THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
345 E. 47th St., New York, N.Y. 10017

98-GT-578

SWEEP IN A TRANSONIC FAN ROTOR: PART 1. 3D GEOMETRY PACKAGE

Hazem F. Abdelhamid [1]

Raymond P. Shreeve [2]

Department of Aeronautics & Astronautics
Naval Postgraduate School
Monterey, California 93943-5106

ABSTRACT

A geometry package was developed which uses six Bezier surfaces to describe an axial compressor blade. The blade is defined by 32 control points and two parameters, which determine the leading and trailing edge extensions. The package was used to represent a reference transonic fan rotor to within machining tolerances, and then to introduce forward and backward sweep holding blade-element design parameters fixed. Blade lean and point geometry manipulations were also demonstrated. All geometries produced by the package are machinable without approximation. The Bezier-surface representation was chosen in order to minimize the number of control points required to specify the blade shape and eventually enable aero-structural-manufacturing optimization.

INTRODUCTION

Design systems for axial fan (or compressor) stages generally begin with a 'quasi-3D' description of the flow through blade rows. Axisymmetric stream surfaces are calculated that satisfy the steady equations for continuity, motion and energy, and 2D blade elements are arranged on those surfaces to produce the turning angles required at each radius. Empiricism is introduced to account for a multitude of effects that characterize real fluid behavior. The blade elements themselves are defined by a camber-line geometry, distribution of thickness and choice of leading and trailing edge shapes. Typically, it takes more than 20 parameters to define each blade element and perhaps more than 20 elements to define the complete blade from hub to tip. The 'stacking line' through the center-of-area of each element can be moved axially or circumferentially to create 'sweep' or 'lean'. A complete account of a quasi-3D design procedure is given by Jennions and Stow (1985).

In recent years, computational fluid dynamics (CFD) has been added as a tool in the design process. Transonic stage designs have been reported by Denton (1994) and Sanger (1996) that resulted from the application of flow codes to predict the 3D flow behavior, and the progressive refinement of the geometry to produce improved performance. In both cases, the refinement of the geometry was effected by changing blade elements individually, because, while the flow analysis was fully 3D, the geometry package was the same as was used in the initial quasi-3D description.

As a consequence of using the quasi-3D geometry package in the 3D design-refinement process, there is little hope of developing a meaningful design-optimization procedure because of the large number of parameters required to specify the blade geometry. Also, since each blade element can be changed in shape and moved axially and peripherally without interaction (by either the aerodynamic or structural designer), complex blade shapes can result which are not easily machined.

The incorporation of sweep in the design of an axial fan was the motivation for the present study. It was immediately apparent that, unlike the incorporation of sweep in aircraft wing design (wherein the oncoming flow is uniform along the span), the fan-blade design problem (with relative Mach number varying along the radius) is inherently three dimensional. In particular, the movement of a single blade-element on a stream-surface (derived using quasi-3D modeling) in a transonic rotor design will affect the overall shock structure. Hence a design procedure that incorporates sweep must be three-dimensional, certainly the design optimization procedure must be three-dimensional, and therefore the perturbation in the blade geometry should logically be made three-dimensionally.

Therefore a suitable three-dimensional representation of the geometry of axial transonic fan blading was sought, and Bezier-surface representation was chosen. Having few control parameters, but allowing flexible manipulation, it could well facilitate 3D blade design

[1] Major, Egyptian Air Force, Graduate Student
[2] Professor of Aeronautics & Astronautics

Presented at the International Gas Turbine & Aeroengine Congress & Exhibition
Stockholm, Sweden — June 2–June 5, 1998

optimization. A significant advantage of the chosen representation was that a very compact file of control points provided a complete analytical description of the blade surfaces. This complete information could be transferred, in a standard IGES file format (IGES, 1996), to structural-analysis and machine-control programs, with no approximation.

The Bezier representation was first shown to approximate typical two-dimensional compressor blade elements, and then a current three-dimensional, transonic-rotor blade shape designed by Sanger (1996), to within the required machining tolerances. The representation was found to be very flexible and capable of rational geometry manipulation. It was also determined that the parameters used currently in blade design could easily be represented in terms of the Bezier control parameters, and that 'sweep' and 'lean' could be introduced and controlled very easily.

The present approach departs from the usual practice of representing a blade in terms of stacked blade elements. In recent work by Miller et al (1996, 1997), non-uniform rational B-splines (NURBS) surfaces were derived from NURBS-represented blade elements laid out on axisymmetric stream surfaces. Geometry changes could be made by moving individual control points or by changing individual blade-element parameters. Their approach was adopted purposefully to accommodate current design practices and to provide the designer maximum flexibility in changing the shape of the blade. In contrast, the present approach limits that flexibility by specifically choosing the Bezier representation (a subset of NURBS) and minimizing the number of control points required to specify the shape of the blade. However, since (as is shown here) transonic fan blade shapes can be represented this simply, the Bezier description does offer the potential of enabling the development of a rational (programmed) 3D aero-structural design optimization procedure based on the use of CFD and finite-element method (FEM) analyses. The Bezier surface shapes are immediately machinable. Also, it is noted that the blade-element description is not lost in the present approach. The ability to control the important blade angles is retained and, while it is not explicitly shown here, all parameters which are important to the designer can be calculated and tracked as the geometry is perturbed.

