NASA-CR-204988

FINAL REPORT

Grant NAG3-1481

NURBS-Based Geometry for Integrated Structural Analysis

Prepared by:
James H. Oliver, Principal Investigator
Department of Mechanical Engineering
Iowa Center for Emerging Manufacturing Technology
Iowa State University
Ames, Iowa 50011-2160

Prepared for:
Scott A. Thorp
NASA Lewis Research Center
21000 Brookpark Road, M.S. 86-12
Cleveland, Ohio 44135

Submitted:
March 13, 1997

James H. Oliver
Associate Professor

Richard E. Hasbrook
Contracts and Grants Officer

3-13-97

INTRODUCTION

This report describes the research and development progress accomplished under NASA Lewis Research Center Grant NAG3-1481, "NURBS-Based Geometry for Integrated Structural Analysis". This grant was initiated in April 1993 and completed in September 1996.

The primary goal of the project was to exploit the emerging defacto CAD standard of Non-Uniform Rational B-spline (NURBS) based curve and surface geometry to integrate and streamline the process of turbomachinery structural analysis. We focused our efforts on critical geometric modeling challenges typically posed by the requirements of structural analysts. We developed a suite of software tools that facilitate pre- and post-processing of NURBS-based turbomachinery blade models for finite element structural analyses. We also developed tools to facilitate the modeling of blades in their manufactured (or "cold) state based on nominal operating shape and conditions.

All of the software developed in the course of this research is written in the *C++* language using the Iris Inventor 3D graphical interface tool-kit from Silicon Graphics. In addition to enhanced modularity, improved maintainability, and efficient prototype development, this design facilitates the re-use of code developed for other NASA projects and provides a uniform and professional "look and feel" for all applications developed by the Iowa State Team.

ACCOMPLISHMENTS

The project accomplishments are presented in two parts. Part I, titled "Geometric Decomposition and Structural Shape Modification for Turbomachinery Blades" describes the research conducted by B.D. Hines and J.H. Oliver aimed at automating the finite element model creation process for blades and for using the results to create a model of the blade in its cold shape.

The technique developed in this research applied a very precise method to extract a mean-camber surface from a blade model. The method involved intersecting the blade with cylinders, mapping the resulting curves onto planes, tracing the mean-camber curves in the planar mapped sections via a numerical scheme, mapping the mean-camber points back into three dimensions, and finally lofting the resulting space curves to form a mean-camber surface. The finite element mesh was then computed by evaluating points on the mean-camber surface, and intersecting corresponding surface normals with the original blade surface models to obtain thickness values for each node. This method was very precise, but sometimes, rather slow.

To address this problem, the process was re-engineered to trade some dimensional accuracy for computational speed. This method, developed by D.C. Rickert and J.H. Oliver, is described as Part II of this report, titled: "Turbomachine Blade Structure Loading Correction and Analysis". This new method assumes that the input blade iso-parameter curves in the chord-wise direction are approximately planar. A heuristic method is employed to extract the mean-camber surface, and mesh generation proceeds in the same fashion. This part of the report also describes a new method for storing and associating analysis data with corresponding blade models to facilitate useful post-processing and analysis interpretation.

Finally, for completeness, an Appendix is included containing a technical paper which abstracts much of the information from Part I of the report. This paper was presented at the ASME Design Automation Conference in Minneapolis, MN, in September 1994.

PART I

Geometric Decomposition and Structural Shape Modification for Turbomachinery Blades

B.D. Hines and J.H. Oliver

CHAPTER 1. INTRODUCTION

Manufacturers of turbomachinery products are under the same competitive pressures driving many industries toward reduced product development time and improved product quality, while simultaneously reducing development costs. The technology underlying the turbomachinery industry has reached relative maturity, so that even small improvements in overall thermodynamic efficiency can provide a competitive advantage. Blade design is one of the most crucial elements of turbomachinery development.

As the performance of turbomachinery is enhanced, blade strength-to-weight ratio has increased to such a degree that blade deflections, due to normal operating conditions, cannot be neglected. Blades are generally designed by aerodynamicists in a nominal operating configuration, known as the "hot-shape". However, deflections due to centrifugal force, pressure loading, and thermal gradients require that the blade be manufactured in its so called "cold-shape" so that the hot-shape is realized after these effects are accounted for. Thus, given a definition of blade hot-shape geometry, engineers need an efficient and accurate method to calculate the corresponding cold shape geometry.

Current methods for cold-shape correction are approximate in both data extraction and displacement application. Also, these methods take a great deal of time due to the manual approach that is required. After the hot-shape geometry has been generated, the mean camber surface (a surface which is equidistant from both the pressure and suction sides of the blade) is generated manually using an approximate approach that bisects points on the pressure and suction sides. Specialized analysis software is employed to calculate the inverse deflections, i.e., those due to removal of the operating boundary conditions (Thorp and Downey, 1992).

The inverse displacements are then applied to the hot-shape geometry using an approximation method in which the cold-shape blade surface points are generated by a weighted average of the deflections. This method is generally adequate for large stiff blades with small displacements because the deflections (and thus errors) are small relative to the overall size of the blade. But for small blades with large deflections the error can be become too great for the accurate generation of the cold-shape geometry.

This thesis describes an integrated method for both finite element model generation and cold-shape correction. Blades are represented precisely as non-uniform rational B-spline (NURBS) surface models (Piegl, 1991). The software implementation of this method incorporates the Initial Graphics Exchange Specification (IGES) (NIST, 1991) format for both input of the hot-shape geometry, and output of the resulting cold-shape NURBS surface model. This representation medium facilitates complete integration of the procedure with existing design, analysis, and manufacturing applications.

The thesis is presented in seven chapters, including the introduction as Chapter 1. The second chapter contains a description of the geometric modeling concepts that were applied in the integrated cold-shape correction method. Chapter 3 outlines blade terminology and technology, including a detailed examination of mean camber line generation. In Chapter 4, a preprocessor is developed that decomposes a hot-shape blade surface model into a mean camber surface and associated thickness functions. A plate finite element mesh is generated on the resulting mean camber surface, i.e., for each mesh node a location and thickness are calculated. The finite element model is used as input for specialized analysis software for cold-shape correction (Ernst, 1992) resulting in the inverse displacements, which is accomplished using the same methods as currently are used. Chapter 5 describes a postprocessor that reconstructs the corrected, cold-shape blade in two steps. First, the nodal deflections are applied to the hot-shape finite element model to generate a cold-shape mean camber surface. Then the original hot-shape thickness functions are applied to the cold-shape mean camber surface to

generate characteristic blade section curves which are lofted to define the cold-shape blade model. Chapter 6 contains examples of actual turbine blades at various steps during the hot-shape decomposition and cold-shape reconstruction processes. Finally, Chapter 7 details the conclusions made from the cold-shape correction method, along with future improvements that will increase the capabilities of the process.

CHAPTER 2. GEOMETRIC PRELIMINARIES

In this work, blades are represented precisely as non-uniform rational B-spline (NURBS) surface models. General descriptions of B-spline curves and their various forms can be found in most geometric modeling textbooks (e.g., Farin, 1993; Mortenson, 1985, deBoor, 1978) or any of several survey articles on the topic (Böhm et al., 1984; Piegl, 1991). A brief description of B-splines and their applications are provided to facilitate the subsequent development of the integrated cold-shape correction method.

2.1 Basis Functions

The B-spline basis functions are typically generated via the Cox-deBoor algorithm (Cox, 1972; deBoor, 1972). Given knot vector $U = \{u_0, \ldots, u_i, u_{i+1}, \ldots, u_m\}$ a monotonically increasing sequence of real numbers, the ith B-spline basis function of degree, p (order, $p+1$), denoted $N_{i,p(u)}$, is defined by the recursive relationship,

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } (u_i \le u < u_{i+1}) \\ 0 & \text{otherwise} \end{cases}$$
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (2.1)$$

where it is understood that $0 / 0 = 0$.

2.2 Non-Rational B-Spline Curves

A pth degree non-rational B-spline is defined as follows,

$$C(u) = \sum_{i=0}^{n} N_{i,p} P_i \qquad 0 \le u \le 1 \tag{2.2}$$

where P_i are the control points and the $N_{i,p}(u)$ are the pth degree B-spline functions defined by equation 1.1 and a non-periodic knot vector $U=\{u_0,...,u_m\}$. The degree, number of knots, and the number of control points are related by the formula: $m = n+p+1$.

2.3 Rational B-Spline Curves

Homogeneous coordinates are used to represent points in 3D space in terms of points in 4D space. Any non-infinite point in 4D can be represented by the four numbers (wx, wy, wz, w), $w > 0$, which represents the Cartesian point (x, y, z) when normalized as $(x, y, z, 1)$. This normalization is a perspective map defined by

$$H(wx, wy, wz, w) = \begin{cases} (wx/x,\ wy/w,\ wz/w) & \text{if } w \neq 0 \\ \text{point at infinity on the line from the origin through } (x,y,z) & \text{if } w=0 \end{cases}$$

Four dimensional points are denoted by $\boldsymbol{P}^w$, and $\boldsymbol{P}=H\{\boldsymbol{P}^w\}$.

The set of 4D control points, $\boldsymbol{P}^w{}_i=\{w_i x_i,\ w_i y_i, w_i z_i, w_i\}$, $i=0,...,n$ defines non-rational curve (in 4D), whose perspective map in 3D is called a rational B-spline curve. Specifically,

$$C(u) = H\{C^w(u)\} = H\{\sum_{i=0}^{n} N_{i,p}(u) P_i^w\} = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} = \sum_{i=0}^{n} R_{i,p}(u) P_i \tag{2.3}$$

where

$$\boldsymbol{P}_i = H\{\boldsymbol{P}_i^w\} = \{x_i, y_i, z_i\}$$

and

$$R_{i,p}(u) = \frac{N_{i,p}(u)\, w_i}{\sum_{j=0}^{n} N_{j,p}(u)\, w_j}$$

The $R_{i,p}(u)$ functions are piecewise rational basis functions. It is assumed that $w_i \geq 0$ for all values of i. The w_i are called weights. These values are additional shape parameters of the curve. Weights allow for B-splines to represent conic and circular arcs that could not be accomplished using non-rational B-splines.

2.4 Non-Rational B-Spline Surfaces

A B-spline surface of degree (p, q) is specified by an $(m - p) \times (n - q)$ grid of control points $\boldsymbol{P}_{ij}$ arranged in a topologically rectangular array and knot vectors $\boldsymbol{U}$ and $\boldsymbol{V}$ of length $(m + 1)$ and $(n + 1)$, respectively. The surface, denoted as $S(u,v)$, is thus defined as the tensor product of the control point array and the B-spline basis functions defined over each knot vector:

$$S(u, v) = \sum_{i=0}^{(m-p-1)} \sum_{j=0}^{(n-q-1)} N_{i,p}(u)\, N_{j,q}(v)\, \boldsymbol{P}_{ij} \tag{2.4}$$

The knot vector governs the relationship between parametric and spatial variation, and its entries represent the parameter values at segment joints (knots). A non-periodic B-spline is characterized by a knot vector in which the first and last knot values are repeated $p + 1$ (order)

times. This results in a surface that interpolates the control points on the edge of the rectangular array. A B-spline is characterized as uniform if the difference between successive interior knots is constant.

2.5 Rational B-Spline Surfaces

Just as with rational B-spline curves, a rational B-spline surface assigns a scalar weight factor to each control point. The degree *(p,q)* rational B-spline surface in 3D is defined as the map H of a tensor product B-spline surface in 4D.

$$S(u,v) = H\{S^w(u,v)\} = H\{\sum_{j=0}^{m}\sum_{j=0}^{n} N_{i,p}(u)\,N_{j,q}(v)\,\boldsymbol{P}_{ij}^w\} \tag{2.5}$$

$$S(u,v) = \frac{\sum_{j=0}^{m}\sum_{j=0}^{n} N_{i,p}(u)\,N_{j,q}(v)\,w_{ij}\boldsymbol{P}_{ij}}{\sum_{j=0}^{m}\sum_{j=0}^{n} N_{i,p}(u)\,N_{j,q}(v)\,w_{ij}} = \sum_{j=0}^{m}\sum_{j=0}^{n} R_{i,p;j,q}(u,v)\,\boldsymbol{P}_{ij}$$

where $\boldsymbol{P}_{ij}$ are the 3D control points and

$$R_{i,p;j,q}(u,v) = \frac{N_{i,p}(u)\,N_{j,q}(v)\,w_{ij}}{\sum_{r=0}^{m}\sum_{s=0}^{n} N_{r,p}(u)\,N_{s,q}(v)\,w_{rs}}$$

The $R_{i,p;i,q}(u,v)$ functions are the bivariate rational basis functions.

Just as with the curve case, the weights w_{ij} are additional shape parameters. They

affect the surface only locally, and their push-pull effect can be precisely quantified.

2.6 Interpolation

The global curve-interpolation problem can be solved as follows. Given a set of data points Q_k, where $k = 0,..., n$, a B-spline curve is desired that for certain parameter values u_k, where $k = 0,..., n$ agrees with Q_k; that is,

$$Q_k = C(u_k) = \sum_{i=0}^{n} P_i N_{i,p}(u_k) \tag{2.6}$$

To solve this equation, the parameter values are needed at which the data points are assumed, the knot vector, and the degree of the curve. The method used will work for curves of any arbitrary degree. One of the several methods to compute the parameter values is the centripetal method,

$$u_0 = 0, u_i = u_{i-1} + \frac{|Q_i - Q_{i-1}|^{1/2}}{\sum_{j=1}^{n-1} |Q_j - Q_{j-1}|^{1/2}}, u_n = 1 \tag{2.7}$$

Given the parameter values, a knot vector is needed that reflects the distribution of these parameters. The following averaging method is commonly used,

$$U = \{0, 0, ..., 0, v_1, ..., v_{n-p}, 1, 1, ..., 1\} \tag{2.8}$$

where the end points are repeated with multiplicity $p+1$, and,

$$v_j = \frac{1}{p} \sum_{i=j}^{j+p+1} u_i \qquad j = 1, \ldots, n-p \tag{2.9}$$

It can be proved that the coefficient matrix,

$$N_{i,p}(u_k)\,|_{i,k=0,\ldots,n}$$

is totally positive and with a bandwidth less than p. Therefore, the linear system (Equation 2.6) can be solved by Gauss elimination.

2.7 Surfaces of Revolution

Surfaces of revolution are used extensively throughout both the hot-shape preprocessor and the cold-shape correction postprocessor. The most conventional way to define these surfaces is to define a profile curve in the a coordinate plane, and rotate it about the one of the

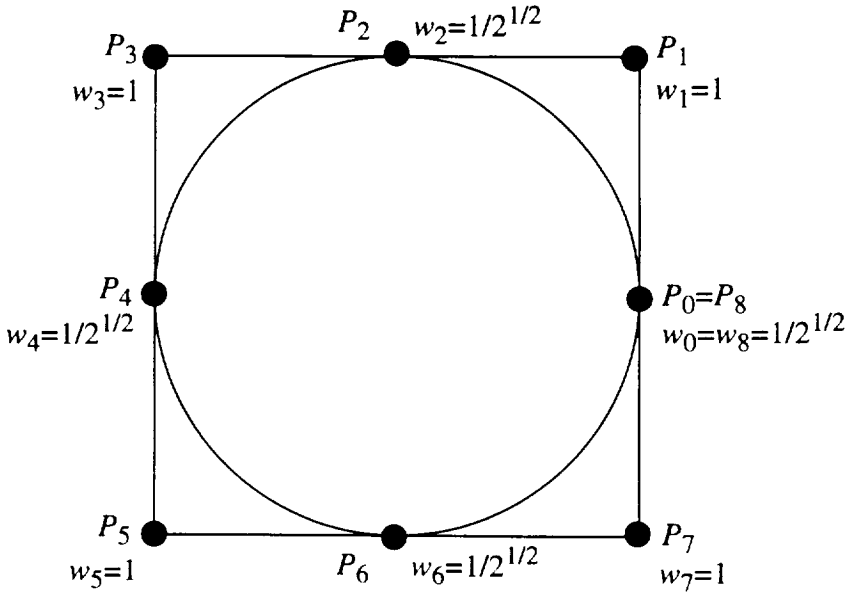

Figure 2.1 The nine-control-point-square-based NURBS circle

axes in the plane. Assume that the profile curve is a qth degree NURBS on the knot vector $\boldsymbol{U}$ of the form,

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} R_{j,q}(u)\,\boldsymbol{P}_j \tag{2.10}$$

Then the surface of revolution is obtained by combining Equation 2.5 with one of the circle definitions. For this example, the nine-control-point square-based NURBS circle, shown in Figure 2.1, is used. The circle is composed of four conic segments, each sweeping 90 degrees. The circle has the representation,

$$\boldsymbol{C}(u) = \sum_{i=0}^{8} \boldsymbol{P}_j R_{j,q}(u) \tag{2.11}$$

where the control points form a square, and,

$$\boldsymbol{U} = \{0, 0, 0, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{3}{4}, \tfrac{3}{4}, 1, 1, 1\} \tag{2.12}$$

$$w_i|_{i=0}^{8} = \{1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1\}$$

Using this circle definition, a full surface of generate by,

$$\boldsymbol{S}(u, v) = \sum_{i=0}^{8}\sum_{j=0}^{m} \boldsymbol{P}_{i,j} R_{i,2;j,q}(u, v) \tag{2.13}$$

where for fixed j, $\boldsymbol{P}_{ij}$ lie in a plane perpendicular to the plane of the generating curve. For fixed j the weights are,

$$w_{i,j} = \{w_j, \frac{1}{\sqrt{2}}w_j, \frac{1}{\sqrt{2}}w_j, w_j, \frac{1}{\sqrt{2}}w_j, \frac{1}{\sqrt{2}}w_j, w_j\}, \qquad (2.14)$$

i=0,...,8

where w_j are the weights for the profile curve from Equation 2.6.

2.8 Lofting

Cross-section design is concerned with surface construction based on curves to generate B-spline surfaces. The technique utilized by the cold-shape correction method is lofting. Given a set of NURBS curves though which a NURBS surface is to pass, the lofted surface is obtained in three steps:

1) Make all cross sectional curves compatible. That is, all the curves should have the same degree and number of control points and be defined over the same knot vector. Assume that this has been done; then,

$$C_k^w(u) = \sum_{i=0}^{n} Q_{i,k}^w N_{i,p}(u) \qquad k = 0, \ldots, k \qquad (2.15)$$

are u-directional curves lying on the surface (isoparametric lines in the u direction) and defined over the same knot vector $\boldsymbol{U}$.

2) Compute v values and a knot vector $\boldsymbol{V}$ for interpolation with degree-q NURBS curves. The v values are needed as the curves from Equation 2.13 are assumed to be at a certain fixed v.

