-0.685949,-0.360224,
1.932468,-0.744123,-0.083624,
1.932467,-0.723605,0.093801,

2.398805,-0.000000,-0.839880,
2.398805,-0.172996,-0.842497,
2.398805,-0.428259,-0.737868,
2.398804,-0.717274,-0.377649,
2.398806,-0.776677,-0.082946,
2.398805,-0.753883,0.105606,

2.763509,-0.000000,-0.876351,
2.763509,-0.178062,-0.879078,
2.763510,-0.440804,-0.770452,
2.763509,-0.738989,-0.396286,
2.763509,-0.801041,-0.089884,
2.763509,-0.778355,0.106440,

3.000000,-0.000000,-0.900000,
3.000000,-0.181798,-0.902801,
3.000000,-0.450051,-0.791063,
3.000000,-0.754257,-0.406235,
3.000000,-0.817307,-0.091200,
3.000000,-0.793886,0.110561,

The trimmed fuselage lower surface:

4,4

25,10

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,4.000000,5.000000,6.000000,7.000000,8.000000,9.000000,10.000000,11.000000,12.000000,13.000000,14.000000,15.000000,16.000000,17.000000,18.000000,20.000000,20.000000,20.000000,20.000000,

0.000000,0.000000,0.000000,0.000000,2.000000,3.000000,5.000000,5.000000,5.000000,5.000000,

0.000000,-0.598483,0.041713,

-0.000000,-0.590610,0.161573,

0.000000,-0.522032,0.337170,

-0.000000,-0.299348,0.543289,

-0.000000,-0.120917,0.601752,

0.000000,0.000000,0.600000,

0.046433,-0.601571,0.042036,

0.046433,-0.593657,0.162824,

0.046433,-0.524725,0.339780,

0.046433,-0.300892,0.547493,

0.046433,-0.121541,0.606409,

0.046433,0.000000,0.604643,

0.116085,-0.606202,0.042519,

0.116085,-0.598228,0.164699,

0.116085,-0.528766,0.343693,

0.116085,-0.303209,0.553800,

0.116085,-0.122477,0.613395,

0.116085,0.000000,0.611608,

0.208949,-0.612378,0.043166,

0.208949,-0.604322,0.167201,

0.208949,-0.534152,0.348913,
0.208949,-0.306297,0.562209,
0.208949,-0.123724,0.622708,
0.208949,0.000000,0.620895,

0.278607,-0.617010,0.043647,
0.278607,-0.608894,0.169073,
0.278607,-0.538193,0.352825,
0.278607,-0.308615,0.568515,
0.278607,-0.124661,0.629694,
0.278607,0.000000,0.627861,

0.348228,-0.621639,0.044144,
0.348228,-0.613459,0.170959,
0.348228,-0.542227,0.356745,
0.348228,-0.310927,0.574821,
0.348228,-0.125594,0.636676,
0.348228,0.000000,0.634823,

0.417985,-0.626281,0.044581,
0.417985,-0.618050,0.172797,
0.417985,-0.546292,0.360638,
0.417985,-0.313262,0.581131,
0.417985,-0.126538,0.643673,
0.417985,0.000000,0.641798,

0.487237,-0.630872,0.045241,
0.487237,-0.622547,0.174814,
0.487237,-0.550237,0.364628,
0.487237,-0.315508,0.587425,
0.487237,-0.127445,0.650618,

0.487237,0.000000,0.648724,

0.558373,-0.635655,0.045068,
0.558373,-0.627391,0.176161,
0.558373,-0.554631,0.368261,
0.558373,-0.318088,0.593780,
0.558373,-0.128487,0.657753,
0.558373,0.000000,0.655837,

0.622478,-0.639724,0.048004,
0.622478,-0.630943,0.180009,
0.622478,-0.557350,0.373222,
0.622478,-0.319423,0.599906,
0.622478,-0.129026,0.664179,
0.622478,0.000000,0.662248,

0.712823,-0.646455,0.039338,
0.712823,-0.639314,0.174525,
0.712823,-0.566321,0.373229,
0.712823,-0.325401,0.606887,
0.712823,-0.131443,0.673252,
0.712823,0.000000,0.671282,