The development of an initial geometry package is reported in the present paper. Subsequently, the effect of sweep on blade row performance was investigated. Sweep was introduced using the Sanger rotor (1996) in a particular manner that sought to keep the parameters of the baseline blading design (maximum thickness, maximum thickness location, section profile, etc.), unchanged. This was in contrast to previously reported sweep attempts that incorporated other changes to the design to meet steady and unsteady stress constraints. The geometries represented by the Bezier format were imported into IDEAS (Lawry, 1997) with no approximation, and purely centrifugal stress analysis was conducted. The results obtained for the effects of sweep on blade row performance and stress levels are given in Part 2 of the present paper (Abdelhamid et al, 1998).

BEZIER REPRESENTATION

A review was made of the different techniques that can be used to represent geometrical shapes (Abdelhamid, 1997; Farin, 1993; Piegl and Tiller, 1987). First, of the various parametric representations available to describe a spline, the cubic Bezier representation had particular advantages for the intended application to blading; namely, that two of the four control points describing the spline are end points, and the other two control points control the slopes at those end points. Thus the leading edge location, trailing edge location, slopes at those points, and shape of a complete camber line of a blade element could, in principle, be controlled by moving only four control points. To describe the camber surface of a complete blade using a Bezier surface description, a total of only sixteen control points are required; four to determine the leading and trailing edge points at the hub and case wall, four pairs of points to determine the two blade edges and contours at the walls, and four to determine the surface shape. However, recognizing that the goal was to work with a geometry representation that could be machined automatically and without further approximation, the camber line and thickness distribution was not the description used here. Rather, the pressure and suction surfaces were separately represented by Bezier surfaces, and separate Bezier surfaces were derived to describe the blade leading and trailing edges. Only those features of the Bezier representation that are important to the blade geometry package, and to a designer who uses it, are reviewed here.

Bezier Curves

The parametric representation of a cubic Bezier spline is

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4 \tag{1}$$

where Q is one of the three coordinates of a point on the curve, the P's are the corresponding coordinates of the four control points, and t is a parameter which varies from 0 to 1 to describe the spline from one end to the other. The variation of one coordinate with another, the geometry of the curve, is obtained by cross plotting.

The coefficients of the P's in Eqn.(1) are the Bernstein polynomials which are, in general, recursive; namely, the higher order polynomial is given in terms of the next lower order polynomial by the relation

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \tag{2}$$

and the Bernstein polynomial is given in the binomial form as

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{3}$$

The Bezier formula is, in general

$$Q(t) = \sum_{i=0}^{n} p_i B_i^n(t) \tag{4}$$

which becomes, in matrix form, for third order,

$$Q(t) = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t^0 \\ t^1 \\ t^2 \\ t^3 \end{bmatrix} \tag{5}$$

The matrix form is useful for computer implementation, especially if matrix multiplication is hard wired.

An example of a third order Bezier curve and the effect of moving one control point 20% of the segment length backwards and forwards along the tangent at the 'trailing edge' are shown in Fig. 1. Notice the curve is moved in the same direction as the control point, generating the broken line from the continuous reference line. In moving the control point, the 'Convex hull' property of the Bezier curve ensures that the curve remains within the polygon described by the control points. This is illustrated in Fig. 2.

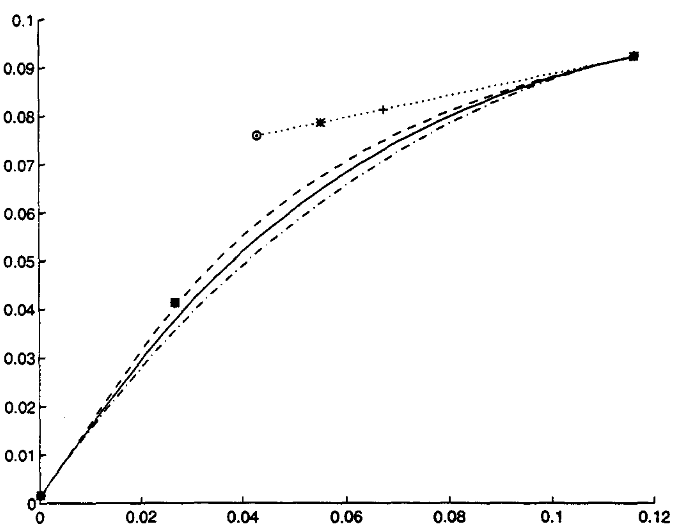

Fig. 1 Control Point Movement In The Tangent Direction

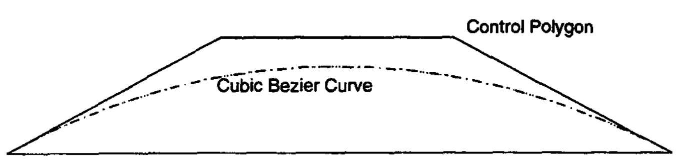

Fig. 2 Convex Hull Property

Bezier curves have other properties which make them particularly attractive for design and graphics. They are invariant under translation, rotation and scaling (but not under perspective projection). For example, the equation

$$Q(t) = \sum_{j=0} p_i n B_i^n(t) + a = \sum_{j=0}^{n} (p_i + a) B_i^n(t) \tag{6}$$

shows that translation of the control points is equivalent to the same translation of the curve. Curves are also invariant under linear combination. Also, examining the Bernstein polynomials, each one is seen to have only one maximum. That maximum is at t=i/n (Farin, 1993), where 'i' is the number of the term in Eqn. 1. and 'n' is the order. So to change the curve at one third of the parameter for example, the movement of the second control point would have the most effect.

