12/6-12-92 JS ①

May 1992 • NREL/TP-257-4099

A Surface Definition Code for Turbine Blade Surfaces

S. L. Yang
*Michigan Technological University
Houghton, Michigan*

D. Oryang
M. J. Ho
*Tuskegee University
Tuskegee, Alabama*

Received by OSTI
JUN 11 1992

National Renewable Energy Laboratory
(formerly the Solar Energy Research Institute)
1617 Cole Boulevard
Golden, Colorado 80401-3393
A Division of Midwest Research Institute
Operated for the U.S. Department of Energy
under Contract No. DE-AC02-83CH10093

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

A Surface Definition Code for Turbine Blade Surfaces

NREL/TP--257-4099

DE92 010574

S. L. Yang
*Michigan Technological University
Houghton, Michigan*

D. Oryang
M. J. Ho
*Tuskegee University
Tuskegee, Alabama*

NREL technical monitor: P. K. C. Tu

National Renewable Energy Laboratory
(formerly the Solar Energy Research Institute)
1617 Cole Boulevard
Golden, Colorado 80401-3393
A Division of Midwest Research Institute
Operated for the U.S. Department of Energy
under Contract No. DE-AC02-83CH10093

Prepared under Subcontract No. XX-7-06092-2

May 1992

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

On September 16, 1991, the Solar Energy Research Institute was designated a national laboratory, and its name was changed to the National Renewable Energy Laboratory.

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Printed in the United States of America
Available from:
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Microfiche A01
Printed Copy A10

Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issue of the following publications which are generally available in most libraries: *Energy Research Abstracts (ERA)*; *Government Reports Announcements and Index (GRA and I)*; *Scientific and Technical Abstract Reports (STAR)*; and publication NTIS-PR-360 available from NTIS at the above address.

ACKNOWLEDGMENT

This research is supported by the National Renewable Energy Laboratory (NREL, formerly Solar Energy Research Institute) under subcontract No. XX-7-06092-2 and has been monitored by NREL Project Manager Jim L. Tangler and Dr. Peter K. C. Tu. The authors would like to express their sincere gratitude to George N. Scott of NREL for many helpful discussions and suggestions.

ABSTRACT

A SURFACE GENERATION CODE FOR TURBINE BLADE SURFACES

A numerical interpolation scheme has been developed for generating the three-dimensional geometry of wind turbine blades. The numerical scheme consists of (1) creating the frame of the blade through the input of two or more airfoils at some specific spanwise stations and then scaling and twisting them according to the prescribed distributions of chord, thickness, and twist along the span of the blade; (2) transforming the physical coordinates of the blade frame into a computational domain that complies with the interpolation requirements; and finally (3) applying the bi-tension spline interpolation method, in the computational domain, to determine the coordinates of any point on the blade surface.

Detailed descriptions of the overall approach to and philosophy of the code development are given along with the operation of the code.

To show the usefulness of the bi-tension spline interpolation code developed, two examples are given, namely CARTER and MICON blade surface generation. Numerical results are presented in both graphic and tabular data forms. The solutions obtained in this work show that the computer code developed can be a powerful tool for generating the surface coordinates for any three-dimensional blade.

TABLE OF CONTENTS

	page
LIST OF FIGURES	vi
LIST OF TABLES	x
NOMENCLATURE	xi
 CHAPTER	
1. INTRODUCTION	1
2. TENSION SPLINE INTERPOLATION	5
2.1 Tension Spline for Curves	6
2.2 Tension Spline for Surfaces	11
3. CREATING THE BLADE FRAME	16
3.1 Choice of Span Stations of the Blade at which to Place Airfoils	16
3.2 Choice of Airfoils	17
3.3 Scaling the Airfoils	18
3.3.1 Scaling the Airfoils to the Required Maximum Thickness-to-Chord Ratio	18
3.3.2 Scaling the Airfoils to the Required Chord Length	18
3.4 Twisting the Airfoils	29
4. TRANSFORMATION OF THE BLADE FRAME	24
4.1 Calculation of the Normalized Z-Coordinate	25
4.2 Computation and Normalization of the Polygonal Arc length for Each Input Airfoil	25
4.3 Interpolation of New X and Y Coordinates for Each Airfoil	26
4.4 Defining and Using the Computational Domain Coordinates	27