0.705240,-0.643251,0.073973,
0.705240,-0.629704,0.203866,
0.705240,-0.551955,0.391723,
0.705240,-0.314056,0.610676,
0.705240,-0.126850,0.672444,
0.705240,0.000000,0.670524,

0.785943,-0.646112,0.093940,

0.785943,-0.629220,0.222387,
0.785942,-0.548530,0.406591,
0.785943,-0.310556,0.620416,
0.785943,-0.125424,0.680506,
0.785943,0.000000,0.678594,

0.905622,-0.651823,0.110843,
0.905622,-0.632223,0.239115,
0.905622,-0.548797,0.421830,
0.905622,-0.309494,0.633255,
0.905622,-0.124984,0.692482,
0.905622,0.000000,0.690562,

1.075137,-0.662583,0.115616,
1.075136,-0.642343,0.246690,
1.075137,-0.557291,0.433247,
1.075136,-0.314137,0.649048,
1.075137,-0.126857,0.709477,
1.075137,0.000000,0.707514,

1.298906,-0.678199,0.114587,
1.298907,-0.658217,0.250597,
1.298906,-0.571739,0.444523,
1.298907,-0.322623,0.669008,
1.298907,-0.130288,0.731925,
1.298906,0.000000,0.729891,

1.543870,-0.698395,0.085997,
1.543870,-0.683258,0.231680,
1.543870,-0.598509,0.442165,
1.543870,-0.340349,0.687394,

1.543870,-0.137467,0.756542,
1.543870,0.000000,0.754387,

1.932195,-0.723018,0.099224,
1.932195,-0.705885,0.251019,
1.932194,-0.616991,0.469613,
1.932195,-0.350167,0.723882,
1.932195,-0.141427,0.795471,
1.932195,0.000000,0.793220,

2.398987,-0.754274,0.101991,
2.398986,-0.736913,0.263204,
2.398988,-0.644581,0.495613,
2.398986,-0.366069,0.766095,
2.398987,-0.147852,0.842288,
2.398987,0.000000,0.839898,

2.763418,-0.778159,0.108248,
2.763419,-0.759942,0.276167,
2.763418,-0.664445,0.518093,
2.763419,-0.377204,0.799566,
2.763419,-0.152348,0.878832,
2.763418,0.000000,0.876342,

3.000000,-0.793886,0.110561,
3.000000,-0.775402,0.283109,
3.000000,-0.678055,0.531755,
3.000000,-0.384979,0.821075,
3.000001,-0.155489,0.902558,
3.000000,0.000000,0.900000,

B.6 Data Used in Figure 7.6

The optimized trimmed wing surface:

4,4

9,17

0.000000,0.000000,0.000000,0.000000,2.000013,4.000071,4.000071,4.000071,4.000071,

0.000000,0.000000,0.000000,0.000000,1.953999,2.877143,3.743359,4.516473,5.213567,5.910782,6.684058,7.550644,8.474067,10.428185,10.428185,10.428185,10.428185,

1.598286,-1.986928,0.236704,1.000000

1.503696,-1.986889,0.236687,1.000000

1.394238,-1.987181,0.234260,1.000000

1.223139,-1.988299,0.220734,1.000000

1.150696,-1.989218,0.210891,1.000000

1.035378,-1.991245,0.187314,1.000000

0.988097,-1.994330,0.152107,1.000000

1.039793,-1.996313,0.129522,1.000000

1.161378,-1.995516,0.138442,1.000000

1.236214,-1.994559,0.149801,1.000000

1.393472,-1.991767,0.181019,1.000000

1.517203,-1.988997,0.213329,1.000000

1.598271,-1.986932,0.236690,1.000000

1.593267,-1.773180,0.215813,1.000000

1.504417,-1.768923,0.216828,1.000000

1.369073,-1.768272,0.216182,1.000000

1.180417,-1.764640,0.202427,1.000000

1.119052,-1.765927,0.195162,1.000000

0.989177,-1.765968,0.171160,1.000000

0.936018,-1.767224,0.134882,1.000000

0.985545,-1.773935,0.108633,1.000000

1.123141,-1.773577,0.115907,1.000000
1.206154,-1.773664,0.127913,1.000000
1.373859,-1.772609,0.159186,1.000000
1.499221,-1.774951,0.189970,1.000000
1.593227,-1.771619,0.215690,1.000000