3) Using the knot vector $\boldsymbol{V}$ and the v values computed in Step 2, interpolate curves through the control points of Equation 2.11. More precisely, for each i, i = 0,...,n,

obtain,

$$C_i^w(v) = \sum_{j=0}^{m} P^w_{i,j} N_{j,q}(v) \tag{2.16}$$

so that Equation 2.12 interpolates $Q^w_{i,k}$ at certain v values (note that if the u-directional curves of Equation 2.10 are rational then rational interpolants are to be used). The control points of Equation 2.11 are then the control points over the lofted surface,

$$S^w(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P^w_{ij} N_{i,p}(u) N_{j,q}(v) \tag{2.17}$$

defined over the knot vectors U and V.

2.9 Implementation

Despite their thorough documentation in the open literature, computational implementation of general purpose NURBS modeling techniques is a large and complex undertaking. Thus, this research takes full advantage of the DT_NURBS Spline Library provided by the Naval Surface Warfare Center (US Navy, 1993). This library provides a complete and robust collection of NURBS-based modeling and analysis functions and is available for public distribution in the United States.

CHAPTER 3. BLADE TERMINOLOGY AND TECHNOLOGY

3.1 Blade Terminology

The following basic terminology is used throughout the thesis. Figure 3.1 shows the typical cylindrical *r*-*z*-θ coordinate system used in blade design. The blade rotates about the z-axis. Generally, for axial turbomachinery components, the radial (*r*-) axis is referred to as the blade span direction, and the z-axis is referred to as the chord direction. The blade tip is the surface at the maximum r-coordinate, and the root is that of minimum *r*-coordinate. The leading edge refers to the upstream edge of the blade, and the trailing edge to its downstream edge. Finally, the convex side of the blade is referred to as the suction side, and the concave side as

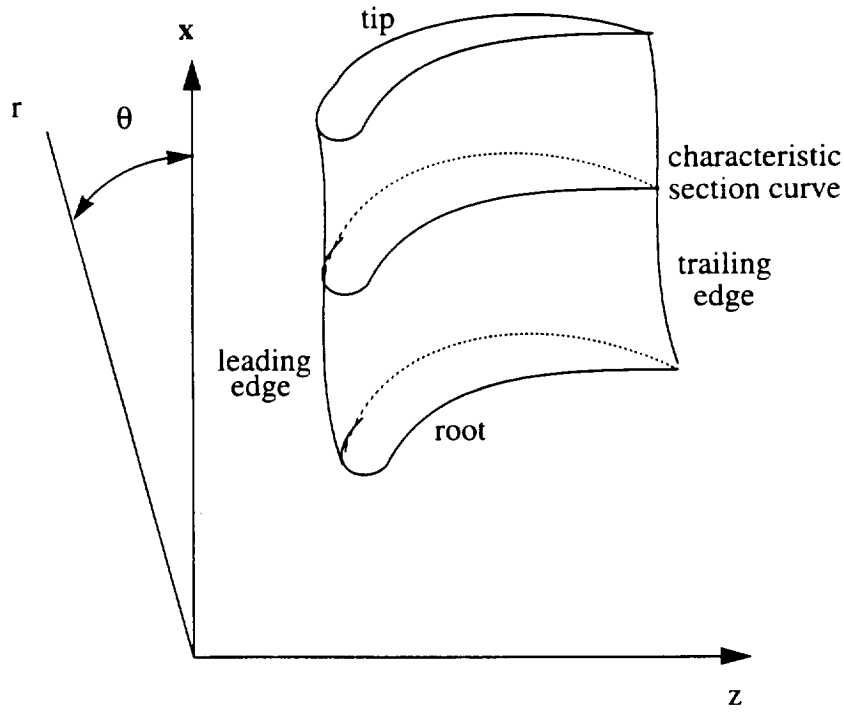

Figure 3.1 Blade terminology

the pressure side.

Blades are typically designed by lofting a number of characteristic blade section curves arranged along a stacking axis which defines the span-wise direction. The blade section curves are generally developed on one of three types of construction spaces: planes perpendicular to the *r*-axis at monotonically increasing *r*-values, concentric cylinders about the z-axis, or on concentric surfaces of revolution defined by *r*-*z*-curves which characterize desired flow field (Oliver et. al., 1994). For the latter two options, transformations exist to map the spaces onto equivalent planar regions. Thus a characteristic blade section curve may be defined in a two-dimensional coordinate system and mapped into a general space curve lying within a surface of revolution. Alternatively, given a surface of revolution containing a blade section (space) curve, an equivalent two-dimensional image can be obtained via the inverse of these mapping functions.

3.2 Mean Camber Technology

A mean camber surface can be considered the skeletal surface of the blade. As shown in Figure 3.2, blade surface points lie on mean camber surface normals at *t/2* and *-t/2*, where *t* is the thickness of the blade.

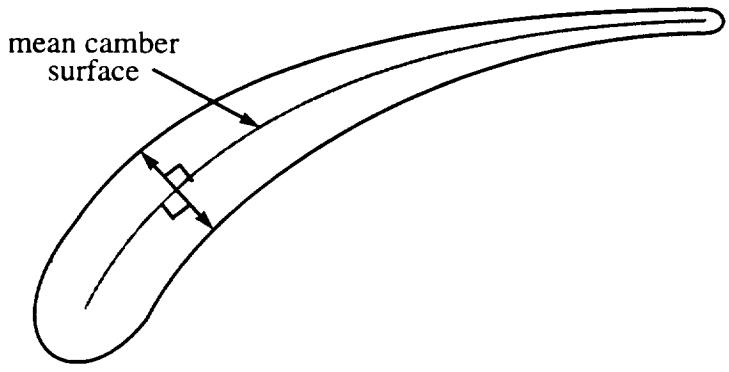

Figure 3.2 Mean camber surface definition

To generate the mean camber surface, mean camber lines must be first calculated for each blade section cut. The mean camber line, *c(s)*, is defined as the locus of the midpoints of the straight line segments spanned across the blade section, such that the normal to the segment at tis midpoint is tangent to the mean camber line. The camber line is a parametric curve, parametrized with respect to its arc-length, *s*. The mean camber line definition is shown in Figure 3.3. The mean camber line is generated by adapting a method Patrikalakis and Bardis (1992) derived that calculated the mean camber lines for marine propellers. For the algorithm to be successful, the blade section curve must be at least C^1 continuous and planar.

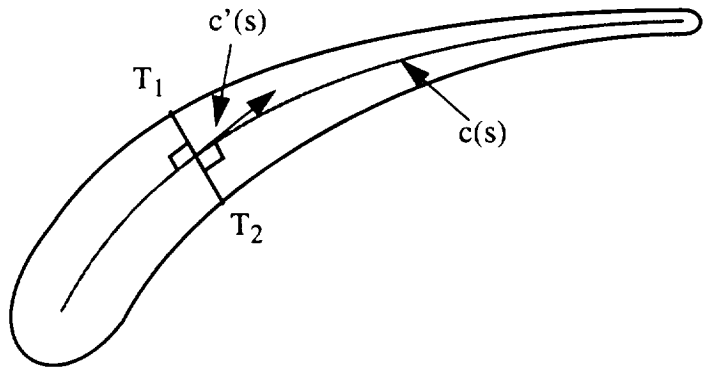

Figure 3.3 Mean camber line and variable definition

The variables used in the algorithm are as follows. The variables *p(t)* and *q(t)* represent the *x*- and *y*- coordinates of the blade section curve.The parameter values t_1, t_2 correspond to the parameter values at the two matching points $\boldsymbol{T}_1$, $\boldsymbol{T}_2$. To define the mean camber line, a system of differential equations is developed that determine $t_1(s)$, $t_2(s)$, $x(s)$, and $y(s)$ as functions of the arc-length parameter *s* of the camber line.

Since $\boldsymbol{T}$ is the midpoint of $\boldsymbol{T}_1\boldsymbol{T}_2$

$$2x(s) = p(t_1(s)) + p(t_2(s)) \tag{2.1}$$

$$2y(s) = q(t_1(s)) + q(t_2(s)) \tag{2.2}$$

Since the tangent to the camber line at T is $c'(s) = (x'(s), y'(s))$ and T_1T_2 is perpendicular to $c'(s)$ then

$$[p(t_2(s)) - p(t_1(s))]x'(s) + [q(t_2(s)) - q(t_1(s))]y'(s) = 0 \tag{2.3}$$

Since s is the arc-length of the mean camber line, $c'(s)$ is a unit vector. Thus

$$x'^2(s) + y'^2(s) = 1 \tag{2.4}$$

Solving 2.3 and 2.4 with respect to $x'(s)$ and $y'(s)$ gives

$$x'(s) = \pm\frac{q(t_2(s)) - q(t_1(s))}{D(t_1(s), t_2(s))} \tag{2.5}$$

$$y'(s) = \mp\frac{p(t_2(s)) - p(t_1(s))}{D(t_1(s), t_2(s))} \tag{2.6}$$

where $D(t_1(s), t_2(s))$ is the Euclidean distance between $T_1(s)$ and $T_2(s)$

$$D(t_1(s), t_2(s)) = \sqrt{[p(t_2(s)) - p(t_1(s))]^2 + [(q(t_2(s)) - q(t_1(s)))]^2} \tag{2.7}$$

Each sign pair corresponds to one of the marching directions along the pressure and suction side. Differentiating 2.1 and 2.2 with respect to s gives

$$2x'(s) = p(t_1(s))\,t_1'(s) + p(t_2(s))\,t_2'(s) \tag{2.8}$$

$$2y'(s) = q(t_1(s))\,q_1'(s) + q(t_2(s))\,q_2'(s) \tag{2.9}$$

Solving the linear system 2.8 and 2.9 with respect to $t_1'(s)\,(s)$ and $t_2'(s)\,(s)$ and substituting $x'(s)$ and $y'(s)$ from 2.5 and 2.6 gives

(2.10)

$$t_1'(s) = \pm\frac{2}{X(t_1(s), t_2(s))} \times \frac{q_t(t_2(s))\,[Q_1] + p_t(t_2(s))\,[P_1]}{D(t_1(s), t_2(s))}$$

(2.11)

$$t_2'(s) = \pm\frac{2}{X(t_1(s), t_2(s))} \times \frac{q_t(t_1(s))\,[Q_1] + p_t(t_1(s))\,[P_1]}{D(t_1(s), t_2(s))}$$

where

$$Q_1 = q(t_2(s)) - q(t_1(s)) \tag{2.12}$$

$$P_1 = p(t_2(s)) - p(t_1(s)) \tag{2.13}$$

and where $X(t_1(s),t_2(s))$ is

$$X(t_1(s), t_2(s)) = p_t(t_1(s))\, q_t(t_2(s)) - p_t(t_2(s))\, q_t(t_1(s)) \tag{2.14}$$

Equations 2.10 and 2.11 form a system of differential equations that can be integrated with respect to s. The mean camber line is then completely defined by 2.1 and 2.2 The sign is chosen such that the section is traversed in the correct direction. Equations 2.10 and 2.11 are then solved using the Adams - Moulten differential equation solving technique.

CHAPTER 4. HOT- SHAPE BLADE FINITE ELEMENT MODELING

4.1 Hot-Shape Preprocessor

The cold-shape finite element analysis requires a model comprised of plate elements. Thus, for each node, the position and corresponding blade thickness must be generated. To facilitate this calculation, the hot-shape blade mean camber surface is calculated. The procedure for mean camber surface construction involves extraction of several characteristic blade profile curves, and application of the two-dimensional algorithm for mean camber curve generation described in Chapter 3. The flowchart of the mean camber surface generation method is shown in Figure 4.1.

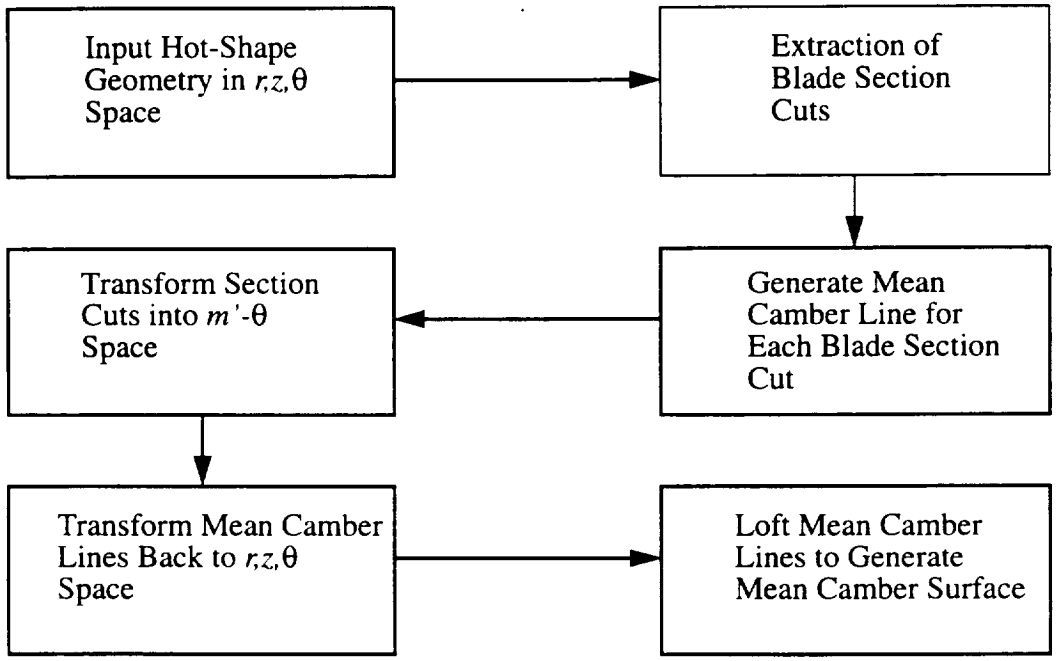

Figure 4.1 Decomposition of hot-shape blade geometry to generate mean camber surface

Ideally, the blade should be decomposed using the exact method used to construct it, i.e., planes, cylinders, or surfaces of revolution. However, since only the surface model of the blade is provided as input, there is generally no way of knowning what method was used to construct the blade. Thus, to generate a representative set of characteristic profile curves, the blade is intersected with concentric cylinders centered about the z-axis. Cylinders were chosen over parallel planes because the concentric cylinder method is more commonly used by blade designers. The number of cylinders (and resulting section profile curves) required to accurately characterize the overall shape of the blade is chosen heuristically as one for each knot in the span-wise parametric direction of the original NURBS blade model.

For a particular blade-cylinder intersection, a general surface-surface intersection algorithm is necessary to extract the intersection curve. Parametric surface intersection is an active research topic in its own right, and a number of techniques have appeared in the literature, see, Patrikalakis and Prakash (1990), Barnhill and Kersey (1990) and Kriezis et al. (1992), for instance. In this application, the surfaces involved are relatively simple, and an intersection algorithm implemented in the DT_NURBS Spline Geometry Library (US Navy, 1993) has proven to be robust and efficient.

For each blade-cylinder intersection curve, the mean camber line must be calculated. The result of the surface-surface intersection is a three-dimensional (space) curve, but the mean camber algorithm of Patrikalakis and Bardis (1992) requires a closed planar curve as input. Thus an angle preserving transformation is developed to map the space curve onto a plane. The transformation is generated from a two-dimensional curve in the r-z-plane which defines a surface of revolution about the z-axis. The transformation maps an r-z-curve onto a one-dimensional coordinate (m') equivalent to meridional arc length normalized with respect to radius. The justification for and details of this mapping are described in Oliver et. al. (1994). It is summarized here for completeness. Curve arc length is defined by,

$$m_i = \int_0^{u_i} \sqrt{(r^u(u))^2 + (z^u(u))^2}\, du \tag{4.1}$$

where the superscripts represent differentiation, and m_i represents accumulated curve length evaluated at a number of parametric samples. These values are interpolated with a cubic NURBS curve so that the resulting arc length function may be normalized with respect to radius,

$$m'_i = \int_0^{u_i} \frac{m^u(u)}{r(u)}\, du \tag{4.2}$$

the resulting m_i' values are interpolated with a cubic NURBS curve to complete the mapping. The transformation has the effect of "un-rolling" the surface of revolution on which the blade profile lies, and "flattening" it in the axial (z-) direction. It allows representation of a space curve defined in cylindrical coordinates in terms of its circumferential ($r\theta$) and axial (m) arc length components. The transformation preserves angles because both length components are normalized with respect to radius, i.e.,

$$\tan\alpha = \frac{d(r\theta)}{dm} = \frac{d\theta}{dm/r} = \frac{d\theta}{dm'} \tag{4.3}$$

where α is an angle in the m-$r\theta$ or m'-θ coordinate system. In this application a cylinder is represented as a line parallel to the z-axis. After transformation, the mean camber curve algorithm is applied to the m'-θ image of the intersection curve, resulting in a series of points which lie on the mean camber curve. These points are then mapped back into the r-z-θ coordinate system, via the inverse of the above transformation.This technique is applied to each intersection curve.

The root and tip of the blade geometry are characterized as iso-parameter curves on the blade surface and must be handled as special cases. In general, these are space curves

which do not lie on a cylinder. The goal is to find a representative r-z-curve defining a surface of revolution which contains the curve. The first step is to intersect the blade root or tip curve with a number of planes of constant z-coordinate. For most axial turbomachinery applications, section curve orientation and convexity are defined such that, excluding the leading and trailing edge, only two intersection points are generated from any particular z-plane intersection. If the blade was constructed according the one of the three methods outlined above, the two intersection points will have identical r-coordinate values. In many cases, however, the values are slightly different. Thus, the following approximate solution is implemented. For each intersection pair, the point with the smallest r-value is retained, if traversing the tip curve, and the one with the largest value is retained if traversing the root curve. This ensures the resulting surface of revolution will completely intersect both sides of the blade.

The r-z coordinates of these points are interpolated with a cubic NURBS curve which is then employed to develop m'-θ transformation functions for both the blade root and tip iso-parameter curves. After transformation, the mean camber curves for each are generated as described above.

Given the necessary mean camber curves in the r-z-θ coordinate system, a mean camber surface is generated via NURBS lofting techniques (Piegl, 1991). The resulting surface serves as the basis for plate mesh generation. An interactive interface allows user specification of the number of nodes desired in both span-wise and chord-wise directions. The default nodal distribution is at uniform parametric intervals in both parametric directions. Alternatively, several options are provided for biasing nodal distribution toward specific areas of interest. Since the underlying geometry is a NURBS surface, the three-dimensional node location is computed via surface evaluation at each parameter pair; generated as described above. To calculate the blade thickness at each node, a mean camber surface normal is computed and intersected with the original (outer) blade surface. The thickness is given by doubling the distance between the node and the intersection point. The resulting blade hot-shape plate finite element

model is output to a file which is compatible with the cold-shape analysis software.