TABLE OF CONTENTS (Continued)

CHAPTER	page
5. PROGRAM DESCRIPTION	32
5.1 Program Structure	32
5.1.1 Input Module	33
5.1.2 Blade Frame Creation Module	34
5.1.3 Transformation Module	34
5.1.4 Partial Derivatives Module	35
5.1.5 Interpolation Module	36
5.1.6 Output Module	37
5.2 Subroutine Functions	37
5.3 Critical Variable Definitions	41
6. HOW TO RUN THE CODE	45
6.1 Program Installation	45
6.2 Input Data Files	47
6.2.1 Airfoil Data File	47
6.2.2 Chord/Twist/Thickness Distribution Data File	50
6.3 Running the Program	52
6.3.1 Input Options	52
6.3.2 Output Options	58
6.4 Sample Run of the Program	60

TABLE OF CONTENTS (Concluded)

	page
CHAPTER	
7. RESULTS AND DISCUSSIONS	72
7.1 CARTER Blade	73
7.2 MICON Blade	93
8. CONCLUSIONS	112
REFERENCES	114
APPENDIX	
A PROGRAM LISTING	A-1
B DERIVATION OF EQUATIONS (2.2) and (2.5)	B-1
C DERIVATION OF EQUATION (2.11)	C-1

LIST OF FIGURES

Figure		Page
1	Expanded Thin-Airfoil Family	2
2	Data Points in the (x,y) Plane Representing a Curve on Which a 1-D Tension Spline is to be Applied	7
3	Chord-Normalized Input Airfoils at Respective Span Stations	21
4	Input Airfoils That Have Been Scaled and Offset, and Then Placed at Their Respective Span Stations	22
5	Input Airfoils That Have Been Scaled, Offset, and Twisted, and Then Placed at Their Respective Span Stations	23
6	Variation in Grid Spacing with Three Different Stretching Parameters for An Airfoil Represented by 65 Points	28
7	Computational Domain with X as the Dependent Variable	29
8	Computational Domain with Y as the Dependent Variable	31
9	Order of Access of Modules in the BLADE Code	32
10	Structure of the Input Module	33
11	Structure of the Transformation Module	34
12	Structure of the Partial Derivatives Module	35
13	Structure of the Interpolation Module	36
14	Structure of the Output Module	38
15	Sample Input Airfoil Data File (Example of S805A7.DAT)	48
16	Sample Chord/Twist/Thickness Distribution Data File (Example of CTE2.DAT)	51

LIST OF FIGURES (Continued)

Figure		Page
17	Sample Input Data File (Example of JAYIN.DAT)	54
18	Data Used to Make the Sample Run	60
19	Concentric Plot of the Input Airfoils that Form the CARTER Blade Frame	76
20	Concentric Plot of the Airfoils Forming the Generated CARTER Blade	77
21	Perspective View of the Generated CARTER Blade	78
22	Top View of the Generated CARTER Blade	79
23	Front View of the Generated CARTER Blade	80
24	Twist Distribution Along the Span of the CARTER Blade	81
25	Chord Distribution Along the Span of the CARTER Blade	83
26	Maximum Thickness Distribution Along the Span of the CARTER Blade	84
27	Span-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated CARTER Blade	85
28	Span-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade	86
29	Span-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated CARTER Blade	87
30	Span-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade	88
31	Chord-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated CARTER Blade	89
32	Chord-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade	90

LIST OF FIGURES (Continued)