1.583110,-1.340053,0.170520,1.000000
1.473098,-1.338996,0.174821,1.000000
1.329697,-1.330579,0.176899,1.000000
1.130351,-1.324399,0.169912,1.000000
1.028702,-1.320731,0.159991,1.000000
0.892470,-1.320696,0.137827,1.000000
0.829656,-1.319842,0.096962,1.000000
0.895757,-1.324835,0.066552,1.000000
1.041363,-1.327649,0.071865,1.000000
1.130579,-1.330263,0.080722,1.000000
1.334260,-1.339064,0.113211,1.000000
1.478484,-1.340989,0.145358,1.000000
1.583184,-1.341920,0.170555,1.000000

1.573026,-0.913409,0.119822,1.000000
1.448198,-0.905789,0.128595,1.000000
1.284599,-0.895160,0.135890,1.000000
1.057632,-0.881669,0.133047,1.000000
0.962712,-0.874100,0.128180,1.000000
0.792409,-0.868935,0.105919,1.000000
0.726448,-0.866501,0.063373,1.000000
0.791731,-0.877608,0.026954,1.000000
0.970080,-0.882051,0.026836,1.000000
1.065841,-0.890234,0.035226,1.000000
1.288998,-0.904038,0.063323,1.000000

1.455338,-0.908357,0.094447,1.000000
1.573046,-0.913922,0.119921,1.000000

1.567774,-0.699105,0.092981,1.000000
1.440686,-0.689362,0.103516,1.000000
1.262441,-0.675521,0.114141,1.000000
1.028543,-0.659538,0.115127,1.000000
0.922196,-0.652763,0.111073,1.000000
0.742867,-0.643803,0.090521,1.000000
0.673894,-0.643869,0.045825,1.000000
0.745711,-0.649633,0.007000,1.000000
0.928168,-0.661719,0.004701,1.000000
1.030056,-0.668460,0.011322,1.000000
1.269353,-0.683926,0.037781,1.000000
1.444099,-0.693526,0.068372,1.000000
1.567785,-0.699073,0.092931,1.000000

References

[Abbo59] Abbott, I., Theory of Wing Sections, Dover Publications, Inc., New York, 1959.

[ACSY93] ACSYNT V2.0 User Manual, ACSYNT Institute, Virginia Tech CAD Laboratory, 1993.

[Appl89] Applegarth, I., Catley, D., Bradley, I., "Clipping of B-spline Patches at Surface Curves", The Mathematics of Surfaces III, Clarendon Press, Oxford, 1989, pp. 229-242.

[AuCK95] Au, CK, Yuen, M M F, "Unified Approach to NURBS Curve Shape Modification", Computer-Aided Design, Vol.27, No.2, 1995, pp.85-93.

[Bake01] Baker, C. A., Watson, L. T., Grossman, B., Mason, W. h., Haftka, R. T., "Parallel Global Aircraft Configuration Design Space Exploration", International Journal of Computer Research, Vol. 10, No. 4, 2001, pp. 501-515.

[Brai84] Braibant, V., Fleury, C., "Shape Optimal Design Using B-Splines", Computer Methods in Applied Mechanics and Engineering, Vol. 44, No. 3, 1984, pp.247-267.

[Char97] Charlton, E., "An Octree Solutions to Conservation-laws over Arbitrary Regions with Applications to Aircraft Aerodynamics," Ph.D. dissertation, Univ. of Michigan, 1997.

[Crow] Crowell, G., "The Descriptive Geometry of Nose Cones," URL: http://www.myweb.cableone.net/cjcrowell/NCEQN2.doc

[Fari] Farin, G., "A History of Curves and Surfaces in CAGD", http://www.elsevier.com/wps/find/journaldescription.cws_home/505604/description

[Fari95] Farin, G. NURB Curves and Surfaces from Projective Geometry to Practical Use, A K Peters, Ltd., Wellesley, Massachusetts, 1995.