If two splines are to be joined, continuity of slope and curvature requires specific relationships between control points. For example, the derivative of Eqn. 1 at t=0 gives the slope as three times the difference in the coordinates of the first two control points, and at t=1, as three times the difference in the last two control points. Hence, if two Bezier curves are to be joined with continuity in slope, the respective control points must be collinear, as shown in Fig. 3. The segment lengths a1 and b1 are not constrained.

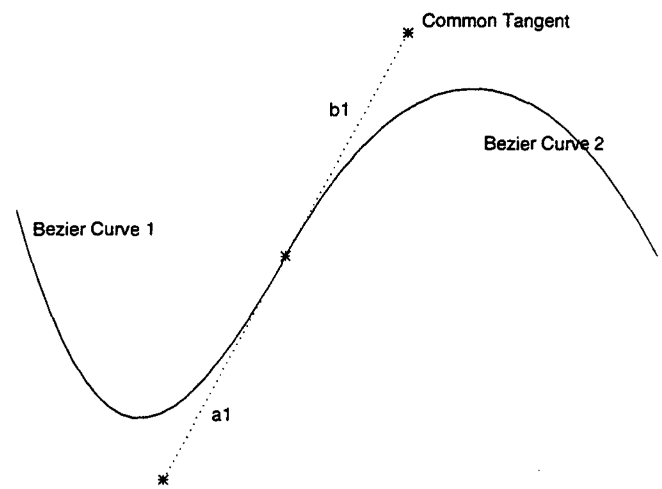

Fig. 3 Two Bezier Curves with First-Derivative Continuity

Continuity of curvature is illustrated in Fig. 4. The point of intersection of the lines joining control points two and three of the two Bezier curves defines the segment lengths b3 and a2. The control points themselves define segment lengths a3 and b2 respectively. Continuity of curvature requires the relationship between segment lengths given in Fig. 4, but the individual lengths a1 and b1 are not constrained. It is noted that continuity of curvature in the parametric representation (C^2 continuity) does not imply the equivalent (G^2 continuity) in the geometry. One type of continuity does not guaranty the other. Geometric continuity is less constraining, and the more important in design applications.

An important property of the Bezier representation, which was needed in the first application of the present work (Abdelhamid, 1997), is subdivison. Given a Bezier curve with a set of control points, it is relatively easy and always possible to find two sets or more of control points. Each set represents a Bezier segment exactly while maintaining C^3 continuity between segments.

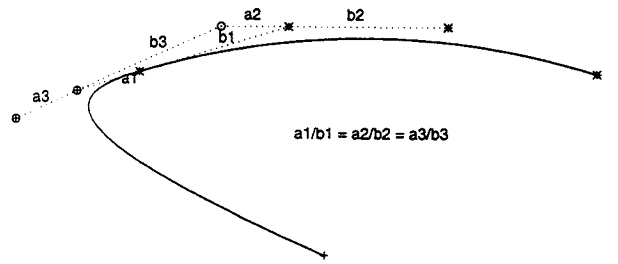

Fig. 4 Condition for Second Derivative Continuity for Bezier Curves

Bezier Surfaces

Bezier surfaces are defined by sixteen control points. The edges by themselves are Bezier curves. The surfaces possess the same properties as the curves. An individual Bezier surface is called a surface patch. Surfaces can be formed by joining surface patches together, with the required degree of continuity. Usually, surface continuity is more difficult to satisfy than continuity for curves. The Bezier surface is defined with two parametric variables u and v. The form of the Bezier patch is

$$Q(u,v)=\sum_{i=0}^{n}\ \sum_{j=0}^{m} p_{ij} N_i^n(u) M_j^m(v) \tag{7}$$

where N and M are Bernstein polynomials. A Bezier surface with its control net is shown in Fig. 5.

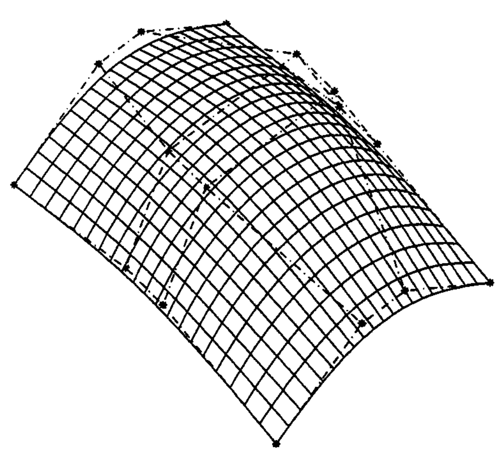

Fig. 5 Bezier Surface with its Control Net

The effect of moving one (the lower left) of the four inner control points for the surface shown in Fig. 5 is illustrated in Fig. 6. Only one coordinate of the one control point was changed in this example. The conditions for surface continuity are the same as for the curves, with the extra condition that the four tangents along the connecting edge shown in Fig. 7 must be such that a1/b1=a2/b2=a3/b3=a4/b4.