4.2 Generation of Cold-Shape Displacements

This section of the describes the generation of inverse displacements (Ernst, 1992). First, a nonlinear displacement analysis is made on the hot-shape geometry, which incorporates aerodynamic and centrifugal loads associated with the operating design point. The displacements that result from the analysis are subtracted from the hot shape geometry, which leads to a preliminary cold-shape blade geometry, called cold-shape 1. The second step consists of running a nonlinear displacement analysis on cold shape 1 with the same aerodynamic and centrifugal loads applied. The displacements that result from the analysis are added to cold shape 1, and a comparison is made with the desired hot-shape geometry. If the maximum displacement between the hot-shape and cold-shape 1 with added displacements is larger than 0.001 inches, then all the differences are subtracted from cold-shape 1 which is called cold-shape 2, and the second step is repeated with the new cold-shape. The second step is repeated until the maximum difference between the hot-shape and cold-shape with added displacements is less than 0.001 inches at all locations on the blade geometry.

A corresponding case to what was described above is shown as a one degree of freedom problem in Figure 4.2. A mass, M, is attached to a nonlinear spring, k. The hot-shape for this model, which is shown in Figure 4.2(a), is such that the mass is a distance a from the fixed end of the spring. A steady force, F, shown in Figure 4.2(b), is applied to the mass, and the resulting displacements is d_1. Figure 4.2(c) represents the cold-shape after the first iteration, where the mass is a distance a-d_1 from the fixed end. Applying load F on the cold-shape model shown in Figure 4.2(d), a new displacement arises such that the difference between the original hot-shape and the cold-shape with added displacement is equal to d_2. The difference between cold-shape 2 and the hot-shape is not yet less than 0.001 inches; thus another iteration takes place, where the new cold shape is represented in Figure 4.2(e). Here, the distance

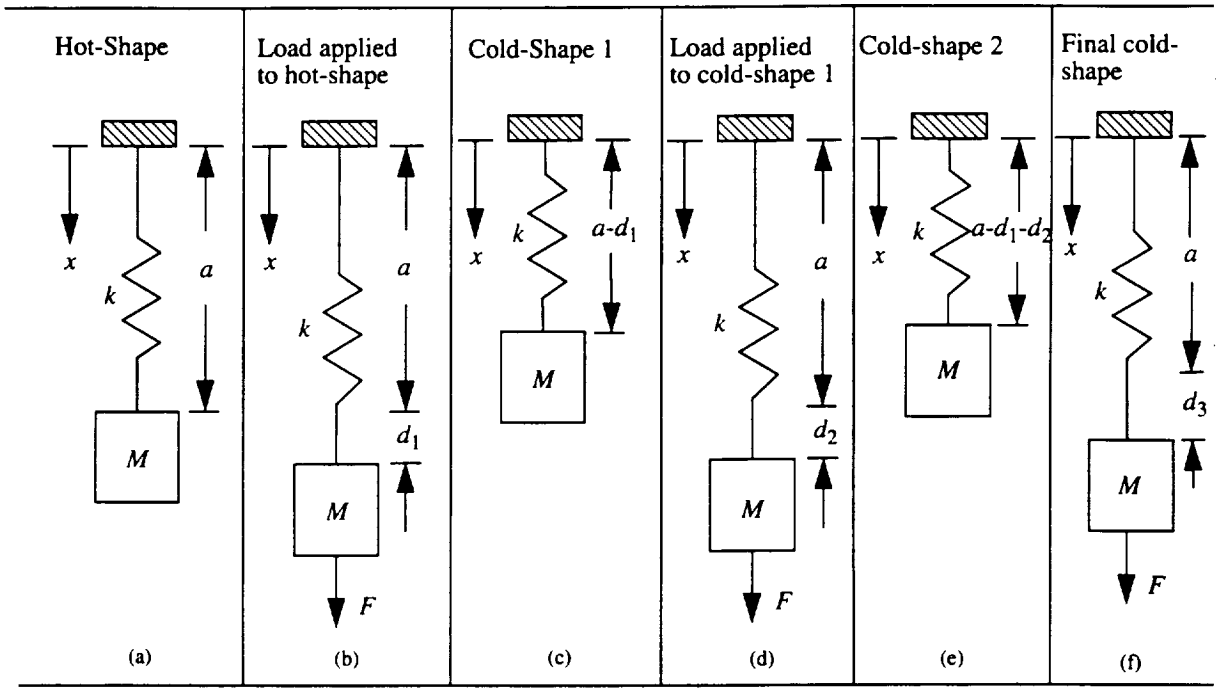

Figure 4.2 Schematic of cold-shape iteration

of the mass form the fixed end of the spring is a-d_1-d_2. After applying the load for the last time, the difference between the hot-shape and the new cold-shape with added displacements is equal to d_3. This is demonstrated in Figure 4.2(f). d_3 happens to be less than 0.001 inches. Thus the final cold-shape has been found and is equal to a-d_1-d_2.

This mesh generation technique has been implemented with an interactive visual interface using the IRIS Inventor graphical toolkit and the *C++* programming language on Silicon Graphics workstations. The technique has been tested successfully on a variety of blade models. The interface allows the user to interactively view the hot-shape blade and the mean camber surface, and to control the variables in generating the finite element mesh.

CHAPTER 5. COLD-SHAPE CORRECTION POSTPROCESSOR

The goal of the cold-shape postprocessor is to generate an accurate NURBS surface model of a blade in its nominal configuration, i.e., its cold-shape. The plate element model described above is augmented with characterizations of the blade's steady state pressure and centrifugal loading, and an iterative finite element analysis procedure is utilized to determine nodal displacements due to "unloading" the blade (Thorp and Downey, 1992). In addition to this the displacement data, the postprocessor requires the hot-shape finite element model as input. Although it is not required for the shape correction algorithm, the original hot-shape blade geometry may also be input to provide visual comparison between hot and cold shapes.

The cold-shape correction postprocessor makes extensive use of the mean camber technology developed for the hot-shape preprocessor. The basic assumption underlying the technique is that deflections due to operational loading are characterized primarily by bending and torsion, and that thickness change in the blade is negligible. The procedure is depicted schematically in Figure 5.1, and the corresponding steps are summarized as follows:

1) displace each mean camber curve
2) find a surface of revolution which contains each one
3) map each to its unique m'-θ coordinate system
4) apply the hot-shape thickness functions to generate section profile curves
5) transform each section profile curve into the cylindrical coordinate system
6) loft the three-dimensional section profile curves

The hot-shape mean camber surface is developed from a number of intersection curves. Because the mesh distribution is user specified, the resulting chord-wise rows of nodes

do not generally correspond to the construction mean camber curves. However, since the mean camber surface is constructed by lofting curves which lie in cylindrical surfaces of revolution, arbitrary iso-parameter curves in the chord-wise direction will also lie on a cylinder (except near the root and tip). The vector displacement output of the cold-shape analysis is applied to each node of a row, and a cubic NURBS curve is interpolated through the points (Figure 5.1a). This displaced mean camber curve will generally not lie on any cylinder. Thus, the surface of revolution on which the displaced (cold-shape) mean camber curve lies must be generated. In this case, the $r(u)$ and $z(u)$ components of the displaced mean camber curves are used to generate the m'-θ mapping for each blade section (Figure 5.1b).

For each row of nodes, a thickness function corresponding to each mean camber curve is generated based on the thickness value specified at each node. Because thickness change is assumed to be negligible the original cylindrical surface of revolution of the hot-shape blade is used to transform each thickness value into an equivalent normalized space (Figure 5.1d). The transformed thickness values are interpolated with a cubic NURBS curve using the same parameterization as the corresponding hot-shape mean camber curve (Figure 5.1c). Thus, for

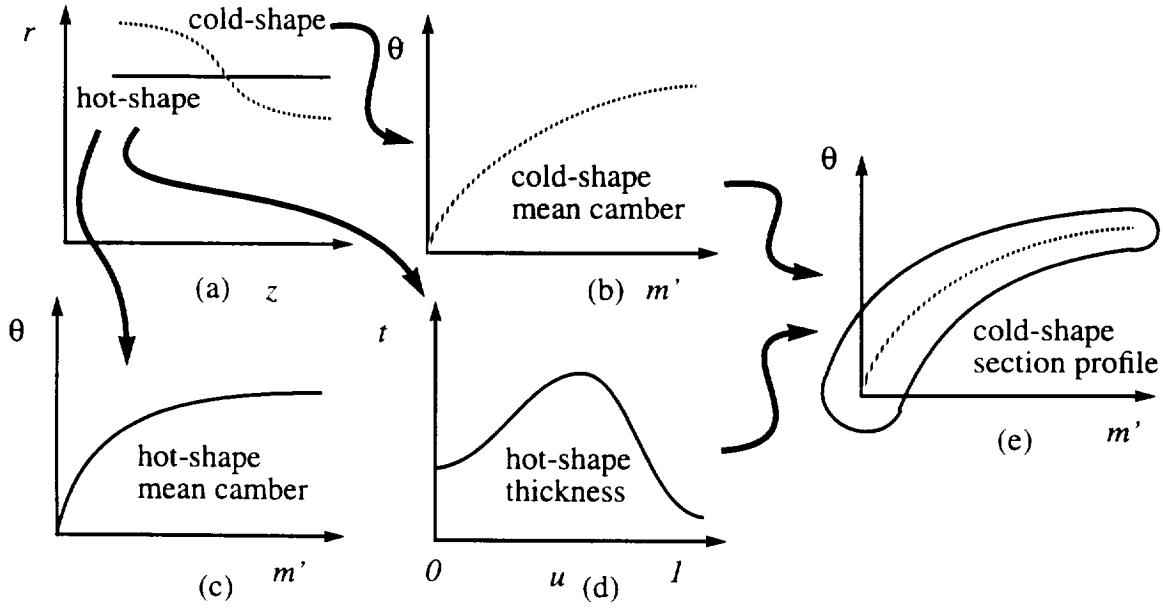

Figure 5.1 Cold-shape correction methodology

every parameter value $u \in [0, 1]$ a blade thickness can be computed relative to a specific location on the mean camber curve.

To create the cold-shape section profile curves, the thickness functions are applied to each cold-shape mean camber curve (Figure 5.1e). A number of points are generated on each section profile by offsetting cold-shape mean camber curve points by one half of the corresponding thickness in both directions perpendicular to the mean camber curve. Since it is based on a mean camber curve, the domain of the m'-θ mapping may not be sufficient to characterize all the generated section points. As shown in Figure 5.2, due to the offsetting procedure, some the section points may have negative m' values, or values larger than the maximum of the mean camber curve. To address this problem, an extrapolation technique is implemented to extend the domain of the m'-θ mapping. After each row of displaced nodes is fit with a NURBS curve in r-z-θ space, two additional points are added on each end based on the maximum thickness of the section and the parametric first derivatives. The curve is then interpolated again with these additional points to extend the mapping domain in each direction.

After the cold-shape section profile points are generated, those on the suction and pres-

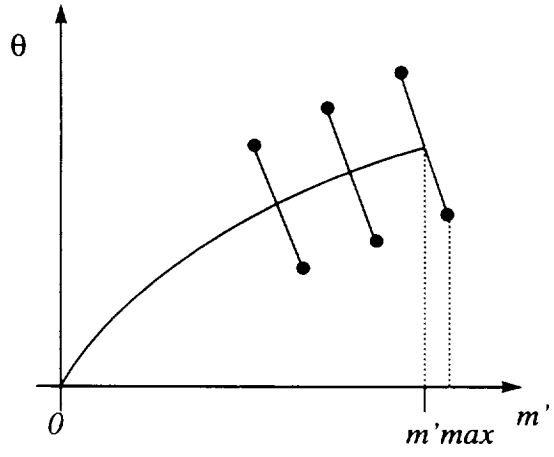

Figure 5.2 Cold-shape section profile points exceeding the domain of the m'-θ mapping

sure sides are interpolated with individual cubic NURBS curves. The leading and trailing edges are modeled with rational quadratic Bezier segments, as shown in Figure 5.3. The interior control point of each quadratic segment is determined by the intersection of the appropriate pressure and suction curve tangents, thus ensuring G^1-continuity. The weights corresponding to each control point are assigned to characterize either circular arcs or ellipse segments, as specified by the user. Note that if the tangents intersect such that the required arc span is larger than p, then two conic segments are generated. The degree of the leading and trailing edge segments is then increased to match that of the pressure and section curves, using a NURBS degree elevation algorithm (US Navy, 1993). Finally, the spline segments are joined, and the curve is reparameterized so that the $u = 0$ and $u = 1$ points corresponds to the mid-arc point of the trailing edge segment.

After the characteristic cold shape blade sections are generated in m'-θ space, they are mapped into r-z-θ space by transforming their respective control points. Then the three dimensional blade section curves are lofted in the span-wise direction to produce the final NURBS surface model of the blade in its cold-shape configuration.

This cold-shape correction technique has been implemented with an visual interface using the IRIS Inventor graphical toolkit and the *C++* programming language on Silicon

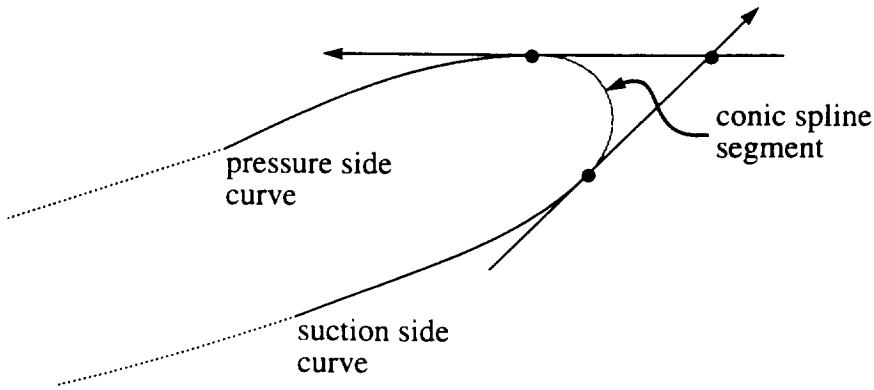

Figure 5.3 Leading/trailing edge conic spline fit

Graphics workstations. The technique has been tested successfully on a variety of blade models. The interface allows the user to interactively view the hot-shape blade and the shape corrected cold -shape blade.

CHAPTER 6. EXAMPLES

6.1 Hot Shape Blade Decomposition Examples

The hot-shape decomposition process and finite element mesh generation procedure for a two general turbomachinery blade examples is demonstrated in the following figures. Figures 6.1-6.4 show the hot-shape decomposition process for the first example blade and Figures 6.5-6.8 for the second example blade. First, the nominal hot-shape blade geometry is displayed in wireframe form in Figures 6.1 and 6.5. Next, the generated blade section curves and their corresponding mean camber curves are shown in Figures 6.2 and 6.6. Finally, in Figures 6.3 and 6.4, and Figures 6.7. and 6.8 the mean camber surface and associated finite element mesh are shown.

Example Blade 1:

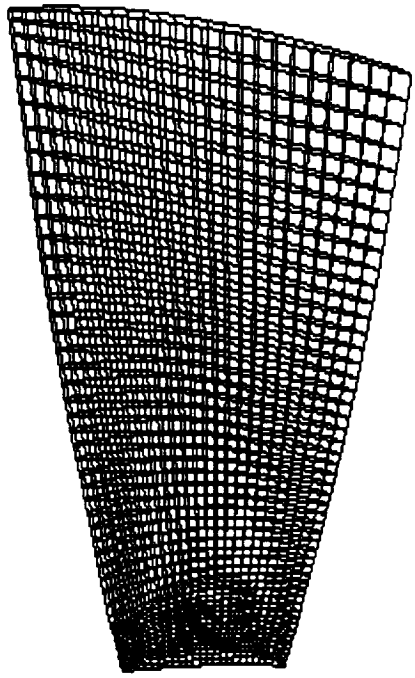

Figure 6.1 Hot-shape blade geometry for example blade 1

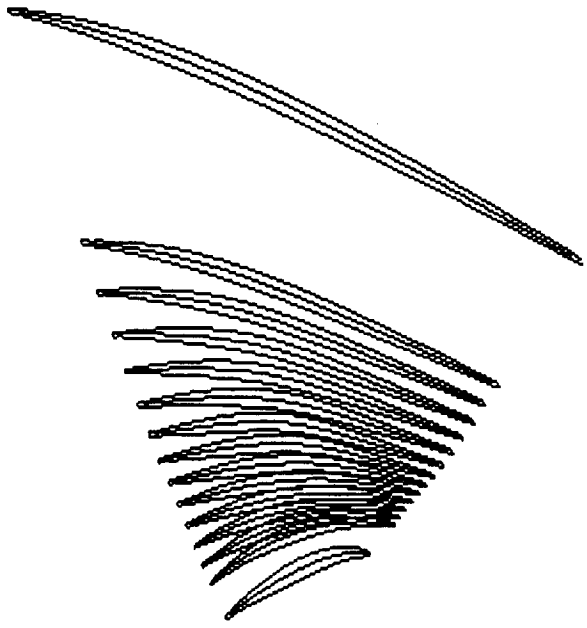

Figure 6.2 Blade section cuts and mean camber lines for example blade 1

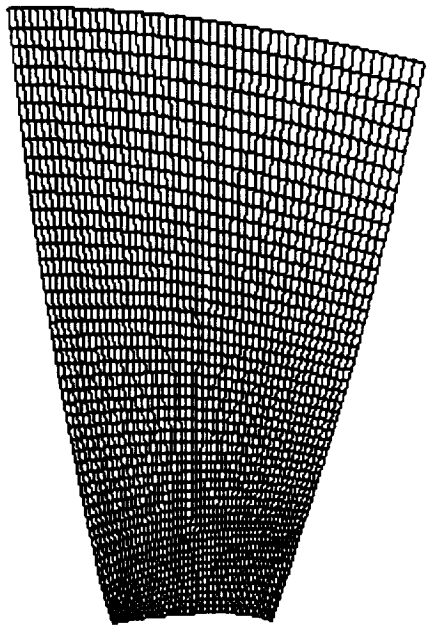

Figure 6.3 Mean camber surface for example blade 1

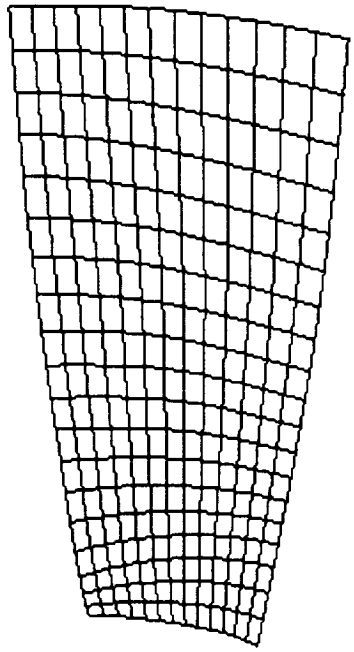

Figure 6.4 Mean camber finite element mesh for example blade 1

Example Blade 2:

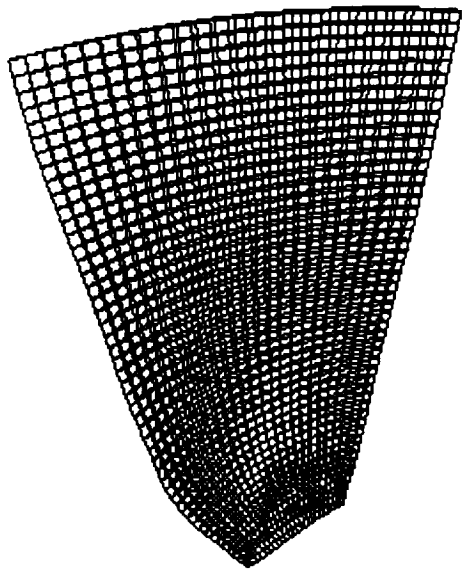

Figure 6.5 Hot-shape blade geometry for example blade 2

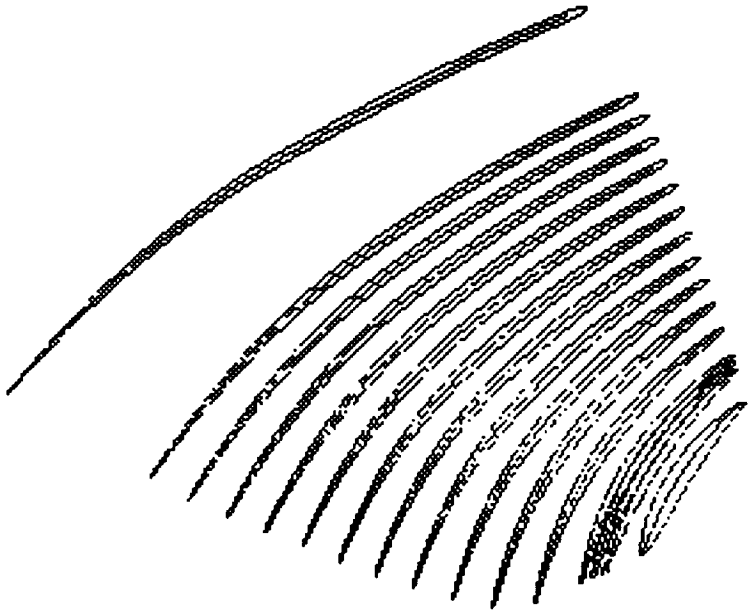

Figure 6.6 Blade section cuts and mean camber lines for example blade 2

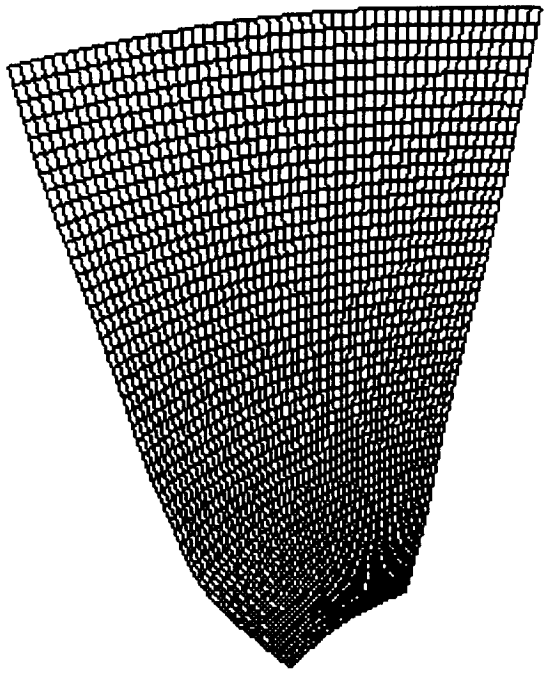

Figure 6.7 Mean camber surface for example blade 2

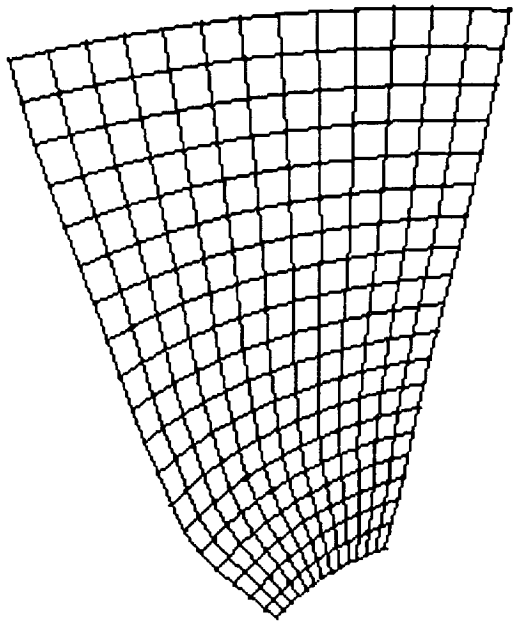

Figure 6.8 Mean camber finite element mesh for example blade 2

6.2 Cold-Shape Correction Examples

The cold-shape correction is demonstrated in Figures 6.9, 6.10, and 6.11. Figure 6.9 shows the results of the cold-shape correction method on example blade 1 presented in Figures 6.1-6.4. Figures 6.10, and 6.11 show the results for the cold-shape correction method on example blade2 presented in Figures 6.5 - 6.8. For reference, the original hot-shape blade geometry is shown along with the resulting cold-shape geometry. The cold-shape displacements have been multiplied by a scale factor so that the shape distortion is obvious.

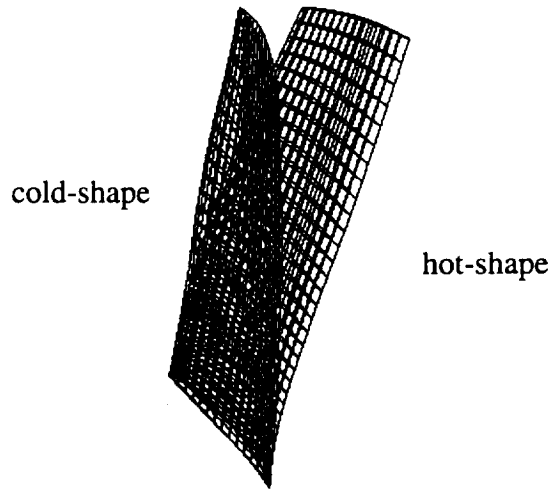

Figure 6.9 Cold-shape correction for example blade 1 with a distortion scale factor of 5

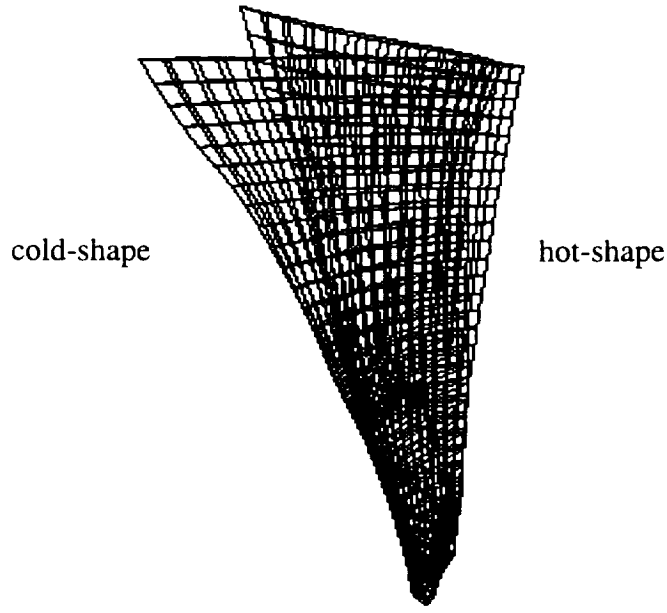

Figure 6.10 Cold-shape correction for example blade 2 with a distortion scale factor of 10

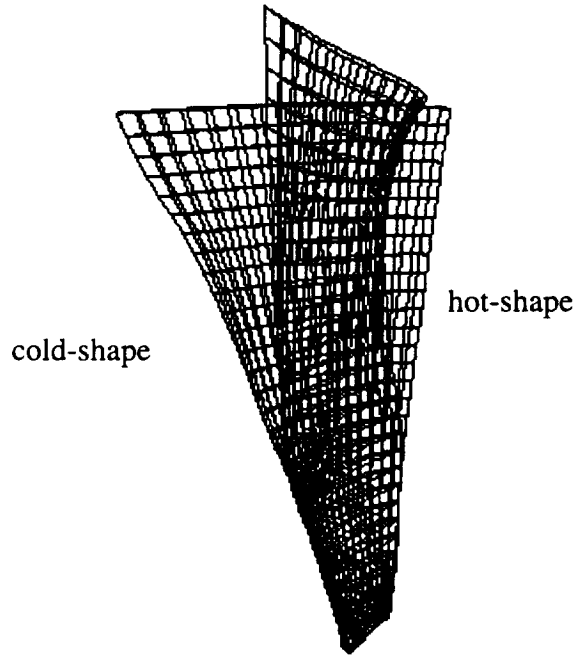

Figure 6.11 Cold-shape correction for a example blade 2 with a distortion scale factor of 20

CHAPTER 7. CONCLUSIONS

The integrated and interactive cold-shape correction method provides an effective tool for the design, analysis and manufacture of turbomachinery blades. Given blade hot-shape geometry in the IGES format the technique generates a NURBS based representation of the cold-shape geometry which may be output in the IGES format. The method is composed of two main components: a preprocessor that generates a plate finite element model of the hot-shape geometry, and a postprocessor that generates the cold-shape corrected blade. The pre-processor decomposes the hot-shape blade surface model into a mean camber surface and an associated set of thickness functions. A plate finite element mesh is generated on the resulting mean camber surface. The finite element model is used as input for specialized analysis software for cold-shape correction which provides displacements due to unloading the blade. The postprocessor constructs the nominal cold-shape blade model in two steps. First, the nodal deflections are applied to the hot-shape finite element model to generate a cold-shape mean camber surface. Then the original hot-shape thickness functions are applied to the cold-shape mean camber surface to generate characteristic blade section curves which are lofted to define the cold-shape blade model. The shape correction method is more accurate and efficient than currently available techniques. Both the finite element mesh generation and cold-shape displacement application are greatly improved. Also, the method takes many of the manual operations out of the cod-shape correction process.The method has been successfully applied to several industrial examples.

Future research will improve the capabilities of the integrated shape correction method. The upgrades to the technique will include mid-span shroud capabilities, the ability

to accommodate blade loading, and three dimensional mesh generation capabilities. The mid-span shroud capability will permit input of NURBS blade models with external supports, i.e. mid-span shrouds. This task will focus on methods to accommodate mid-span shroud geometry within a mesh. The blade loading ability will be integrated by utilizing DT_NURBS capability to allow an arbitrary number of dependent number of dependent variables, which will allow loads, such as temperature and pressure, to be incorporated into the surface representation itself. Finally, the preprocessor will allow the generation of three dimensional solid elements which will accommodate internal voids and other three dimensional shape details.

BIBLIOGRAPHY

Barnhill, R.E. and Kersey, S.N., 1990, "A marching method for parametric surface/surface intersection," *Computer Aided Geometric Design*, 7 (1990), pp.257-280.

Böhm, W., Farin, G. and Kahmann, J., 1984, "A Survey of Curve and Surface Methods in CAGD," *Computer Aided Geometric Design*, Vol. 1, pp. 1-60.

Cox, M., 1972, "The numerical Evaluation of B-splines," *Journal of the Institute of Mathematics Applications*, Vol. 10, pp. 134-149.

de Boor, C., 1972, "On Calculating with B-splines," *Journal of Approximation Theory*, Vol. 6, pp. 50-62.

deBoor, C., 1978, *A Practical Guide to Splines, Applied Mathematical Sciences*, Vol. 27, Springer-Verlag, New York

Ernst, M.A., 1992, "Structural Analysis of Low-Speed Composite Propfan Blades for the LRCSW Wind Tunnel Model," *NASA Technical Memorandum* 105266, NASA Lewis Research Center, Cleveland, Ohio.

Farin, G., 1993, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, San Diego

Kriezis, G.A., Patrikalakis, N.M. and Wolter, F-E, 1992, "Topological and differential equation methods for surface intersections," *Computer-Aided Design*, Vol. 24, No. 1, pp. 41-55

Nair, N.K. and Oliver, J.H., 1994, "Geometric Design of Turbomachinery Blades on General Stream Surfaces," submitted to *ASME Winter Annual Meeting*, Chicago, IL, November

NIST, 1991, "The Initial Graphics Exchange Specification (IGES) Version 5.1," National Institute of Standards and Technology, Gaithersburg, Maryland.

Patrikalakis, N.M. and Bardis, L., 1992, "Feature Extraction from B-spline Marine Propeller Representations," *Journal of Ship Research*, Vol. 36, No. 3, pp. 233-247

Patrikalakis N.M. and Prakash, P.V. 1990, "Surface Intersections for Geometric Modeling," *ASME Transactions, Journal of Mechanical Design*, Vol. 112, pp. 100-107

Piegl, L., 1991, "On NURBS: A Survey," *IEEE Computer Graphics and Applications*, Vol. 11, No. 1, pp. 55-71

Thorp, S.A. and Downey K.M., 1992, "Computer Aided Design and Manufacturing of Composite Propfan Blades for the Cruise Missile Wind Tunnel Model," *NASA Technical Memorandum* 105269, NASA Lewis Research Center, Cleveland, Ohio.

US Navy, 1993, "DT_NURBS Spline Subprogram Library Users' Manual," Naval Surface Warfare Center, David Taylor Model Basin, Bethesda, MD

PART II

Turbomachine Blade Structure Loading Correction and Analysis

D.C. Rickert and J.H. Oliver

CHAPTER 1. INTRODUTION

1.1 Background

Turbomachinery is in widespread use throughout the world. Applications such as power generation, aircraft, ships, natural gas distribution systems, and even automobiles make use of rotating fluid machinery in multiple ways. Billions of dollars are involved in equipment, maintenance and operating costs for this type of machinery. Given these widespread applications and capital investments, very beneficial economic effects can be gained from improvements in efficiency, affordability, reliability, and weight and noise reduction.

The cost of implementing new technology in propulsion systems is becoming prohibitively expensive. One of the major contributors to the high cost is the need to perform many large scale system tests. Extensive testing is used to capture the complex interactions among the multiple disciplines (e.g., fluid dynamics or structural dynamics) and the many components in complex systems. The National Aeronautics and Space Administration (NASA) is pioneering the effort of developing computational tools for turbomachinery design, analysis and manufacture. NASA's aim is to allow for comprehensive evaluation of new concepts early in the design phase before a commitment to hardware is made. The culmination of this effort, the Numerical Propulsion System Simulator (NPSS), is an interdisciplinary project to unite the various disciplines in turbine design.

NPSS will allow for rapid assessment of field-related problems, particularly in cases where operational problems were encountered during conditions that would be difficult to simulate experimentally. The tremendous progress taking place in computational

engineering, the rapid increase in computing power and visualization tools make this concept feasible.

1.2 Motivation

The traditional design and analysis procedure that is applied to a complex turbine system decomposes the system into isolated components and focuses attention on each single physical discipline (e.g., fluid dynamics or structural dynamics). Consequently, the interactions that naturally occur between components and disciplines can be masked by the limited interactions that occur between individuals or teams doing the design and analysis. This can pose serious problems for a highly integrated propulsion system, where tight coupling can produce unforeseen interactions having adverse effects on system performance. If the coupling is not identified until the system has been built and tested, then the system must undergo redesign and retesting. Typically, several iterations of the design-build-test cycle are required before desired performance is achieved. This is an extremely costly and time consuming process. As a result, the introduction of advanced technology takes many years as these systems slowly evolve. The need exists to reduce the time and cost associated with introducing new technology.

Blade geometry is one of the most critical elements of turbomachinery manufacture. Very precise geometric definitions for the operating geometry is specified by an aerospace designer. Even slight deviations from a blade's precise definition can severely affect engine performance. The blade geometry, when subjected to operating loads, is often referred to as the operating geometry. Conversely, the blade's geometry in its unloaded state is often referred to as the manufacturing geometry. The ability to manufacture a blade geometry that will have the intended design shape under fully loaded operating conditions has been limited. For instance, surface pressures derived from an aerodynamic analysis may be used in a structural analysis of the blading. However, bi-directional coupling effects are rarely introduced in the overall analysis process. With

the advent of lighter and stronger materials, which deform severely under loading, accurate shape correction for the manufacturing geometry can greatly reduce cycle time of the finished product.

In addition, data management capabilities must be provided that can persistently represent the complex relationships that exist between the analysis data. For instance, a finite element analysis model is typically derived from a particular version of blade geometry, and the analysis model has multiple analysis result states, each associated with different operating conditions. The design of turbomachinery is often a trade-off between efficiency and stall at both design and off-design operating conditions. While the compressor must operate efficiently at design conditions it must also be able to start, idle, and transition from one condition to another. Further, the compression system must perform with a sufficient safety margin between it's expected range of operation and compressor stall.

The intent of this project is to provide an efficient tool to facilitate blade analyses, and to assess the rotor's structural performance. The analysis is carried out by first performing a rotor internal secondary flow analysis to generate rotor internal pressures and temperatures. These results then are applied, along with rotational speed and the outer diameter pressures and temperatures, to compute the structural response of the blade. The structural analysis is performed using a finite element analysis program, which can create single or multiple load result states. In order to smoothly interpolate analyses results between manufacturing and operating conditions, the displacements field is interpolated with a tri-parametric B-spline referred to as a hyperpatch. The hyperpatch can be applied to the blade's undeformed NURBS representation to determine the blade's manufacturing geometry, off-design geometry, and a wide range of auxiliary structural analysis results, such as blade tip clearances, blade twist, and blade radial growth.

The hyperpatch allows the designer to generate a NURBS surface representation of the deformed blade, which includes the manufacturing or any off-design geometry within the range of the finite element analysis. The designer can scan through the range of loading conditions and observe immediate visual feed back of the deformed blade at any displacement desired. The off-design geometry can give the designer insight in to the machine's performance as it transitions from one operating point to the next, potentially saving the design process a costly build and test cycle if the off-design geometry proves inadequate.

The distance between a rotor blade and the casing, or a stator blade and the rotor, has a substantial effect on the efficiency and stall characteristics of a blade row. This distance is called tip clearance and is a function of the operating point and the system structural response. The running tip clearance is dependent on the structural and aerodynamic design of the system, and in most cases is not accurately known. In many ways, accurate prediction of tip clearance is the essence of stall line management.

A blade's twist angle characterizes its shape relative to its axis of rotation. The performance of the fluid flow over the blade is dependent on the blade twist, thus the performance of the blade will vary due to structural displacement. The variation in loading associated with the transition from one operating point to the next, produces a blade twist that is dynamically changing, and usually unknown. Blade twist gives some indication of the pressure distribution over the surface of the blade, and can indicate operating ranges where problems could occur.