[Fari97] Farin, G., Curves and Surfaces for Computer Aided Geometric Design, Fourth Edition, Academic Press, 1997.

[Ferg86] Ferguson, D.R., "Construction of Curves and Surfaces Using Numerical Optimization Techniques", Computer-Aided Design 18, 1986, pp. 15-21.

[Flem92a] Fleming, S. "The Enhancement of PHIGS Plus B-spline Functionality for Geometric Modeling in CAD", Master's thesis, Virginia Polytechnic Institute and State University, April 1992.

[Flem92b] Fleming, S. Myklebust, A., "The Enhancement of PHIGS Plus B-spline Functionality for Geometric Modeling in CAD", Fourth IFIP WG5.2 on Geometric Modeling in Computer Aided Design, Rensselaerville, New York, September 27 – October 1, 1992.

[Floa92] Floater, M.S., "Derivatives of Rational Bezier Curves", Computer-Aided Geometric Design 9 ,1992, pp.161-174.

[Frit92] Fritsch, Frederick N. and Nielson , Gregory M., "On the Problem of Determining the Distance Between Parametric Curves", Curve and Surface Design, SIAM, edited by Hans Hagen, 1992, pp. 123-141.

[Glou96] Gloudemans, J., Davis, P., Gelhausen, P., "A Rapid Geometry Modeler for Conceptual Aircraft", Aerospace Sciences Meeting and Exhibit, 34th, Reno, NV, Jan. 15-18, 1996, AIAA-1996-52.

[Hohe95] Hohenberger, W., Reuding, Thomas, "Smoothing Rational B-spline Curves Using the Weights in an Optimization Procedure", Computer-Aided Geometric Design 12, 1995, pp.837-848.

[Hosc87] Hoscheck, J., "Approximate Conversion of Spline Curves", Computer-Aided Geometric Design, Vol. 4, 1987, pp.59-66.

[HuSM01] Hu, S.-M., Li, Y.-F., Ju, T., Zhu, X., "Modifying the Shape of NURBS Surfaces with Geometric Constraints", Computer-Aided Design 33, 2001, pp.903-912.

[Jain99] Jain, Aashish, “Error Visualization in Comparison of B-spline Surfaces”, Master thesis, Virginia Polytechnic Institute and State University, 1999.

[Jone91] Jones, R. W., “Intersection and Filleting of Non-Uniform B-spline Surfaces”, Master’s thesis, Virginia Polytechnic Institute and State University, 1991.

[Juha99] Juhasz, Imre, “Weight-based Shape Modification of NURBS Curves”, Computer-Aided Geometric Design 16, 1999, pp. 377-383.

[Kirk83] Kirkpatrick, S., Gelatt Jr., C. D., Vecchi,, M. P. “Optimization by Simulated Annealing,” Science 220, 1983, pp. 671-680.

[Lass86] Lasser, D., “Intersection of Parametric Surfaces in the Bernstein-Bezier Representation”, Computer-Aided Design, Vol. 18, No. 4, 1986, pp.186-192.

[Lepi00] Lepine, J., Trepanier J.-Y., Pepin, F., “Wing Aerodynamic Design Using an Optimized NURBS Geometrical Representation”, Aerospace Sciences Meeting and Exhibit, 38th, Reno, NV, Jan. 10-13, 2000 , AIAA-2000-669.

[Lepi01] Lepine, J., Guibault, F., Trepanier, J.-Y., Pepin, F., “Optimized Nonuniform Rational B-Spline Geometrical Representation for Aerodynamic Design of Wings”, AIAA Journal, Vol. 39, No. 11, Nov. 2001, pp. 2033-2041.

[Li05] Li, Y., Strauss, C., Gorin, A., “Hybrid Parallel Tempering and Simulated Annealing Method in Rosetta Practice”, in preparation.