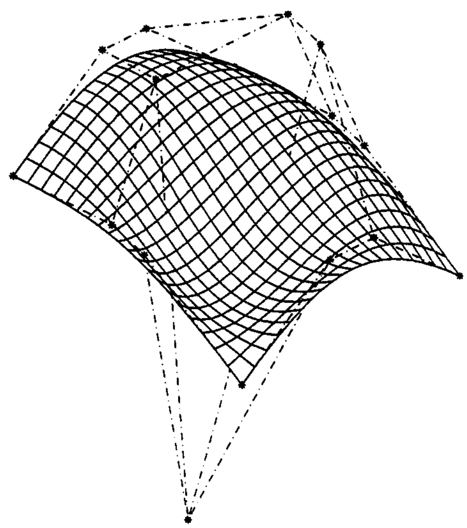

Fig. 6 Effect of Moving One Bezier Surface Control Point

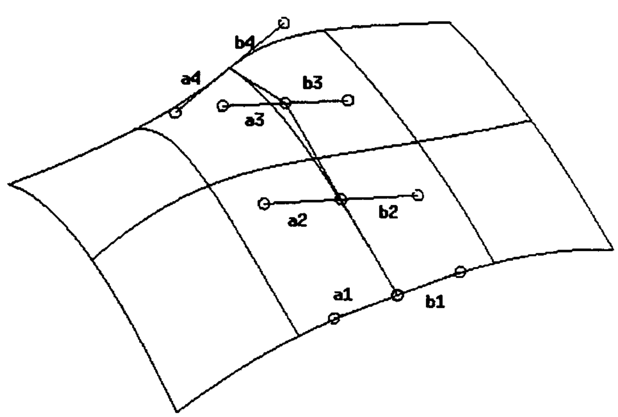

Fig. 7 Tangency Condition for Bezier Surfaces

Surface Fitting

Whether the Bezier representation is suitable for axial blading depends on the ability to adequately represent typical blade element profiles and, if this is successful, typical blade surfaces. The problem is then, given a set of data points, what are the control points that give the best parameterization for the given curve or surface. Hoscheck (1988) solved this problem by splitting it into two least-squares problems. The first is the least-squares problem of finding the control points for the surface that best fits the given geometry data points assuming a given parameterization. The second is a nonlinear least-squares problem that solves for the best parameterization given the data points and a set of control points. The procedure described by Hoschek (1988) was programmed and incorporated in the initial geometry package.

INITIAL GEOMETRY PACKAGE

Overview

The initial geometry package was developed incrementally toward the goal of designing with sweep. As a first step, the suitability of the Bezier representation for describing typical 2D compressor blade section profiles was examined. It was found that Bezier curves were suitable and the section shape was well represented. Second, a complete compressor blade was treated as two separate surfaces (pressure and suction), and the representation was again found to be suitable. Treating the pressure and suction surfaces as Bezier patches, leading and trailing (Bezier) surfaces were added. This resulted in a complete blade represented in a form that was compact (32 control points and 2 parameter values), easy to manipulate, and easy to import into different programs without approximation. Most importantly, the coordinates at any position on the blade surfaces could be retrieved with no loss in accuracy (other than that due to machine precision), because the complete shape was analytically described.

The package was developed with current compressor design systems in mind. The initial package can read an existing geometry specified in MERIDL2, MERIDL3 and general format (Chima, 1992). The package can also read the control points that represent the six surfaces, or only the ones that represent the pressure and suction surfaces and then add leading and trailing-edge surfaces. To facilitate further development and the addition of a graphical user interface, the package was implemented using object-oriented programming.

The blade surfaces are numbered from 1 to 6 as shown in Fig. 8.

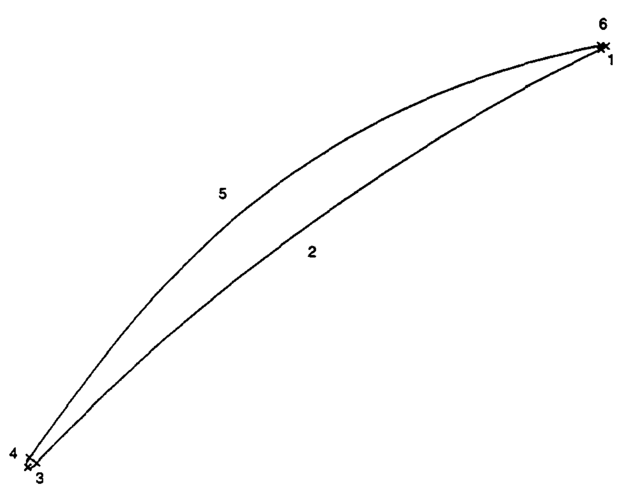

Fig. 8 Blade Surface Numbering

The control points for each surface are numbered from 0 to 15 as shown in Fig. 9. Where surfaces are joined, the surface edge with the lower numbered control points is always on the left (when moving clockwise around the blade). For example, surface No. 2 (pressure side) control points 3, 7, 11 and 15 will match (are the same as) control points 0, 4, 8 and 12 of surface No. 3.

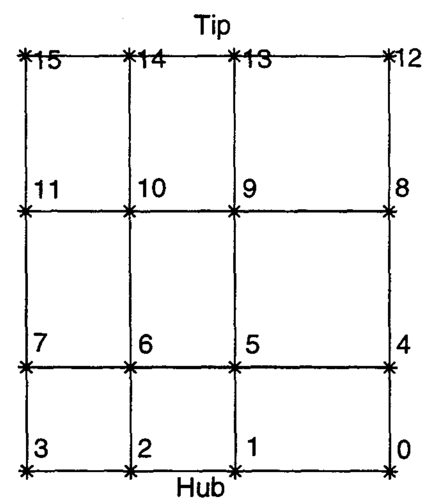

Fig. 9 Control Point Numbering

The blade geometry is manipulated by moving control points individually or in groups that are defined in the program. The groups are defined such that one row of control points for the six surfaces can be moved simultaneously. The movements allowed give the freedom to shape the geometry with great flexibility. The use of the present representation for axial turbomachinery is new and the present work is considered to be far from complete as an exploration of the potential of the method. The intent was to find a rational approach to design with sweep. Hence particular geometry manipulation routines and prescribed parameter variations were implemented. Specifically, two blade sweep schedules, blade lean, and individual control point manipulations were programmed to be computed relative to a reference geometry.