Improved turbomachinery design resulting from high performance computing promises to give the U.S. aerospace industry a critical competitive edge. No software tools are

available to automatically construct a NURBS representation of the blade's deformed shape from the finite element displacement results. This project addresses this need through the development of analytical methods and their implementation in software to integrate analysis results with geometric blade models. This allows displacement results to be applied to the undeformed blade NURBS surface representation to determine the blade's manufacturing geometry, and any off-design geometry with-in the operating range modeled by the finite element analysis.

CHAPTER 2. BLADE DESIGN METHODOLOGY

A clear understanding of some basic concepts of blade design is paramount to understanding the need for, and the usefulness of this work.

2.1 Traditional Blade Design

Traditional blade designers use unique skills to determine optimal blade geometries. Blade designers represent the blade geometry as a grid of points on the surface of the blade or with non-standard algebraic equations [1]. Axial blade design typically begins with two-dimensional stream line curvature through-flow algorithms that aid in determining the blade cross-section geometry from aerodynamic requirements [2]. Input to these includes overall aerodynamic performance and basic blade section information. Output is generally a grid of points on the blade surface. The restrictions of these programs are that they will only generate certain types of blade geometry (i.e., axial compressor blades).

Blades are typically defined by a series of cross-sectional profiles stacked at several radial locations. The profiles (or blade sections) are generally developed on planes, or on concentric cylinders of cones, which are intended to approximate the path of the fluid flow through the turbomachine. The blade sections are related to these surfaces and to one another by a stacking axis, which is generally limited to a line segment.

There is currently no application available that allows geometric manipulation in an intuitive, user-friendly environment that displays the blade in its three dimensional form, and is general enough for all types of turbomachinery.

2.2 Recent Design Methods

Recent developments in algorithm design allows a general stream surface, which can exactly represent the boundary conditions of the turbomachines's fluid flow. This set of algorithms, developed by Perry L. Miller [3], allows for interactive blade construction on general stream surfaces (surfaces of revolution) and stacking axes, while providing an interface and design methodology familiar to the areo-designer. This method of blade design is motivated by the need to define and modify blade sections within general surfaces of revolution, while providing intuitive interaction techniques.

2.3 Coordinate system

A blade is represented in a cylindrical coordinate system comprised of r- z-, and θ- coordinates. The blade is oriented such that it revolves about the *z*-axis in the θ- direction. For axial turbomachinery components, the *r*-axis is referred to as the blade span direction, and the *z*-axis is referred to as the chord direction. In an axial flow turbomachine, fluid flows past the blade in the positive *z*-direction. The axial path of fluid flow can be represented as a stream curve in the *r-z* plane.

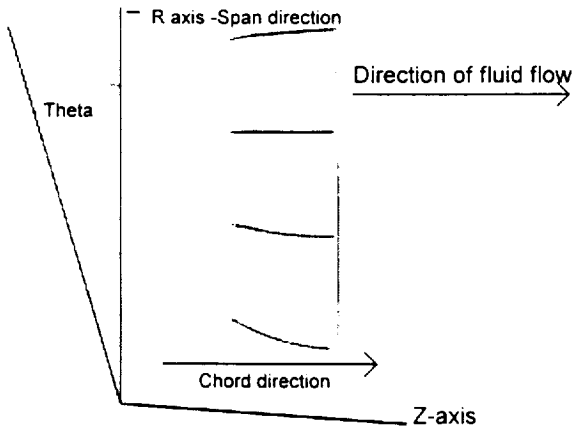

Figure 2.1 Blade in *r-z-theta*

2.4 Stream Curves and Surfaces

A blade model is generated from several blade section curves which are each related to a specific stream surface by a stacking axis (Figure 2.2). A stream surface characterizes the desired axial fluid flow through the machine and is defined by a stream curve in the *r-z* plane. The stream surface can be any general surface of revolution because the stream curve can be any planar curve. The use of stream curves stems from Bernoulli's principle [4] named after the Swiss mathematician and Scientist Daniel Bernoulli. The principle states that the total mechanical energy of an incompressible and inviscid flow is constant along a streamline. Typically, several stream curves are required to define a blade. Figure 2.2 shows several stream curves and one stream surface. Several stream curves which are used to develop stream surfaces on which blade sections will be defined. As shown in Figure 2.2 a stream surface is generated by revolving a stream curve around the *z*-axis in the theta direction. The stream surface characterizes the desired axial fluid flow through the machine, as defined by a stream curve.

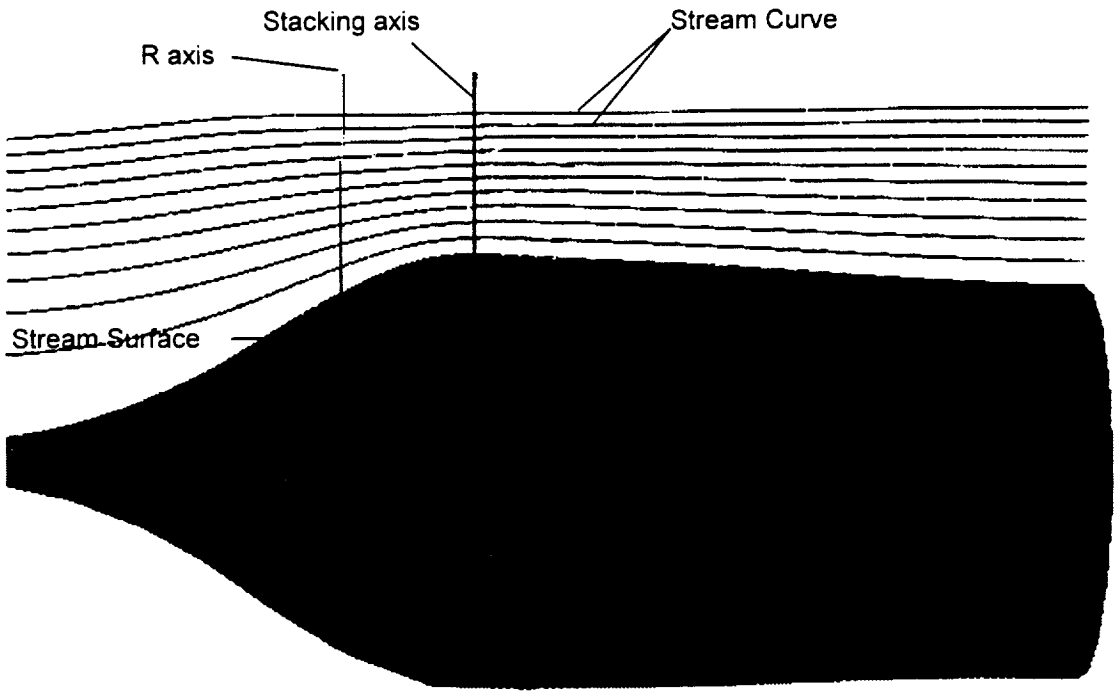

Figure 2.2 Stream curves and surface

2.5 Stacking Axis

A blade model can be generated from several blade section curves which are each related to a specific stream surface by the stacking axis. The stacking axis can be any general space curve. The only requirement is that the stacking axis intersect all stream surfaces. The stacking axis in this method characterizes the placement of the blade sections relative to their corresponding stream surfaces.

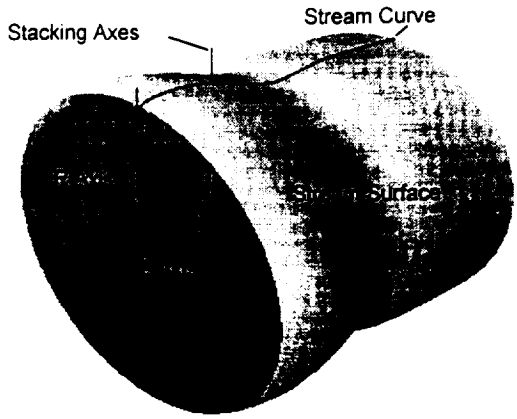

Figure 2.3 Stacking Axis with stream information

2.6 Blade Sections and B-Splines

Regardless of the design method, the blade's final surface is generally computed by skinning the blade sections, or directly interpolating the grid of surface points [5] . Skinning is the process of interpolating a surface through a series of roughly parallel curves, which is analogous to the process of placing a hull (skin) on a ships keel frame (blade sections). In Millers method, the blade section geometry is the intersection of the *intended* blade with the stream surface.

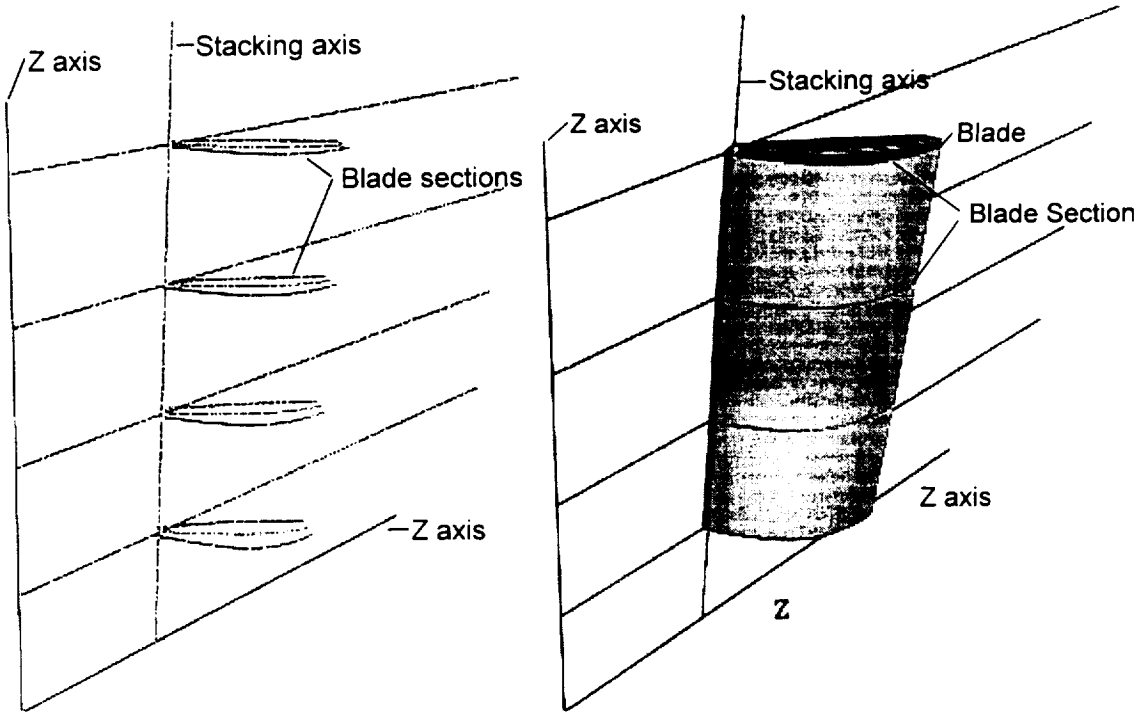

Figure 2.4 Section to blade view

In this project, blades, blade sections, stream curves, and stream surfaces are represented as non-uniform rational B-splines (NURBS). The NURBS , representation, provides a precise mathematical description for geometry, that is easily manipulated and analyzed. The basic definition of B-spline curves and surfaces can be found in most geometric modeling textbooks, (e.g., Farin, 1993; Mortenson, 1985, deBoor, 1978) or in any of the many survey articles on the subject (Bohm et al., 1984; Piegl, 1991). The following provides a brief description of B-splines to facilitate a better understanding of the methods developed in this project.

2.7 NURBS Curves and Surfaces

B-spline curves are characterized as a linear combination of points, called control points. The basis functions defining the linear combination, referred to as B-splines, are typically generated via the Cox-deBoor algorithm [6]. Given knot vector $U = \{0,\ldots,0,u_{p+1},\ldots, u_{n-p-2},\ldots,1,\ldots 1\}$ a monotonically increasing sequence of real numbers, the ith B-spline

basis function of degree p, denoted $N_{i,p}(u)$, is defined by the recursive relationship shown in equation (2).

$$N_{i,0}(u) = \begin{cases} 1 & if \quad (u_u \le u \le u_{i+1}) \\ 0 & otherwise \end{cases} \qquad (1)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (2)$$

where it is assumed that 0/0 = 0.

The knot vector governs the relationship between parametric and spatial variation, and its entries represent the parameter values at segment joints (knots). A non-periodic B-spline is characterized by a knot vector in which the first and last knot values are repeated $p+1$ (order) times. This results in a curve that interpolates the first and last control points.

A B-spline curve, of degree p, is defined as follows,

$$C(u) = \sum_{i=0}^{n} N_{i,p} P_i, \qquad 0 \le u \le 1 \qquad (3)$$

where P_i are the control points and the $N_{i,p}(u)$ are the pth degree B-spline functions, defined above, and a non-periodic knot vector $U = \{0,\ldots,0,u_{p+1},\ldots,u_{n-p-2},\ldots,1,\ldots 1\}$. The degree p, number of knots m, and the number of control points are related by formula (4):

$$m = n + p + 1. \qquad (4)$$

2.8 B-Spline Surfaces

A B-spline surface of degree (p,q) is specified by an $(m\text{-}p)$ X $(n\text{-}q)$ grid of control points P_{ij} arranged in a topologically rectangular array and knot vectors U and V of length $(m+1)$ and $(n+1)$, respectively. The surface, denoted as $S(u,v)$, is thus defined as the tensor product of the control point array and the B-spline basis functions defined over each knot vector:

$$S(u,v) = \sum_{i=0}^{(m-p-1)} \sum_{j=0}^{(n-q-1)} N_{i,p}(u)\, N_{j,q}(v)\, P_{i,j} \qquad (5)$$

2.9 Curve Interpolation

The curve interpolation problem can be solved as follows. Given a set of data points Q_k, where $k = 0,\ldots,n$, a B-spline curve is desired that for certain parameter values u_k,

$$Q_k = C(u_k) = \sum_{i=0}^{n} P_i\, N_{i,p}(u_k) \qquad (6)$$

To solve this equation, the parameter values u_k corresponding to the data points are required, as well as the knot vector, and the degree of the curve. One of the several methods to compute the parameter values is the centripetal method [5]. The cetripital method allows curves to fit tighter corners with fewer interpolant points than other methods such as chordal, or uniform. The centripital parameters are found as follows.

$$u_0 = 0, \qquad u_i = u_{i-1} + \frac{\left|Q_i - Q_{i-1}\right|^{1/2}}{\sum_{j=1}^{n-1}\left|Q_j - Q_{j-1}\right|^{1/2}}, \qquad u_n = 1 \tag{7}$$

Given the parameter values, a knot vector is needed that reflects the distribution of these parameters. The following averaging method is commonly used, to determine interior knots

$$u_j = \frac{1}{p}\sum_{i=j}^{j+p+1} u_i \quad j = 1,\ldots,n-p \tag{8}$$

It can be proved that the coefficient,

$$N_{i,p}(u_k)\big|_{i,k=0,\ldots,n} \tag{9}$$

is totally positive and with a bandwidth less than or equal to $p + 1$. Therefore, the linear system can be solved by Gauss elimination.

2.10 Skinning

Blade surface geometry is typically created by *skinning* blade cross-sections. Given a set of NURBS curves, a NURBS surface can be constructed to interpolate these curves. The one restriction of these curves is a common knot vector. The curves can either be constructed with a common knot vector, or their knot vectors may be merged after construction. The knot merging technique increases the size of each curve definition, commonly making each curve definition equal in size to the sum of all the original curves

in the merging process. To avoid knot merging, a common parameter vector, u_k, is developed for all the curves prior to interpolation.

To create a common parameter vector, the number of interpolant points for each curve must be equal. As described in section 2.8, a parameter vector, u_k, *is* developed for each set of curve points based on the centripital method. The elements of the parameter vectors are then averaged, creating a common parameter vector. The common parameter vector is then used to create a common knot vector. Each set of curve points is then interpolated as describe in section 2.8 using the common parameter and knot vectors.

Consider a set of (n+1) NURBS curves that have the same degree p and knot vector U:

$$C_k(u) = \sum_{i=0}^{n} N_{i,p}(u) Q_{i,k} \qquad (10)$$

The control points for these curves ($\boldsymbol{Q}_{ik}$) make an (n+1) by (m+1) grid of points (shown as black and gray points in Figure 2.5. Calculating a knot vector V with degree q in the v-direction (as described above) and interpolating the control point grid in the v-direction yields,

$$C_i(u) = \sum_{j=0}^{m} N_{i,q}(u) P_{i,j} \qquad (11)$$

The control points for these curves ($\boldsymbol{P}_{ij}$) also make an (n+1) by (m+1) grid of points (shown as gray and white points in Figure 2.5). Evaluating this grid of control points ($\boldsymbol{P}_{ij}$) with knot vectors U and V, and degrees p and q, yields a surface $S(u,v)$ that interpolates

the original curves $C_k(u)$. The curves $\mathbf{C}_k(u)$ are isoparametric curves (constant v) in the u-direction on the surface $S(u,v)$.