Figure	Page
33 Chord-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated CARTER Blade	91
34 Chord-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade, with the Twist and Offset Removed	92
35 Concentric Plot of the Input Airfoils that Form the MICON Blade Frame	96
36 Concentric Plot of the Airfoils Forming the Generated MICON Blade	97
37 Perspective View of the Generated MICON Blade	98
38 Top View of the Generated MICON Blade	99
39 Front View of the Generated MICON Blade	100
40 Twist Distribution Along the Span of the MICON Blade	101
41 Chord Distribution Along the Span of the MICON Blade	102
42 Maximum Thickness Distribution Along the Span of the MICON Blade	103
43 Span-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated MICON Blade	104
44 Span-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade	105
45 Span-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated MICON Blade	106
46 Span-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade	107
47 Chord-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated MICON Blade	108

LIST OF FIGURES (Concluded)

Figure		Page
48	Chord-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade	109
49	Chord-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated MICON Blade	110
50	Chord-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade, with the Twist and Offset Removed	111

LIST OF TABLES

Table		Page
1	Input Data to Form the CARTER Blade Frame	74
2	Input Chord/Twist/Thickness Data for the CARTER Blade	75
3	Input Data to Form the MICON Blade Frame	94
4	Input Chord/Twist/Thickness Data for the MICON Blade	95

NOMENCLATURE

$A_{i,k}$	vector containing the four 1-D spline coefficients defined by equation (2.5)
$A_{i,j,k,l}$	matrix containing the sixteen 2-D spline coefficients defined by equation (2.19)
B	vector appearing in equation (2.14)
C	matrix defined in equations (2.6) and (2.20)
CL	chord length of an airfoil
CM	coefficients matrix appearing in equation (2.14)
$f_{i,j}$	surface tension spline function defined by equation (2.18)
g_i	curve tension spline function defined by equation (2.2)
K_i	degrees of freedom characteristic vector for a linear element of the 1-D computational domain, as defined by equation (2.7)
$K_{i,j}$	degrees of freedom characteristic matrix for a rectangular element of the 2-D computational domain, as defined by equation (2.21)
m	total number of airfoils representing the blade frame
n	total number of points representing the airfoil surface
OS_x, OS_y	airfoil twist center
P	vector appearing in equation (2.14)
$p_{i,j}$	partial derivative defined in equation (2.24)
$q_{i,j}$	partial derivative defined in equation (2.24)
$r_{i,j}$	partial derivative defined in equation (2.24)
s_j	term defined in equation (2.36)
S_i	polygonal arc length as defined in equation (4.3)

NOMENCLATURE (Continued)

$\bar{S}_i$	normalized polygonal arc length as defined in equation (4.4)
t_i	term defined in equation (2.35)
u_{ij}	term defined in equation (2.24)
v_i	term defined in equations (2.8) and (2.22)
w_j	term defined in equation (2.23)
X_i, x_i	physical domain X-coordinate
Y_i, y_i	physical domain Y-coordinate
Z_j, z_j	physical domain Z-coordinate
Greek Symbols	
α_i	tension parameter in the ξ direction
β_j	tension parameter in the η direction
η_j	computational domain coordinate defined by equation (4.1)
τ	stretching parameter
Θ_j	twist angle of airfoil
ξ_i	stretching function and computational domain coordinate defined by equation (4.8)
$\bar{\xi}_i$	equally spaced polygonal arc lengths, defined by equation (4.7)
Φ_k, Φ_l	interpolation functions in the ξ – and η – directions respectively, as represented by equation (2.3)

NOMENCLATURE (Concluded)

Subscripts

- i, j denote the grid indices in (1) the ξ – and η – directions respectively for the computational domain; and (2) the chordwise- and spanwise- directions respectively for the physical domain
- k, l denote the position placement of coefficients in the interpolation functions vectors, the spline coefficients matrices and the degrees of freedom vectors and matrices
- $i_{\max}, j_{\max}$ denote the total number of points representing a particular input airfoil, and the total number of airfoils input to form the blade frame, respectively
- mid denotes the grid index in the ξ – direction which represents the leading edge point on a particular airfoil

CHAPTER 1

INTRODUCTION

In 1983, the Solar Energy Research Institute (SERI), in conjunction with Airfoils, Incorporated, began the development of some special-purpose airfoil families for horizontal-axis wind turbines (HAWTs) [1]. Specific objectives for this effort were to

- (1) Restrain peak power and reduce sensitivity to insect accumulation
- (2) Lower blade oscillatory loads for better rotor fatigue life
- (3) Improve annual energy output at wind sites having mean annual wind speeds in the 10- to 14-mile-per-hour range.