Approximating Blade Geometry

The blade geometry is read in as r, z, θ_1, θ_2 as defined by either MERIDL3 or MERIDL2. MERIDL3 reads in the blade section with θ_1 being the mean camber line tangential coordinate and θ_2 the tangential thickness angle. MERIDL2 reads in the pressure surface and suction surface tangential coordinates as θ_1 and θ_2 respectively. In either case, the coordinates are transformed to x, y, z coordinates, and the surfaces are treated as two separate surfaces by the program. The surfaces are assumed to start and end at their defined leading and trailing edge tangency points respectively. Each surface is fitted by a Bezier surface separately, using the method presented by Hoschek (1988). The fitting routine starts with initial parameterization that uses the cubic root of the arc length. The linear least-squares fitting uses QR factorization, while the parameter correction uses the method given by Hoschek (1988).

Since Bezier surface corner points are control points, the values for the corner points are substituted into the equations to fix the corners. The number of equations then is reduced by four. The problem becomes a classical, least-squares problem in 12 unknowns. The nonlinear part is solved by calculating the error vector at each point and calculating its inner product with the surface tangent with respect to both u, and v parameters [Eq. (7)]. The result is scaled with respect to the corresponding arc length and added to the parameter value at each corresponding point.

Two-dimensional blade sections were tested first. Two test cases which are representative of profiles used in transonic fans are shown in Fig. 10 and Fig. 11. In these figures, the circles are the blade coordinates and the lines are the fitted Bezier curves. Then, using surface fitting, the pressure and suction surfaces of the baseline transonic rotor blade used in the present study (Sanger, 1996) were found to be represented to well-within the tolerances called for in manufacture (0.1 mm for a rotor diameter of 27.94 cm). The remaining problem was to represent the leading and trailing edges using Bezier patches.

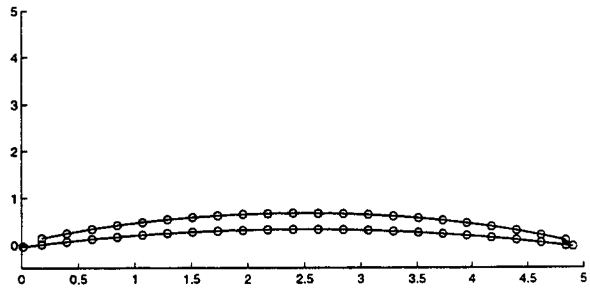

Fig. 10 Double Circular Arc Blade Section [Bezier Approximation]

Adding Leading and Trailing Edge Surfaces

In defining the leading and trailing surfaces there were two considerations. First, the connecting edges had to maintain the appropriate continuity requirement. Second, a certain amount of flexibility needed to be provided so that the leading edge geometry could be changed. Joining the two surfaces using one surface, with even the least amount of continuity, would not satisfy these requirements. However, the use of two surface patches (at both the leading and trailing edge), was found to give continuity of curvature at the pressure and suction sides, continuity of slope at the edge connecting the two added surfaces, and one free parameter which controlled the size of the extension (or bluntness). Thus the added surfaces were defined by the requirement to have a constant ratio of tangent segments along the surface edges [Fig. 7], the specification of one segment length, and the requirement for collinear control points along the common edge. For the transonic-fan rotor blade, since the blade surface arc length was larger at the tip than at the hub, the tip extension was larger than the hub extension. The chord length was adjusted back to its required length everywhere by using subdivision to adjust the fraction of the chord attributable to the basic blade. After adding the leading and trailing edge surfaces the blade was fully defined, with the six surfaces numbered from 1 to 6 as shown in Fig. 8 (clockwise starting from the edge connecting the two trailing edge surfaces).

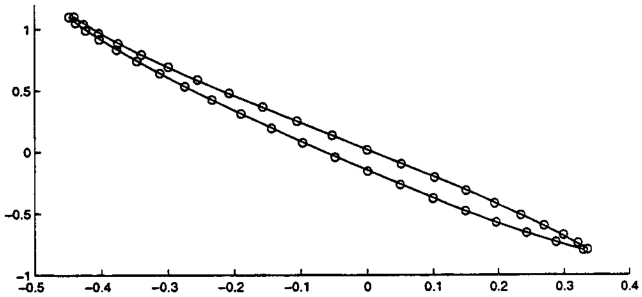

Fig. 11 Transonic Fan Blade Section [Bezier Approximation]

Examples of Geometry Manipulation

The manipulations implemented in the present package are of two types. One is a general point manipulation in which each individual control point can be moved in any of the three coordinate directions (r,z,θ). Second is a programmed manipulation in which a defined set of control points is moved in any of the three directions (r,z,θ). This second type includes a manipulation in which the blade is swept along the blade chord. This is applied either to the top row of control points (sweep_1) or to the top two rows (sweep_2).

General (Point) Manipulation. The general manipulation of the control points allows the designer to move the individual control points in any of the three coordinate directions (r,z,θ). Each point can be moved in one direction at a time. The axial direction (z) and radial direction (r) are in the same units as are used for the data read by the program. The tangential coordinate is in radians. Figure 12 shows the effect on the blade geometry of moving one (surface edge) control point in the tangential direction.