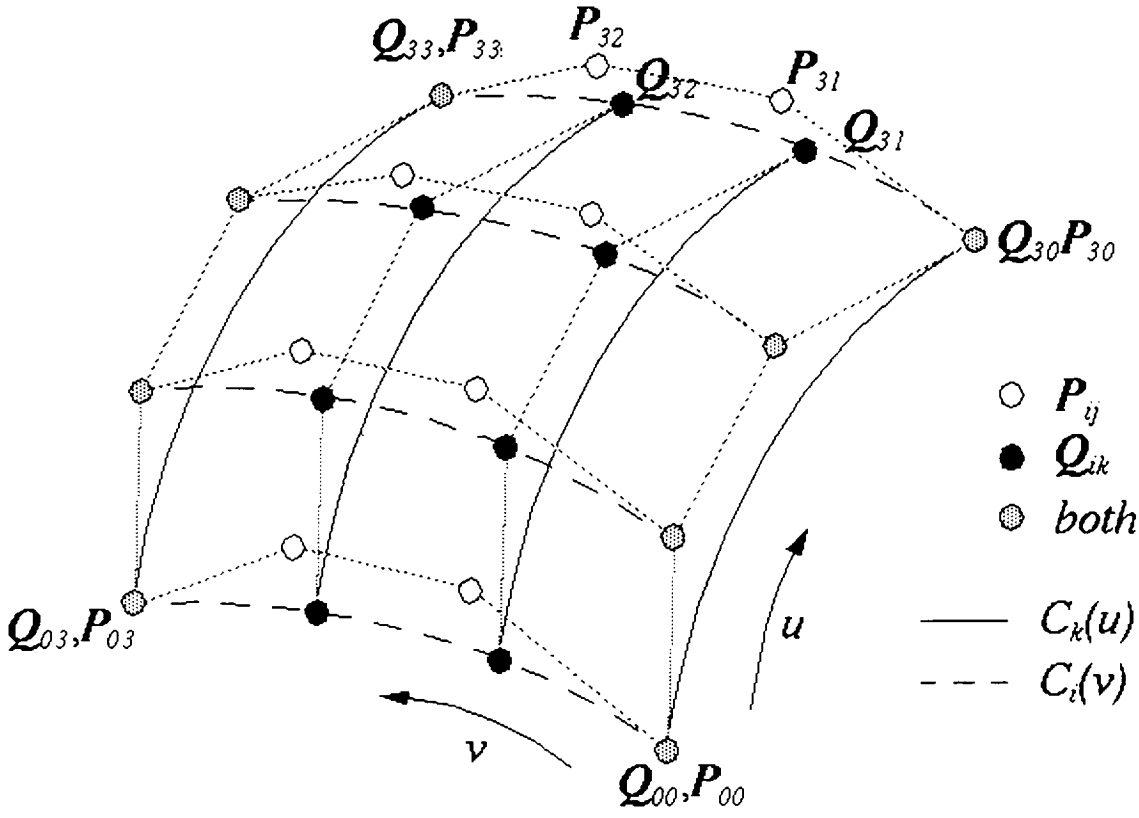

Figure 2.5 Control point grid

2.11 Implementation

Despite their thorough documentation in the open literature, computational implementation of general purpose NURBS modeling techniques is a large and complex undertaking. Thus, this research takes full advantage of the DT_NURBS Spline Library provided by the Naval Surface Warfare Center [9]. This library provides a complete and robust collection of NURBS-based modeling and analysis functions and is available for public distribution in the United States.

CHAPTER 3. BLADE PREPROCESSING

The blade preprocessing methodology presented in this paper is a continuation of the pioneering work done by Brian Hines [7]. The original blade preprocessing algorithm, while accurate, was computationally expensive, which impaired the interactive nature of the software specifications.

Blade preprocessing requires the development of model suitable for finite element analysis (FEA). Many finite element methods performed on blades require the use of plate elements. Plate elements are characterized as triangles or quadrilaterals with thickness defined at each vertex (node). Thus for blades this motivates the need to compute a mean camber surface from the blade surface.

Hines original method for determining the mean camber curve involved intersecting the blade geometry with stream surfaces to develop a set of blade sections. The surface/surface intersection to find the blade rows is a Newton/Raphson search. For each intersection curve the mean camber line must be calculated. The result of the surface-surface intersection is a three-dimensional (space) curve, but the mean camber algorithm, used by Hines, requires a closed planar curve as input. Thus an angle preserving transformation is developed to map the space curve onto a plane. The transformation is generated from a two-dimensional curve in the r-z-plane which defines a surface of revolution about the z-axis. The transformation maps an r-z an r-z-curve onto a one-dimensional coordinate (m') equivalent to meridional arc length normalized with respect to radius [7]. Hines procedure, while very accurate, was computationally expensive, and a faster method was sought.

3.1 Mean camber technology

As with Hines work, the goal is to develop a plate element model for FEA analysis. The plate model for a blade is its mean camber surface which can be considered the skeletal surface of the blade. The procedure for mean camber surface construction involves extraction of several characteristic blade profile curves. Figure 3.1 shows a blade cross section, with blade surface points lying on mean camber surface normals at $t/2$ and $-t/2$, where t is the thickness of the blade. The following procedure takes advantage of section skinning and de-engineers the geometry to calculate the needed information.

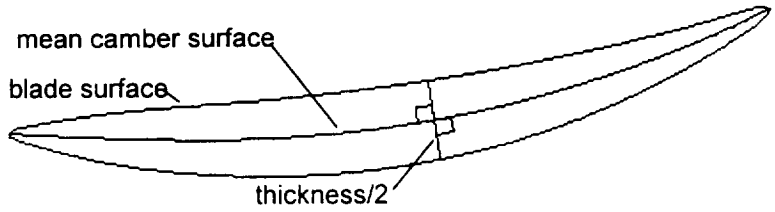

Figure 3.1 Mean camber surface definition

Step 1.

The process of mean-camber production starts by extracting chord-wise iso-parameter curves along the stacking axes of the air-foil surface, including the root and the tip, section curves corresponding to $v = 0$ and $v = 1$ respectively. The number of parameters is based on the span-wise knot vector: Knots are averaged based on the degree of the curve to yield more parameter values than internal knots. These parameter values are generally referred to as the "nodes" of the B-spline [5] curve (unrelated to the FEA term). The nodes t_i of a B-spline curve are given by;

$$k_i = \frac{1}{p}\sum_{j=1}^{p} u_{i+j} \qquad i = 0,\ldots,n \tag{12}$$

Using the span-wise nodes, iso-parameter curves (blade sections) are generated from the surface definition as follows.

$$C_{v_j}(u) = S\left(u, v_j\right) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,n}(u)\, N_{j,m}\left(v_j\right) P_{i,j}$$

$$= \sum N_{j,m}(u)\left(\sum_{i=0}^{n} N_{i,n}\left(v_j\right) P_{i,j}\right) = \sum_{j=0}^{m} N_{j,q}(u)\, Q_j\left(v_j\right) \qquad (13)$$

$$Q_j\left(v_i\right) = \sum_{i=0}^{n} N_{i,p}\left(v_i\right) P_{i,j} \qquad (14)$$

Where $Q_j(v_i)$ are the curve control points, using the knots from the blade's chord-wise knot vector.

Step 2.

The leading and trailing edges of each iso-parameter curve must be found. These edges are defined as the two points of minimum radius of curvature. The search for these points starts by finding a set of parameter values that yield successive chordal line segments which deviate from the curve by a user-specified maximum. Since the advent of numerically controlled machines many algorithms have been developed for this purpose. One such algorithm that uses small steps to approximate the curve by its osculating circle, so that the local curvature of the intersection curve may be used to determine the step length.

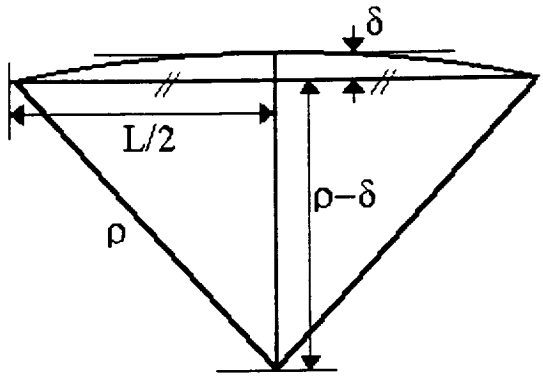

Figure 3.2 Curve Approximation

By Pythagora's theorem, we may show that

$$L^2 = 4*\delta*(2*\rho-\delta) \quad (15)$$

where L is the step length, p is the radius of curvature, and δ is the chord height.

The algorithm used to implements the search, DTSCHT, returns an array of parametric values corresponding to equal chordheights. The curve is then evaluated at these points and a cordlength is then determined for each segment. The lengths are compared, with the two smallest distances being the bounds of the leading and trailing edge points. The parametric boundaries of these two line segments are subdivided, yielding the parameters of the blades leading and trailing edge.

Step 3.

Once the parametric values of the leading and trailing edges are identified, knot insertion is used at those locations until the two knots are of degree equal to that of the curve. The repeated knots create areas suitable to break the spline into two separate NURBS entities.

Step 4.

At this step each of the two curves are divided into equal arc length segments using another algorithm called DTEQAR. The number of segments used is determined by the

combined number of control points of both curve, thus relating the procedure to the amount of definition available in the curves. This algorithm returns an array of parameters, which yield points of equal arc lengths on both curves. Corresponding points on both curves are joined with line segments which are bisected to yield points on the mean-camber of the blade points on the mean-camber curve of the blade. This method is commonly used in industry, and is found to provide an accurate mean camber surface. Figure 3.3 shows a blade section dissected as described.

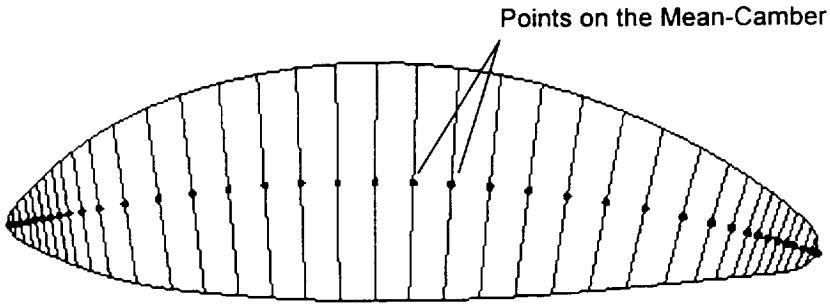

Figure 3.3 Mean camber points

Step 5.

The mean-camber points for each blade section are then interpolated with a cubic NURBS curve. These curves are then skinned to form a mean-camber surface.

After a mean-camber surface is found the production of a subrange and thickness map can begin, as described in the next chapter.

3.2 FEA analysis

Plate elements are typically required for the FEA analysis. A triangle or quadrilateral mesh should be developed using the mean camber surface. Plate elements require the

location of each vertex (node), with a corresponding thickness. For accurate shape correction, the parametric locations of the vertices must be stored for post processing. The reason for storing these values for accurate manufacturing shape correction will become apparent in the *Post Processing* section.

Using this model, FEA analyses are performed on the blade with a variety of different operating conditions. For each analysis, centrifugal loading along with pressure and temperature information are provided to the FEA analysis. One of the results of the FEA analysis, are the blade displacement files. The displacement file contains several grids of vectors which indicates how the operating blade will deform at different speeds.

CHAPTER 4. BLADE POST PROCESSING

4.1 Introduction to post processing

The goal of post processing is to use the FEA analysis to reconstruct the full blade geometry at any operating point, and provide structural response analysis such as blade twist, radial growth, etc. The data developed by the FEA analysis has multiple result states reflecting a range of operating conditions. Each result state is a separate displacement field covering the mean camber surface at a specific speed. In order to smoothly interpolate analyses results between manufacturing and operating conditions, the displacement fields are interpolated with a tri-parametric B-spline referred to as a hyperpatch.

4.2 Displacement Hyperpatch

Blade post processing takes advantage of a four dimensional NURBS concept called a hyperpatch. Conceptually a hyperpatch can be considered as an extension of the NURBS surface. A NURBS definition of a surface or vector field, can describe any number of dependent variables, usually *XYZ*, as a function of two independent parameters. A NURBS solid or hyperpatch, can also describe any number of dependent variables, but it uses three independent parameters. The advantage of the third independent variable is that it allows smooth interpolation between complete surfaces, or complete vector fields. The hyperpatch application is often used to describe a surface changing over time, but for this application, it is a vector field changing over an operating speed range. The advantage of using a hyperpatch is that it provides a smooth and continuous definition of how the blade displacements vary throughout its operating range.

Turbine blades are designed by aerodynamicists in a nominal operating configuration, known as the "operating geometry". However, deformations caused by centrifugal loading, pressure forces, and thermal gradients require that the blade be produced in its

"manufacturing geometry" so that the operating geometry is realized after these effects are accounted for.

First, nonlinear displacement FEA analysis is performed on the operating geometry, which incorporates aerodynamic and centrifugal loads associated with the operating design point. Due to the complexity of the dynamics involved with turbine blades, the FEA analysis is usually performed incrementally creating displacement results through out it operating range. The vector displacements file that results from the FEA analysis are developed into a NURBS triparametric vector solid (hyperpatch).

When the displacement hyperpatch is evaluated at a constant parameter value corresponding to speed, it produces a vector surface. This vector surface correlates parametrically to positions on the mean camber surface. Since the FEA mesh generally does not correspond to the original blade sections geometry, blade surface points are generated from the mean camber surface and corresponding displacements is applied to both pressure and suction-side points yielding points on a displaced blade. The blade is then reconstructed via section interpolation and skinning.

4.3 Blade subrange map

The term *subrange* describes a B-spline which, when evaluated, provides parameters specific to another B-spline. The mean camber to surface subrange map is a NURBS definition that contains the parametric location of corresponding blade surface points that are perpendicular to any mean camber surface point. When this subrange surface is evaluated at any parameter value (u_m, v_m), it will produce seven dependent values. Four of these seven values (u_s, v_s, u_p, v_p) are the parameters which, when used to evaluate the blade surface, generate points on the corresponding blade surfaces, where the subscript s refers to the suction side of the surface, and p the pressure side. In addition, the subrange-surface evaluation produces the blade thickness, and may produce pressure and

temperature data if it is available. Figure 4.1 shows a normal to the mean-camber surface with the intersection points on the pressure and suction sides of the blade section.

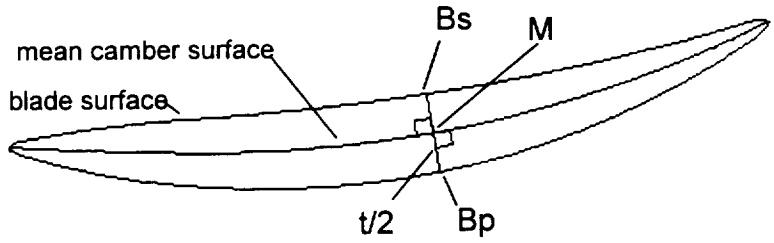

Figure 4.1 Subrange values

Where:

M(u,v) : mean camber B-spline definition.

B(u,v) : blade surface B-spline definition.

Map(u,v) : subrange map surface B-spline definition.

M = M(u,v); A point on the mean-camber surface.

$t/2$, u_s, v_s, u_p, v_p = Map(u,v); corresponding parameter values of the blade surface.

B_s = Blade(u_s, v_s): A coordinate value on the suction side of the airfoil, which is perpendicular to the mean-camber surface at M.

B_p = Blade(u_p, v_p): Point on the pressure side of the air-foil, which is perpendicular to the mean-camber surface at M.

The map's fifth dependent variable contains the thickness of the blade, with the sixth and seventh dependent variables being reserved for pressure and temperature from aerolastic analysis.

The critical design criteria of the map generating algorithm was determining the density of parameters to be fit, relative to their position on the mean camber surface. The map is

used in several functions, but its primary use is for blade reconstruction. To reconstruct an accurate blade surface, care must be taken to insure heavier information density at the rapidly changing blade edges. The best behaved blade definitions generally pack 70% of their information in the 10% of blade geometry which comprise the two edges of the blade. To accomplish this, the map and thickness generating algorithm incorporates the ability to smoothly increase the information density as the edges of the blade are reached. The creation of the mean camber subrange map is outlined in the following steps.

Step 1.

A number of normals are found on the mean camber surface at parameter values developed using the algorithm described above. The number of normals is determined by the original blade definition, since each normal represents a point on either side of the reconstructed blade the number is based on half the number of control points on the original blade. In practice this number was often larger than necessary, due to the fact that most of the original blades were developed in standard CAD systems by lofting blade rows. When blade rows are lofted, knot merging is almost universally used, which explodes the NURBS definition far beyond what is necessary to determine the blade. With the parameter values determined, the normal is found by taking the cross product of the surfaces partial derivatives,

$$n = S_u(u,v) \otimes S_v(u,v)$$

(The DTNURBS function DTSPDR is used to evaluate a NURBS surface and its derivatives.) Knowing the normal and the point at which it was created, a lines is constructed, which will intersect both sides of the air foil surface. The intersections on

both sides of the blade surface is found using DTSLXT, which returns the intersection points in both parametric and Cartesian space.

Step 2.

The intersection points (Cartesian and parametric), and the parameters of the mean camber surface evaluation, are all needed to fit the subrange map surface. Map construction starts with building the knot vectors. In both parametric directions the parameters used to evaluate the mean camber surface normals are used to create the knot vectors as described in section 2.6. The two real coordinates found on each intersection line are used to determine a blade thickness at mean camber surface point. Spline interpolation is used next to fit the four blade surface, the thickness, and any temperature and pressure date that may be available into roughly parallel seven dependent variable curves using DTLSA. The curves are then skinned, as described in section 2.7 using DTCRBL, which develops the subrange map.

With the subrange map, and the vector surface of displacements, the manufacturing geometry or any off-design geometry within the range of the FEA analysis can be quickly generated.

4.4 Blade Reconstruction

When the user wants to view or produce a portable file of the manufacturing geometry, he/she may adjust the slider shown in Figure 4.2. The blade geometry will immediately update, and can be written to an IGES file if desired.

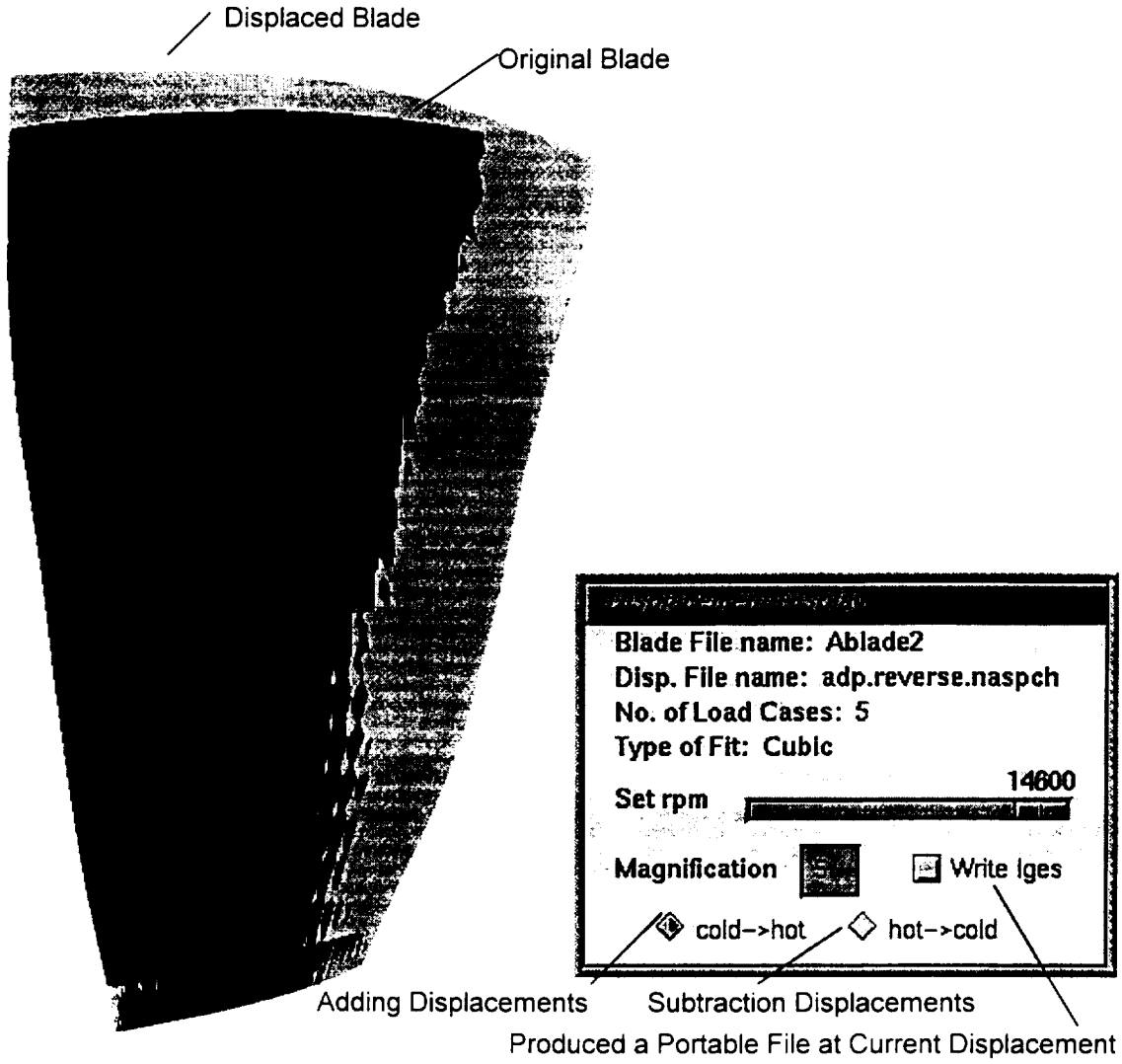

Figure 4.2 Displacing the blade

With the blade-map, and the displacement vector hyperpatch, it is a straight forward process to develop the manufacturing, or off-design geometry. The process starts with evaluating the blade map, blade geometry, and hyperpatch concurrently. The hyperpatch has three parameters, as opposed to only two for the blade-map and blade *B*-spline. The first two parameters of the hyperpatch represents the position on the mean camber surface, with the third parameter of the hyperpatch corresponding to the speed of vector displacements. The blade map evaluation produces parametric values, which are used to

evaluate two points on the corresponding blade *B*-spline. The hyperpatch is then evaluated at the same parametric locations as the blade map, with the third parametric location being set from the selected speed (RPM) setting, generating vector displacements. The vector displacements are then added to the blade surface points, and the process is continued over the entire surface of the blade until enough definition exists to create a displaced blade.