[Limi44] Liming, R.A., Practical Analytic Geometry with Applications to Aircraft, The Macmillan Company, New York, 1944.

[Maso] Mason, W., “Appendix A: Geometry for Aerodynamicists”, in Applied Computational Aerodynamics Text/Notes, http://www.aoe.vt.edu/~mason/Mason_f/CAtxtTop.html

[Maso98] Mason, W., Knill, D., Giunta, A., Haftka, R., “Getting the Full Benefits of CFD in Conceptual Design”, Applied Aerodynamics Conference, 16th, Albuquerque, NM, June 15-18, 1998, AIAA-1998-2513.

[Metr53] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., “Equation of State Calculations by Fast Computing Machines”, Journal of Chemical Physics 21, 1953, pp. 1087-1092.

[More92] Moreton, H.P., Sequin, C.H., “Functional Optimization for Fair Surface Design”, ACM SIGGRAPH Computer Graphics, Proceedings of the 19th annual conference on Computer graphics and interactive techniques, Vol. 26, No. 2, 1992, pp.167-176.

[Mykl93] Myklebust, A. and P. Gelhausen, “Improving Aircraft Conceptual Design Tools - New Enhancements to ACSYNT”, AIAA Aircraft Design, Systems and Operations Meeting, Monterey, California, August 11-13, 1993. AIAA-1993-3970.

[Mykl94] Myklebust and Gelhausen, P., "Putting the ACSYNT on Aircraft Design,"AERO SPACE America, September 1994, pp 26-30 (cover article).

[Peng84] Peng, Q.S., “An Algorithm for Finding the Intersection Lines Between Two B-spline Surfaces”, Computer-Aided Design, Vol. 16, No. 4, 1984, pp.191-196.

[Pieg89] Piegl, L., “Modifying the Shape of Rational B-spline, Part I: Curves”, Computer-Aided Design 21, 1989, pp.509-518.

[Pieg97] Piegl, L., Tiller, W., The NURBS Book, 2nd Edition, Springer, Berlin, 1997.

[Roja94] Rojas, R., “Geometric Trimming of B-spline Surface”, Master’s thesis, Virginia Polytechnic Institute and State University, July 1994.

[Sama99] Samareh, J., “Status and Future of Geometry Modeling and Grid Generation for Design and Optimization”, Journal of Aircraft, Vol. 36, No. 1, Jan.-Feb. 1999, pp. 97-104.

[Sama01] Samareh, J., “Survey of Shape Parameterizaiton Techniques for High-Fidelity Multidisciplinary Shape Optimization”, AIAA Journal, Vol. 39, No. 5, May 2001, pp.877-884.

[Sobi97] Sobieczky, H., “Geometry Generator for CFD and Applied Aerodynamics”, in: New Design Concepts for High Speed Air Transport. CISM courses and Lectures Vol. 366, pp. 137-158, Springer, Wien, New York, 1997.

[Song04] Song, X., Sederberg, W., Zheng, J., Rarouki, R., Hass, J., “Linear Perterbation Methods for Topologically Consistent Representations of Free-form Surface Intersections”, Computer Aided Geometric Design 21, 2004, pp. 303-319.

[Trap99] Trapp, J., Sobieczky,H., “Interactive Parametrical Geometry Design”, Aerospace Sciences Meeting and Exhibit, 37th, Reno, NV, Jan. 11-14, 1999, AIAA-1999-829.

[Trep00] Trepanier, J.-Y., Lepine, J.L., Pepin, F., “An Optimized Geometric Representation for Wing Profiles Using NURBS”, Canadian Aeronautics and Space Journal Vol. 46, No. 1, March 2000, pp. 12-19.

[Wang91] Wang, G., Xu, W., “The termination criterion for subdivision of the rational Bezier curves”, Graphical Models and Image Processing, Vol. 53, No. 1, Jan 1991, pp. 93-96.

[Wang01] Wang, X., “Geometric Trimming and Curvature Continuous Surface Blending for Aircraft Fuselage and Wing Shapes”, Master’s thesis, Virginia Polytechnic Institute and State University, 2001.