Programmed (Shape) Manipulation. The programmed (shape) manipulations, as currently implemented, move all the control points for the six surfaces that share the same row number, simultaneously. The tangential movement of the top row only, or the top two rows together, gives different types of blade lean. Figures 13 and 14 show examples of the effect of tangential perturbations.

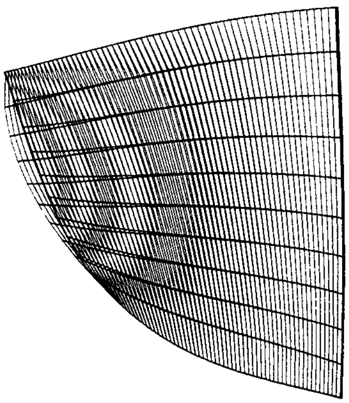

Fig. 12 The Effect of Moving Control Point 11 in the Tangential Direction .05 Radians

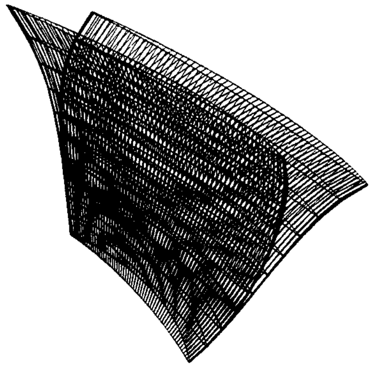

Fig. 13 Blade Lean Created by Moving the Top Row of Control Points 0.05 Radians in Each Direction

Two special manipulations were implemented for sweep. Both sweep the blade tip along the chord direction either forward or backward. The first (sweep_1) moves only the top row of control points a prescribed percentage of the rotor chord in the required direction (see Fig. 15). The second (sweep_2) moves the top two rows of control points simultaneously, the same fraction of chord, in the same way as in sweep_1 (see Fig. 16). Moving one row of control points will affect the whole blade span, but with gradually decreasing effect the further the distance from the row being moved.

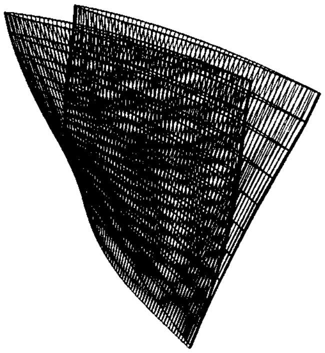

Fig. 14 Blade Lean Created by Moving the Top Two Rows of Control Points 0.05 Radians in Each Direction

The chord slide or 'sweep in the chord direction' implemented in sweep_1 and sweep_2 is achieved by a composed (z,θ) manipulation. The movement of the tip section is shown in Fig. 17. The axial displacement required is calculated as a percentage of the axial chord. That percentage is the parameter passed to the function. The tangential displacement is calculated from the rate of change of the tangential coordinate along the chord. Sweep_2 preserves the original blade slope at the wall, whereas sweep_1 does not.

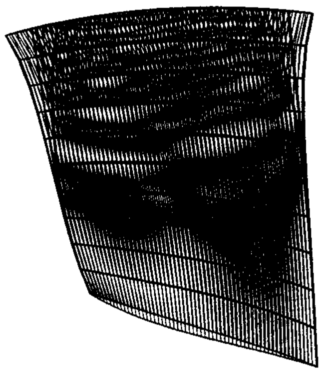

Fig. 15 Forward and Aft Sweep (10% of Chord) Using Sweep_1

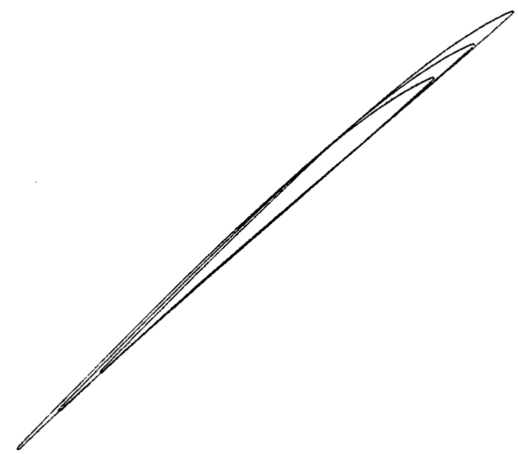

Fig. 17 Movement of the Tip Section with Forward and Aft-Sweep of 10% of Chord

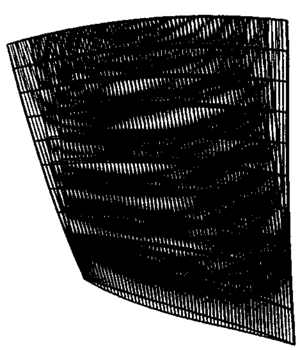

Fig. 16 Forward and Aft Sweep (10% of Chord) Using Sweep_2

<u>Program Language and Structure</u>

A listing of the initial version of the code is given in Abdelhamid and Shreeve (1997). The package was implemented using C++ object-oriented programming. The program consists of six classes; namely, point, vector, matrix, surface, bsurface and blade. Some of the classes are used as tools by the main classes. The class is considered an entity with defined functions. The user is allowed only access to class members through the class interface (class member functions). This facilitates the development because once the class is developed it is not affected by any following development in the program. The application that uses the class works through the interface and is not affected by the internal implementation. For example, one of the member functions in the blade class is the "read_surf" function that reads the pressure and suction surfaces, transforms the data to (x,y,z) coordinates, fits the two Bezier surfaces, and then adds the leading and trailing edge surfaces. A graphical user interface must be added for the benefit of the designer.