With the capacity to determine the blade's geometry over its operating range the opportunity existed to provide geometric analysis of engineering interests.

4.5 Blade analysis

The design of turbo-machinery is often a trade-off between efficiency and stall at both design and off-design operating conditions. While the compressor must operate efficiently at design conditions it must also be able to start, idle, and transition from one condition to another. Further, the compression system must perform with a sufficient safety margin between its expected range of operation and compressor stall.

Having both a continuous definition of how the blade will deform, and the ability to rapidly apply it to the blade, at descrete speed values, provides the means to do some critical blade analysis. This information can help predict likely operating areas where engine stall may occur, and it can also indicate performance trends such as power output of the blade.

4.5.1Airfoil Twist

Figure 4.3 show Airfoil twist the blade twisting displacement from its base through the tip of the blade. Shown on the graph, no appreciable twist occurs on the blade until 50% span of the blade. The blade twist is loosely related to the power output of the blade, due to the bending moments of the fluid pressure upon the blade.

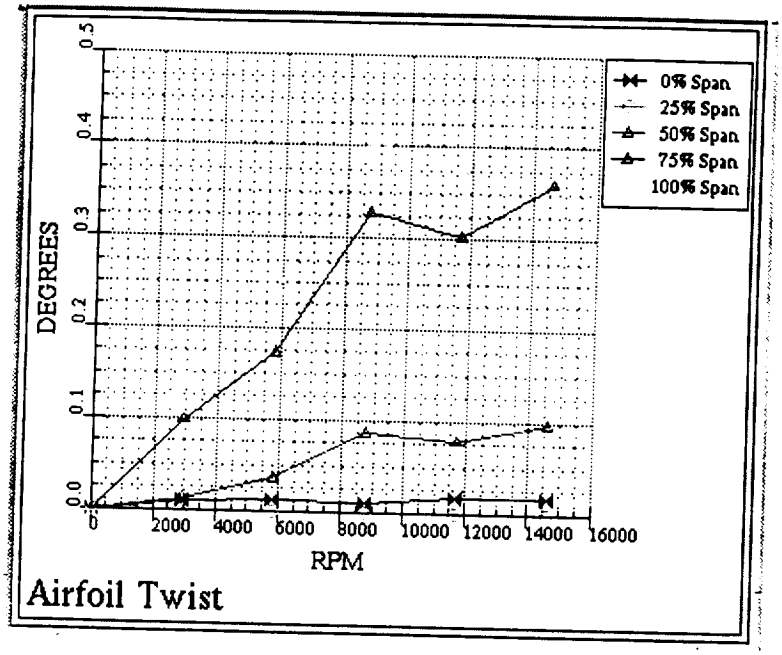

Figure 4.3 Airfoil Twist

4.5.2 Radial Growth

Figure 4.4 shows Radial Growth Points along the tip of the blade extend radially from its axes of revolution. The radial growth of the blade is a function of the centrifugal forces loading the blade. The two main performance features indicated by the blade's growth are related to its tip clearance to the wall. If the tip clearance is too large it can be an indication of engine stall conditions, and if the tip clearance is too small it may indicate a potential for the blade to rub on the flow surface wall.

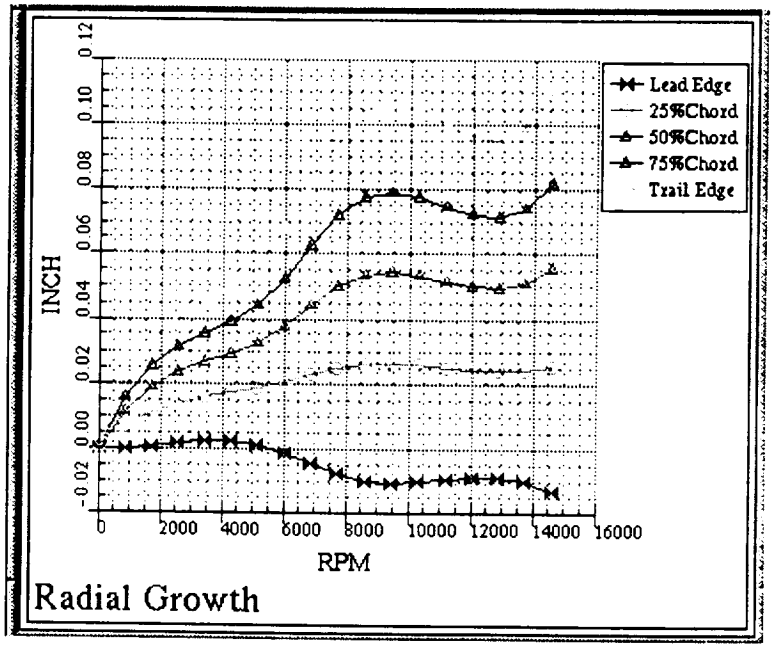

Figure 4.4 Radial Growth

4.5.3 Tip Clearance

If the blades stream surface (Figure 2.3) definition is available, Figure 8.3 can be developed which gives a more accurate representation of how the blade is interacting with its flow surface. The two main performance features indicated by the blade's tip clearance are related to its distance From the wall of its flow surface. If the tip clearance is too large it can be an indication of engine stall conditions, and if the tip clearance is too small it may indicate a potential for the blade to rub on the flow surface wall.

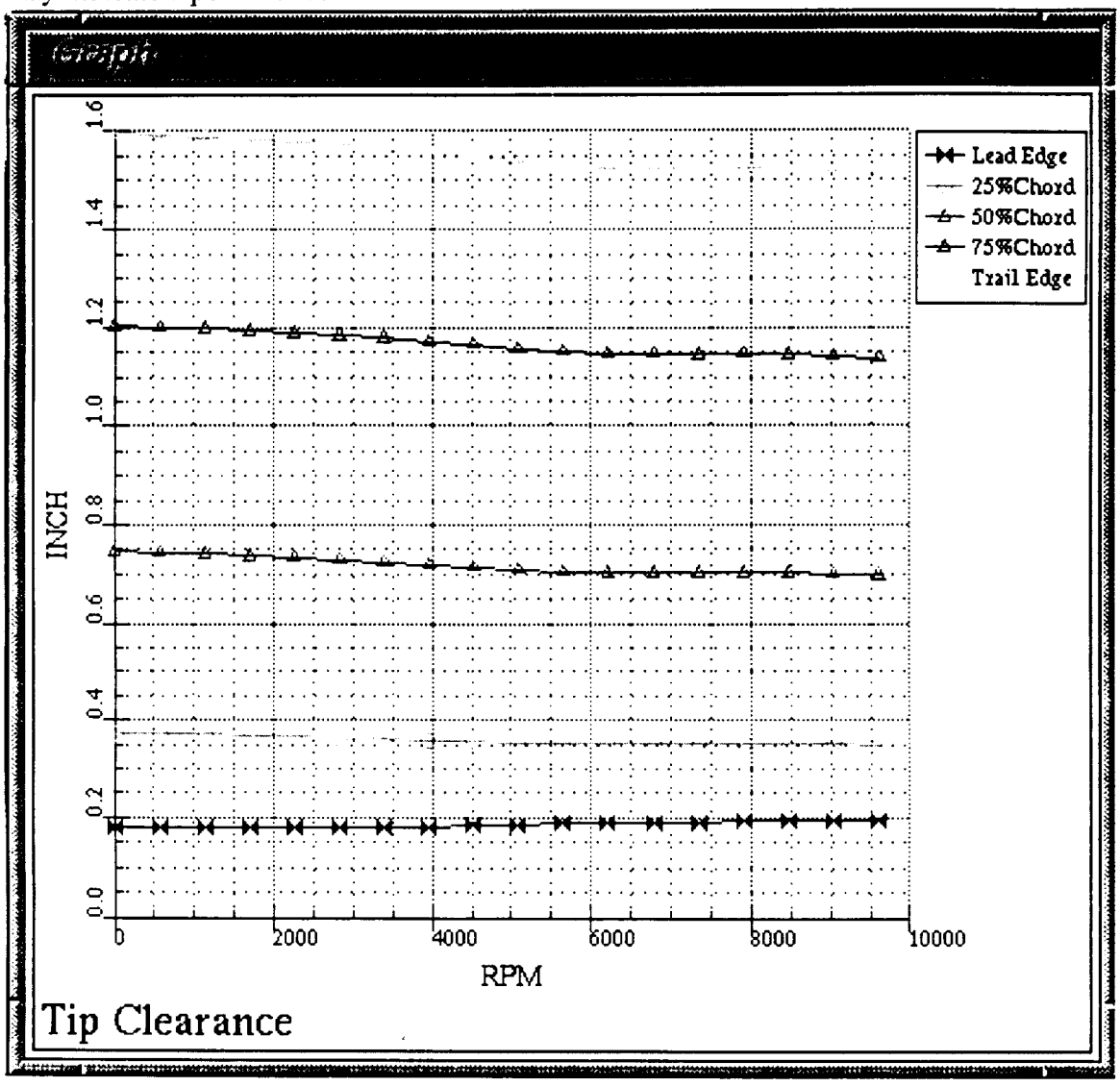

Figure 4.5 Tip Clearance

CHAPTER 5. CONCLUSIONS

This paper describes an effective tool for the design, analysis and manufacture of turbomachinery blades. Given the blade *operating* geometry in IGES format this technique generates a NURBS based representation of the *manufacturing* geometry which may be output in the IGES format. The method is composed of two main components: a preprocessor that generates a plate finite element model of the operating geometry, and a post processor that generates the manufacturing corrected blade. The preprocessor decomposes the operating blade surface model into a mean camber surface and an associated subrange map. A plate finite element mesh is generated on the resulting mean camber surface. The finite element model is used as input for specialized analysis software for shape correction which provides displacement due to loading the blade. The post processor constructs the nominal manufacturing blade model in two steps. First, the deflections are mapped to the operating geometry to generate a point representation of the manufacturing geometry. The blade points are then developed into a NURBS geometry with the same amount of definition as the original operating geometry. The shape correction method is more accurate and efficient than currently available techniques. Both the finite element mesh generation and manufacturing displacement application are improved. Also, the method takes many of the manual operations out of the blade correction process. This software is currently being tested at the NASA Lewis Research center, and is due to be released to industry in October, 1996.

REFERENCES

[1] Crouse, J.E., 1981, "Computer Program for Aerodynamic and Blading Design of Multistage Axial-Flow Compressors," NASA Technical Paper 1946

[2] Law, C.H, and Puterbaugh, S.L., 1982; "A Computer Program for Axial Compressor Design (UD0300m)," Report AFWAL-TR-82-2074, Wright-Patterson AFB, Ohio

[3] Miller, P.L., and Oliver, J.H., 1995, "Interactive Turbomachinery Blade Modeler," ASME 1745

[4] Drysdale, G.R., Ferguson, E,S, 1923, *Bernoulli's use of Stream Curves*, Houghton Mifflin, Washington

[5] Tiller and Piegl, 1995, *The NURBS Book*, Springer Verlag, Bedlin

[6] Cox, M., 1972, "The Numerical Evaluation of B-splines, *Journal of the Institute of Mathematics Applications*, Vol. 10, pp. 134-149

[7] Hines, B.D., and Oliver, J.H., 1994, "Geometric Decomposition and Structural Shape Modification for Turbomachinery Blades," ASME Advances in Design Automation

[8] Faux, I.D., and Pratt, M.J., 1979, *Computational Geometry for Design and Manufacture*, Ellis-Horwood, New York

[9] US Navy, 1993, *DT_NURBS Spline Geometry Subprogram Library Users' Manual*, Naval Surface Warfare Center, David Taylor Model Basin, Bethesda, MD

APPENDIX

ASME Technical Paper

Geometric Decomposition and Structural Shape Modification for Turbomachinery Blades

DE-Vol. 69-1, Advances in Design Automation – 1994
Volume 1
ASME 1994

GEOMETRIC DECOMPOSTION AND STRUCTURAL SHAPE MODIFICATION FOR TURBOMACHINERY BLADES

Brian D. Hines and James H. Oliver
Department of Mechanical Engineering
Iowa Center for Emerging Manufacturing Technology
Iowa State University
Ames, Iowa

ABSTRACT

An efficient and precise method is presented for the generation of turbomachinery blade models in nominal configuration, i.e., the "cold-shape" given the blade geometry at operating conditions, i.e., the "hot-shape." The shape correction technique has two main components: a preprocessor that generates a plate finite element model of the hot-shape geometry, and postprocessor that performs the blade shape correction. The preprocessor decomposes the hot-shape blade surface model into a mean camber surface and an associated set of thickness functions. A plate finite element mesh is generated on the resulting mean camber surface. The finite element model is used as input for specialized analysis software for cold-shape correction which provides displacements due to unloading the blade. The postprocessor constructs the nominal cold-shape blade model in two steps. First, the nodal deflections are applied to the hot-shape finite element model to generate a cold-shape mean camber surface. Then the original hot-shape thickness functions are applied to the cold-shape mean camber surface to generate characteristic blade section curves which are lofted to define the cold-shape blade model. Several examples of turbomachinery blades in their hot-shape and resulting geometry are presented to demonstrate the capabilities of the technique.

INTRODUCTION

Manufacturers of turbomachinery products are under the same competitive pressures driving many industries toward reduced product development time and improved product quality, while simultaneously reducing development costs. The technology underlying the turbomachinery industry has reached relative maturity, so that even small improvements in overall thermodynamic efficiency can provide a competitive advantage. Blade design is one of the most crucial elements of turbomachinery development. As the performance of turbomachinery is enhanced, blade strength-to-weight ratio has increased to such a degree that blade deflections, due to normal operating conditions, cannot be neglected. Blades are generally designed by aerodynamicists in a nominal operating configuration, known as the "hot-shape". However, deflections due to centrifugal force, pressure loading, and thermal gradients require that the blade be manufactured in its so called "cold-shape" so that the hot-shape is realized after accounting for these effects are account. So, given a definition of blade hot-shape geometry, engineers need an efficient and accurate method to calculate the corresponding cold shape geometry.

Current methods for cold-shape correction are approximate in both data extraction and displacement application and they can be very time consuming due to the manual nature of the procedure. Given the hot-shape geometry, the mean camber surface (a surface which is equidistant from both sides of the blade) is generated manually using an approximate approach that bisects line segments between corresponding points on the pressure and suction sides of the blade. Specialized analysis software is employed to calculate the inverse deflections, i.e., those due to removal of the operating boundary conditions (Thorp and Downey, 1992). The inverse displacements are then applied to the hot-shape geometry using an approximation method in which the cold-shape blade surface points are generated from mean camber surface points and weighted averages of the deflections. This method is generally adequate for stiff blades because the deflections (and thus errors) are small relative to the overall size of the blade. But for blades with large deflections the error can be significant.

This paper describes an integrated method for both finite element model generation and cold-shape correction. Blades are represented precisely as non-uniform rational B-spline (NURBS) surface models (Piegl, 1991). The software implementation of this method incorporates the Initial Graphics Exchange Specification (IGES) (NIST, 1991) format for both input of the hot-shape geometry, and output of the resulting cold-shape NURBS surface model. This rep-

resentation medium facilitates complete integration of the procedure with existing design, analysis, and manufacturing applications.

The paper is presented in two sections. First, a preprocessor is developed that decomposed a hot-shape blade surface model into a mean camber surface and associated thickness functions. A plate finite element mesh is generated on the resulting mean camber surface, i.e., for each mesh node a location and thickness are calculated. The finite element model is used as input for specialized analysis software for cold-shape correction (Ernst, 1992) resulting in the inverse displacements. The second section of the paper describes a postprocessor that reconstructs the corrected, cold-shape blade in two steps. First, the nodal deflections are applied to the hot-shape finite element model to generate a cold-shape mean camber surface. Then the original hot-shape thickness functions are applied to the cold-shape mean camber surface to generate characteristic blade section curves which are lofted to define the cold-shape blade model.

BLADE DESIGN METHODOLOGY AND TERMINOLOGY

The following basic terminology is used throughout the paper. Figure 1 shows the typical cylindrical r-z-θ coordinate system used in blade design. The blade rotates about the z-axis. Generally, for axial turbomachinery components, the radial (r-) axis is referred to as the blade span direction, and the z-axis is referred to as the chord direction. The blade tip is the surface at the maximum r-coordinate, and the root is that of minimum r-coordinate. The leading edge refers to the upstream edge of the blade, and the trailing edge to its downstream edge. Finally, the convex side of the blade is referred to as the suction side, and the concave side as the pressure side.

Blades are typically designed by lofting a number of characteristic blade section curves arranged along a stacking axis which defines the span direction. The blade section curves are generally developed on one of three types of construction spaces: planes perpendicular to the r-axis at monotonically increasing r-values, concentric cylinders about the z-axis, or on concentric surfaces of revolution defined by r-z-curves which characterize desired flow field (Oliver et al., 1994). For the latter two options, transformations exist to map the spaces onto equivalent planar regions. Thus a characteristic blade section curve may be defined in a two-dimensional coordinate system and mapped into a general space curve lying within a surface of revolution. Alternatively, given a surface of revolution containing a blade section (space) curve, an equivalent two-dimensional image can be obtained via the inverse of these mapping functions.

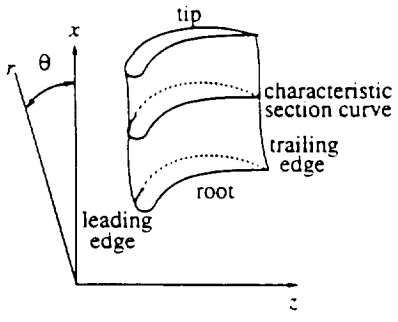

Figure 1 . Blade terminology

A mean camber surface can be considered as a skeletal surface of the blade. It is distinct, however, from those generated from the medial axis transform (MAT) (Nackman and Pizer, 1985). The MAT is defined as the locus of the centers of all maximal inscribable spheres within the boundary of an object. Thus the MAT-derived skeleton is defined relative to boundary surface normals. In contrast, the mean camber surface normals form the basis for its generation. As shown in Figure 2, boundary surface points are defined relative mean camber surface normals at $t/2$ and $-t/2$, where t is the thickness of the blade.

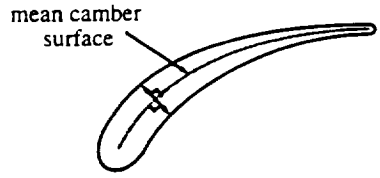

Figure 2 . Mean camber surface definition

HOT SHAPE BLADE FINITE ELEMENT MODELING

The cold-shape finite element analysis requires a model comprised of plate elements. Thus, for each node, the position and corresponding blade thickness must be generated. To facilitate this calculation, the hot-shape blade mean camber surface is calculated. The procedure for mean camber surface construction involves extraction of several characteristic blade profile curves, and application of a two-dimensional algorithm for mean camber curve generation (Patrikalakis and Bardis, 1992). Ideally, the blade should be decomposed using the exact method used to construct it, i.e., planes, cylinders, or surfaces of revolution. However, since only the NURBS surface model of the blade is provided as input, there is generally no way of knowning what method was used to construct the blade. Thus, to generate a representative set of characteristic profile curves, the blade is intersected with concentric cylinders centered about the z-axis. Cylinders were chosen over parallel planes because the concentric cylinder method is more commonly used by blade designers. The number of cylinders (and resulting section profile curves) required to accurately characterize the overall shape of the blade is chosen heuristically as one for each knot in the span parametric direction of the original NURBS blade model.

For a particular blade-cylinder intersection, a general surface-surface intersection algorithm is necessary to extract the intersection curve. Parametric surface intersection is an active research topic in its own right, and a number of techniques have appeared in the literature, see, Patrikalakis and Prakash (1990), Barnhill and Kersey (1990) and Kriezis et al. (1992), for instance. In this application, the surfaces involved are relatively simple, and an intersection algorithm implemented in the DT_NURBS Spline Geometry Library (US Navy, 1993) has proven to be robust and efficient.

For each blade-cylinder intersection curve, the mean camber line must be calculated. The result of the surface-surface intersection is a three-dimensional (space) curve, but the mean camber algorithm

of Patrikalakis and Bardis (1992) requires a closed planar curve as input. Thus an angle preserving transformation is developed to map the space curve onto a plane. The transformation is generated from a two-dimensional curve in the r-z-plane which defines a surface of revolution about the z-axis. The transformation maps an r-z-curve onto a one-dimensional coordinate (m') equivalent to meridional arc length normalized with respect to radius. The justification for and details of this mapping are described in Oliver et al. (1994). It is summarized here for completeness.

Curve arc length is defined by,

$$m_i = \int_0^{u_i} \sqrt{(r^u(u))^2 + (z^u(u))^2}\, du \tag{1}$$

where the superscripts represent differentiation, and m_i represents accumulated curve length evaluated at a number of ordered parametric samples.

These values are interpolated with a cubic NURBS curve so that the resulting arc length function may be normalized with respect to radius,

$$m'_i = \int_0^{u_i} \frac{m^u(u)}{r(u)}\, du \tag{2}$$

the resulting m_i' values are interpolated with a cubic NURBS curve to complete the mapping. The transformation has the effect of "unrolling" the surface of revolution on which the blade profile lies, and "flattening" it in the axial (z-) direction. It allows representation of a space curve defined in cylindrical coordinates in terms of its circumferential ($r\theta$) and axial (m) arc length components. The transformation preserves angles because both length components are normalized with respect to radius, i.e.,

$$\tan\alpha = \frac{d(r\theta)}{dm} = \frac{d\theta}{dm/r} = \frac{d\theta}{dm'} \tag{3}$$

where α is an angle in the m-$r\theta$ or m'-θ coordinate system. In this application a cylinder is represented as a line parallel to the z-axis. After transformation, the mean camber curve algorithm is applied to the m'-θ image of the intersection curve, resulting in a series of points which lie on the mean camber curve. These points are then mapped back into the r-z-θ coordinate system, via the inverse of the above transformation.This technique is applied to each intersection curve.

The root and tip of the blade geometry are characterized as iso-parameter curves on the blade surface and must be handled as special cases. In general, these are space curves which do not lie on a cylinder. The goal is to find a representative r-z-curve defining a surface of revolution which contains the curve. The first step is to intersect the blade root or tip curve with a number of planes of constant z-coordinate. For most axial turbomachinery applications, section curve orientation and convexity are defined such that, excluding the leading and trailing edge, only two intersection points are generated from any particular z-plane intersection. If the blade was constructed according the one of the three methods outlined above, the two intersection points will have identical r-coordinate values. In many cases, however, the values are slightly different. Thus, the following approximate solution is implemented. For each intersection pair, the point with the smallest r-value is retained, if traversing the tip curve, and the one with the largest value is retained if traversing the root curve. This ensures the resulting surface of revolution will completely intersect both sides of the blade.

The r-z coordinates of these points are interpolated with a cubic NURBS curve which is then employed to develop m'-θ transformation functions for both the blade root and tip iso-parameter curves. After transformation, the mean camber curves for each are generated as described above.

Given the necessary mean camber curves in the r-z-θ coordinate system, a mean camber surface is generated via NURBS lofting techniques (Piegl, 1991). The resulting surface serves as the basis for plate mesh generation. An interactive interface allows user specification of the number of nodes desired in both the span and chord directions. The default nodal distribution is at uniform parametric intervals in both parametric directions. Alternatively, several options are provided for biasing nodal distribution toward specific areas of interest. Since the underlying geometry is a NURBS surface, the three-dimensional node location is computed via surface evaluation at each parameter pair; generated as described above. To calculate the blade thickness at each node, a mean camber surface normal is computed and intersected with the original blade surface. Since, by definition, the mean camber surface is equidistant from the original surface, measured along its normal, either the either the suction or pressure side intersection can be used. The thickness is given by doubling the distance between the node and the intersection point. The resulting blade hot-shape plate finite element model is output to a file which is compatible with the cold-shape analysis software.

This mesh generation technique has been implemented with an interactive visual interface using the IRIS Inventor graphical toolkit and the C++ programming language on Silicon Graphics workstations. The technique has been tested successfully on a variety of blade models. The interface allows the user to interactively view the hot-shape blade and the mean camber surface, and to control the variables in generating the finite element mesh.

EXAMPLES

Figures 3, 4, 5, and 6 demonstrate the hot-shape decomposition process and finite element mesh generation procedure for a general turbomachinery blade. First, the nominal hot-shape blade geometry is displayed in wireframe form in Figure 3. Next, the generated blade section curves and their corresponding mean camber curves are shown in Figure 4. Finally, in Figures 5 and 6, the mean camber surface and associated finite element mesh are shown.

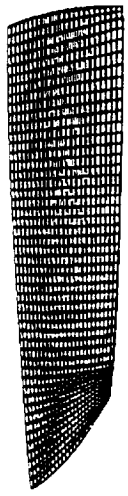

Figure 3 . Hot-shape turbine blade geometry

Figure 5 . Mean camber surface

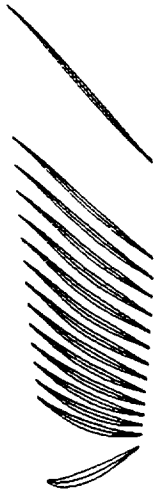

Figure 4 . Blade sections and mean camber curves

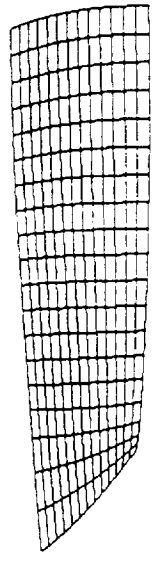

Figure 6 . Mean camber surface finite element mesh

COLD-SHAPE CORRECTION POSTPROCESSOR

The goal of the cold-shape postprocessor is to generate an accurate NURBS surface model of a blade in its nominal configuration, i.e., its cold-shape. The plate element model described above is augmented with characterizations of the blade's steady state pressure and centrifugal loading, and an iterative finite element analysis procedure is utilized to determine nodal displacements due to "unloading" the blade (Thorp and Downey, 1992). In addition to this the displacement data, the postprocessor requires the hot-shape finite element model as input. Although it is not required for the shape correction algorithm, the original hot-shape blade geometry may also be input to provide visual comparison between hot and cold shapes.

The cold-shape correction postprocessor makes extensive use of the mean camber technology developed for the hot-shape preprocessor. The basic assumption underlying the technique is that deflections due to operational loading are characterized primarily by bending and torsion, and that thickness change in the blade is negligible. The procedure is depicted schematically in Figure 7, and the corresponding steps are summarized as follows:

1) displace each mean camber curve
2) find a surface of revolution which contains each one
3) map each to its unique m'-θ coordinate system
4) apply the hot-shape thickness functions to generate section profile curves
5) transform each section profile curve into the cylindrical coordinate system
6) loft the three-dimensional section profile curves.

The hot-shape mean camber surface is developed from a number of intersection curves. Because the mesh distribution is user specified, the resulting chord rows of nodes do not generally correspond to the construction mean camber curves. However, since the mean camber surface is constructed by lofting curves which lie in cylindrical surfaces of revolution, arbitrary iso-parameter curves in the chord-wise direction will also lie on a cylinder (except near the root and tip). The vector displacement output of the cold-shape analysis is applied to each node of a row, and a cubic NURBS curve is interpolated through the points. (Figure 7a). This displaced mean camber curve will generally not lie on any cylinder. Thus, the surface of revolution on which the displaced (cold-shape) mean camber curve lies must be generated. In this case, the $r(u)$ and $z(u)$ components of

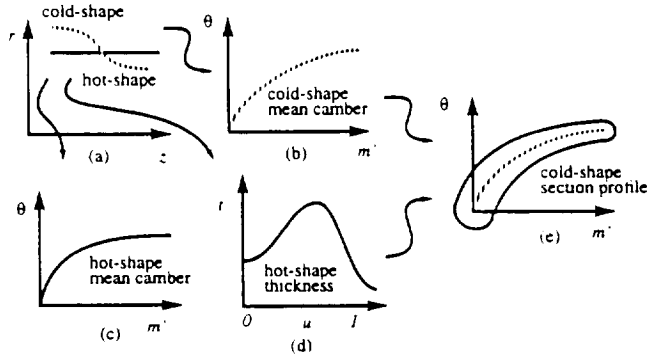

Figure 7 . Cold-shape correction methodology

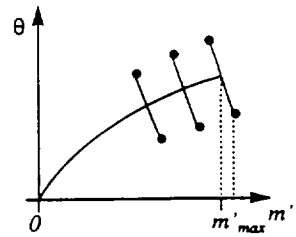

Figure 8 . Cold-shape section profile points exceeding the domain of the m'-θ mapping

the displaced mean camber curves are used to generate the m'-θ mapping for each blade section (Figure 7b).

For each row of nodes, a thickness function corresponding to each mean camber curve is generated based on the thickness value specified at each node. Because thickness change is assumed to be negligible the original cylindrical surface of revolution of the hot-shape blade is used to transform each thickness value into an equivalent normalized space (Figure 7d). The transformed thickness values are interpolated with a cubic NURBS curve using the same parameterization as the corresponding hot-shape mean camber curve (Figure 7c). Thus, for every parameter value $u \in [0, 1]$ a blade thickness can be computed relative to a specific location on the mean camber curve

To create the cold-shape section profile curves, the thickness functions are applied to each cold-shape mean camber curve (Figure 7e). A number of points are generated on each section profile by offsetting cold-shape mean camber curve points by one half of the corresponding thickness in both directions perpendicular to the mean camber curve. Since it is based on a mean camber curve, the domain of the m'-θ mapping may not be sufficient to characterize all the generated section points. As shown in Figure 8, due to the offsetting procedure, some of the section points may have negative m' values, or values larger than the maximum of the mean camber curve. To address this problem, an extrapolation technique is implemented to extend the domain of the m'-θ mapping. After each row of displaced nodes is fit with a NURBS curve in r-z-θ space, two additional points are added on each end based on the maximum thickness of the section and the parametric first derivatives. The curve is then interpolated again with these additional points to extend the mapping domain in each direction.

After the cold-shape section profile points are generated, those on the suction and pressure sides are interpolated with individual cubic NURBS curves. The leading and trailing edges are modeled with rational quadratic Bezier segments, as shown in Figure 9. The interior control point of each quadratic segment is determined by the intersection of the appropriate pressure and suction curve tan-

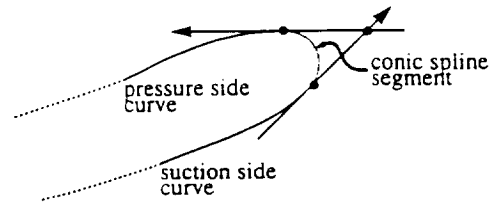

Figure 9 . Leading/trailing edge conic spline

gents, thus ensuring G^1-continuity. The weights corresponding to each control point are assigned to characterize either circular arcs or ellipse segments, as specified by the user. Note that if the tangents intersect such that the required arc span is larger than π, then two conic segments are generated. The degree of the leading and trailing edge segments is then increased to match that of the pressure and section curves, using a NURBS degree elevation algorithm (US Navy, 1993). Finally, the spline segments are joined, and the curve is reparameterized so that the $u = 0$ and $u = 1$ points corresponds to the mid-arc point of the trailing edge segment.

After the characteristic cold shape blade sections are generated in m'-θ space, they are mapped into r-z-θ space by transforming their respective control points. Then the three dimensional blade section curves are lofted in the span-wise direction to produce the final NURBS surface model of the blade in its cold-shape configuration.

EXAMPLES

The cold-shape correction is demonstrated in Figure 10. The figure shows results of the cold-shape correction method on the turbine blade presented in Figures 3-6. For reference, the original hot-shape blade geometry is shown along with the resulting cold-shape geometry. The cold-shape displacements have been multiplied by a scale factor of five so that the shape distortion is obvious.

CONCLUSIONS

The integrated and interactive cold-shape correction pre- and postprocesors provide an effective tool for the design, analysis and manufacture of turbomachinery blades. Given blade hot-shape geometry in the IGES format the technique generates a NURBS based representation of the cold-shape geometry which may be output in the IGES format. The shape correction method is more accurate and efficient than currently available techniques. The method has been successfully applied to several industrial examples.

Future research will improve the capabilities of the integrated shape correction method. The upgrades to the technique will include mid-span shroud capabilities, the ability to accommodate blade loading, and three dimensional mesh generation capabilities. The mid-span shroud capability will permit input of NURBS blade models with external supports, i.e. mid-span shrouds. This task will focus on methods to accommodate mid-span shroud geometry within a mesh. The blade loading ability will be integrated by utilizing DT_NURBS capability for sub-range surfaces which will allow loads, such as temperature and pressure, to be incorporated into the surface representation itself. Finally, the preprocessor will allow the generation of three dimensional solid elements which will accommodate internal voids and other three dimensional shape details.

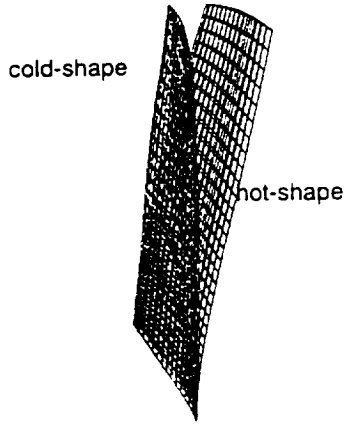

Figure 10 . Cold-shape correction

REFERENCES

Barnhill, R.E. and Kersey, S.N., 1990, "A Marching Method for Parametric Surface/Surface Intersection," *Computer Aided Geometric Design*, 7, pp.257-280

Ernst, M.A., 1992, "Structural Analysis of Low-Speed Composite Propfan Blades for the LRCSW Wind Tunnel Model," *NASA Technical Memorandum* 105266.

Kriezis, G.A., Patrikalakis, N.M. and Wolter, F-E, 1992, "Topological and Differential Equation Methods for Surface Intersections," *Computer-Aided Design*, Vol. 24, No. 1, pp. 41-55

Oliver, J.H., Nair, N.K. and Shanahan, D.E., 1994, "Geometric Design of Turbomachinery Blades on General Stream Surfaces," to appear, *Proceedings of ASME Winter Annual Meeting*, Chicago, IL, November

Nackman, L.R. and Pizer, S.M., 1985, "Three-dimensional Shape Description Using the Symmetrical Axis Transform I: Theory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 2, pp. 187-202

NIST, 1991, "The Initial Graphics Exchange Specification (IGES) Version 5.1," National Institute of Standards and Technology

Patrikalakis, N.M. and Bardis, L., 1992, "Feature Extraction from B-spline Marine Propeller Representations," *Journal of Ship Research*, Vol. 36, No. 3, pp. 233-247

Patrikalakis, N.M. and Prakash, P.V. 1990, "Surface Intersections for Geometric Modeling," *ASME Transactions, Journal of Mechanical Design*, Vol. 112, pp. 100-107

Piegl, L., 1991, "On NURBS: A Survey," IEEE Computer Graphics and Applications," Vol. 11, No. 1, pp. 55-71

Thorp, S.A. and Downey K.M., 1992, "Computer Aided Design and Manufacturing of Composite Propfan Blades for the Cruise Missile Wind Tunnel Model," *NASA Technical Memorandum* 105269

US Navy, 1993, "DT_NURBS Spline Subprogram Library Users' Manual," Naval Surface Warfare Center, David Taylor Model Basin, Bethesda, MD