APPLICATION TO SWEEP

The geometry package was used to generate and analyze the flow through a baseline rotor and swept geometries using CFD. The baseline geometry used was the transonic rotor designed by Sanger (1996). The Sanger rotor coordinates were read by the geometry package in MERIDL3 format and fitted with Bezier surfaces. The control points obtained for the Bezier representation are given in Table 1.

Sweep perturbations were implemented using the control points in Table 1 and the package predefined routines. After creating a geometry, the blade coordinates were calculated on conical sections for the grid generator, TCGRID, supplied by the NASA Lewis Research Center (Chima 1990). Geometry perturbations included swept forward and swept backward, with different values of sweep using sweep_1 and sweep_2. The grid obtained for the baseline rotor is shown in Fig. 18 and an example of the grid for one swept geometry is shown in Fig. 19.

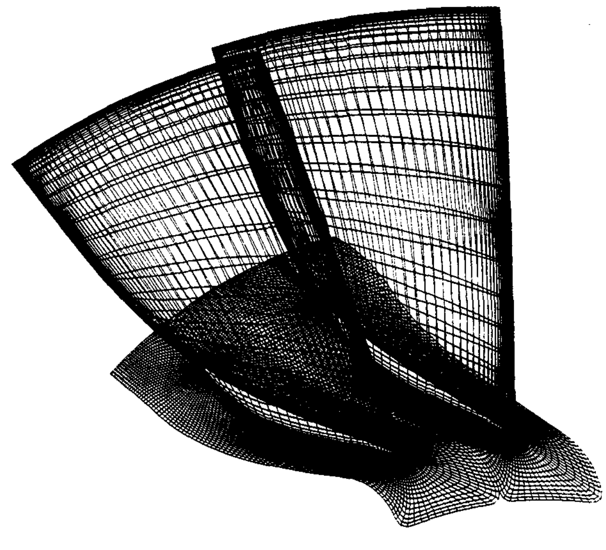

Fig. 18 Grid for Fitted (Base) Rotor Geometry

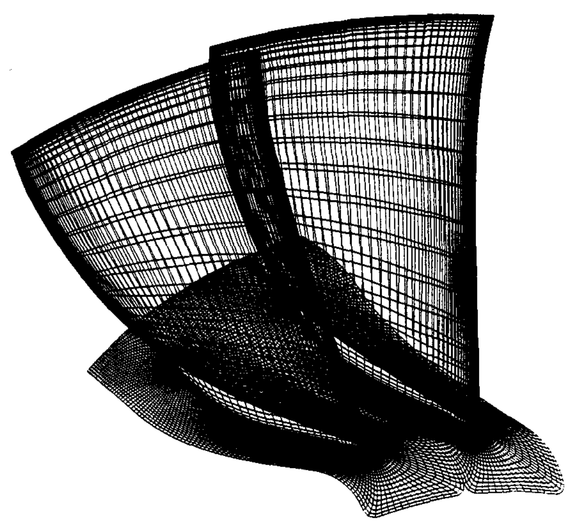

Fig. 19 Grid for a Swept-Forward Rotor Geometry (10% Using Sweep_1)

CONCLUSIONS

A geometry package has been developed which uses six Bezier surfaces to describe an axial compressor blade. The blade is defined by 32 (independent) control points and two parameters, which define the extent of the extensions of the leading and trailing edges. The package can be used to derive a best fit to an existing blade, and programmed routines then allow the geometry to be perturbed. The representation was shown to fit a current transonic rotor geometry to within machining tolerances and forward and backward sweep variations were readily introduced. Since the geometry is defined fully by Bezier surfaces, baseline or perturbed geometries are machinable without further approximation.

ACKNOWLEDGEMENTS

The present study was part of the compressor program sponsored at the NPS Turbopropulsion Laboratory by the Propulsion and Power Department of the Naval Air Warfare Center, Patuxent River, MD, and was carried out while the first author was an international student in the Ph.D. program. The sponsorship of the Egyptian Air Force and support of the International Programs Office are acknowledged.

REFERENCES

Abdelhamid, H.F., "Incorporation of Sweep in a Transonic Fan Design Using a 3D Blade-Row Geometry Package Intended for Aero-Structural-Manufacturing Optimization", Ph.D. Thesis, Naval Postgraduate School, Monterey, California, September 1997.

Abdelhamid, H. F. and Shreeve, R. P., "Three Dimensional Geometry Package for Axial Compressor Blading: Initial Code Listing", Technical Report NPS-AA-97-001, Naval Postgraduate School, Monterey California, September 1997.

Abdelhamid, H.F., Shreeve, R.P. and Hobson, G.V., "Sweep in a Transonic Fan Rotor: Part 2. Flow and Stress Analyses", ASME Paper, International Gas Turbine and Aeroengine Conference and Exhibit, Stockholm, Sweden, June 1998.

Chima, R. V., "TDGRID User's Manual", NASA Lewis Research Center, Cleveland, Ohio, November 1990.

Chima, R. V., "RVC3D - Rotor Viscous Code 3-D", User's Manual and Documentation Version 920318, NASA Lewis Research Center, Cleveland, Ohio, March 1992.

Denton, J. D., "Designing in Three Dimensions", AGARD Lecture Series 195, pp. 3.1-3.14, May 1994.

Farin, G., "Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide", Academic Press, 1993.

Hoschek, J., "Intrinsic Parameterization for Approximation", Computer Aided Geometric Design, pp.27-31,1988

"Initial Graphics Exchange Specification IGES 5.3",U.S.Product Data Association, N.Charleston, South Carolina, 1996

Jennions, I. K. and Stow, P., "A Quasi-Three-Dimensional Turbomachinery Blade Design System: Part I. Throughflow Analysis", Journal of Engineering for Gas Turbines and Power",Vol.107, pp.301-307,1985.

Jennions, I. K. and Stow, P., "A Quasi-Three-Dimensional Turbomachinery Blade Design System: Part II. Computerized System, Journal of Engineering for Gas Turbine and Power, Vol.107, pp.308-316, 1985.

Lawry, M.H., "IDEAS Master Series, Mechanical CAE/CAD/CAM Software", Structural Dynamics Research Corporation, Milford, Ohio, 1997,

Miller, P.L., Oliver, J.H., Miller, D.P. and Tweedt, D.L., "BladeCAD: An Interactive Geometric Design Tool for Turbomachinery Blades", ASME Paper No.96-GT-58, ASME International Gas Turbine and Aeroengine Congress, Exposition and User's Symposium, Birmingham, UK, June 1996.

Miller, P.L., Oliver, J.H., Miller, D.P. and Tweedt, D.L., "Using Stream Surfaces for Blade Design", Mechanical Engineering", April 1997.

Piegl, L. and Tiller, W., "Curve and Surface Constructions Using Rational B-Splines", Computer Aided Design, Vol.19, No.9, November 1987.

Sanger, N. L., " Design of A Low Aspect Ratio Transonic Compressor Stage Using CFD Techniques", Journal of Turbomachinery, Vol.118, pp.479-491, July 1996.

Table 1. Bezier Representation of the Sanger Rotor.

Surface 1			Surface 2		
0.1170	0.0922	0.2700	0.1163	0.0915	0.2699
0.1169	0.0918	0.2701	0.0739	0.0693	0.2537
0.1166	0.0917	0.2700	0.0315	0.0338	0.2398
0.1163	0.0915	0.2699	0.0014	0.0003	0.2236
0.1073	0.1261	0.3139	0.1075	0.1240	0.3128
0.1079	0.1248	0.3129	0.0809	0.0737	0.3050
0.1077	0.1244	0.3129	0.0214	0.0300	0.2455
0.1075	0.1240	0.3128	0.0055	0.0057	0.3801
0.0891	0.1477	0.3888	0.0898	0.1453	0.3831
0.0901	0.1461	0.3834	0.0694	0.0963	0.3591
0.0900	0.1457	0.3832	0.0463	0.0668	0.4362
0.0898	0.1453	0.3831	0.0155	0.0067	0.4038
0.0886	0.1593	0.4274	0.0885	0.1583	0.4278
0.0889	0.1591	0.4275	0.0641	0.1063	0.4468
0.0887	0.1587	0.4276	0.0369	0.0574	0.4544
0.0885	0.1583	0.4278	0.0157	0.0114	0.4555

Surface 3			Surface 4		
0.0014	0.0003	0.2236	0.0002	0.0002	0.2230
0.0011	3.7E-5	0.2235	-0.0003	0.0008	0.2226
0.0008	-0.0003	0.2233	-0.0001	0.0012	0.2228
0.0002	0.0002	0.2230	0.0001	0.0016	0.2229
0.0055	0.0057	0.3801	0.0043	0.0046	0.3689
0.0053	0.0055	0.3815	0.0034	0.0039	0.3549
0.0051	0.0052	0.3828	0.0036	0.0042	0.3534
0.0043	0.0046	0.3689	0.0038	0.0046	0.3520
0.0155	0.0067	0.4038	0.0144	0.0058	0.4062
0.0152	0.0061	0.4035	0.0140	0.0062	0.4093
0.0148	0.0055	0.4032	0.0143	0.0068	0.4099
0.0144	0.0058	0.4062	0.0146	0.0075	0.4105
0.0157	0.0114	0.4555	0.0149	0.0106	0.4555
0.0155	0.0110	0.4555	0.0146	0.0107	0.4555
0.0152	0.0105	0.4555	0.0149	0.0114	0.4555
0.0149	0.0106	0.4555	0.0152	0.0120	0.4555

Surface 5			Surface 6		
0.0001	0.0016	0.2229	0.1162	0.0923	0.2695
0.0269	0.0413	0.2371	0.1166	0.0924	0.2697
0.0551	0.0786	0.2398	0.1171	0.0925	0.2700
0.1162	0.0923	0.2695	0.1170	0.0922	0.2700
0.0038	0.0046	0.3520	0.1061	0.1266	0.3150
0.0228	0.0363	0.2075	0.1064	0.1270	0.3150
0.0612	0.0783	0.3170	0.1068	0.1273	0.3150
0.1061	0.1266	0.3150	0.1073	0.1261	0.3139
0.0146	0.0075	0.4105	0.0877	0.1486	0.3935
0.0441	0.0688	0.4690	0.0879	0.1490	0.3939
0.0555	0.0949	0.3450	0.0882	0.1494	0.3942
0.0877	0.1486	0.3935	0.0891	0.1477	0.3888
0.0152	0.0120	0.4555	0.0879	0.1592	0.4274
0.0435	0.0787	0.4544	0.0881	0.1594	0.4274
0.0676	0.1373	0.4358	0.0882	0.1596	0.4273
0.0879	0.1592	0.4274	0.0886	0.1593	0.4274