Three special-purpose airfoil families were developed using the Eppler airfoil design and analysis code [2]. Each family consists of (1) a primary airfoil designed for the 75% radial position and (2) two secondary airfoils, designed to complement the primary airfoil that are located inboard and outboard at radial stations of 30% to 40%, and 95% [1]. Additional airfoils were derived, through interpolation and extrapolation, to aid the manufacturer in designing a blade using these airfoils. Figure 1 shows the expanded thin-airfoil family that is one outcome of this effort.

However, because of the highly twisted and tapered nature of the wind turbine blades, it is not always possible to generate a smooth surface for representing the desired blade through an interpolation based on the above limited available airfoil data. Therefore, an accurate and reliable interpolation technique becomes a very important tool in fabrication processes. Through this interpolation, it is possible to define the whole surface of the blade, and in so doing generate any number of airfoils to represent the blade. The coordinates of the generated airfoils can then be used to make templates for fabricating the blade.

Figure 1 Expanded Thin-Airfoil Family

An early attempt at this task reported that, because of the tapered and twisted nature of the blade, the data generated by interpolation could produce wavy surfaces toward the root of the blade [3]. Another approach, using the naval architecture versions of interpolation for ships and water-planes, avoids double concave/convex curvature, but the data generated are difficult to mold [4]. Most recently, Andrews and Van Doren [5] used one-dimensional tension spline interpolation in both the chordwise and spanwise directions, to generate a blade surface. They reported that the generated geometry is highly dependent on the tension parameter.

The objective of this study is to develop a FORTRAN computer code for generating a three-dimensional (3D) wind turbine blade through the input of two or more airfoils at some specific spanwise stations as the basis for interpolation. In addition, the generated 3D blade geometry should also satisfy the prescribed distributions of chord, thickness, and twist along the span of the blade. In this work, an algorithm used to define the smooth surface of a twisted and tapered wind turbine blade by using the bi-tension spline interpolation method [6, 7] is presented. The numerical algorithm consists of (1) creating the frame of a blade, (2) transforming the physical coordinates of the blade frame into computational coordinates, and finally (3) applying spline-under-tension interpolation to determine the coordinates of any point on the blade surface.

To show the versatility and usefulness of the tension spline interpolation code, two examples are given. Numerical results are presented in both graphic and tabular data forms. The solutions obtained in this work show that the computer code can be a powerful tool for generating the surface coordinates for any 3D blade.

In Chapter 2, the tension spline interpolation method for curves and surfaces is given. Chapter 3 discusses the blade frame creation process, which is necessary as the basis for the interpolation. To comply with the surface interpolation requirements and to facilitate the interpolation, a transformation scheme is developed and is presented in Chapter 4. In Chapter

5, the overall organization of the code and the program modules (subroutines) are given. Chapter 6 describes the operation of the code, including installation and execution of the code, input requirements, and output options. The results of applying the code to two sample problems and the numerical solutions are given in Chapter 7. Finally, Chapter 8 presents the conclusions of this research. The listing of the code is given in Appendix A.

CHAPTER 2

TENSION SPLINE INTERPOLATION

In general, there are two approaches to fitting a curve or surface to a given set of data [8]. One, called least-square fit, uses an approximating function to generate the curve or surface that does not necessarily pass through the given data points. Another, called exact fit, requires that the approximating function pass exactly through the given data points. In this work, the latter approach is used.

One available technique for solving interpolation problems is the cubic spline function [9]. The curve (or surface) generated by the cubic spline function passes through the given data points; has continuous slope or curvature; and, in general, has the local properties of smoothness. It seems that the cubic spline function is a very useful interpolation tool. However, because of the nature of the cubic spline function (a cubic polynomial), it occasionally produces some unwanted inflection points in the curve, especially at regions where the curvatures are negative [10]. It is therefore necessary to provide some way to remove these unwanted inflection points in the interpolated curve while still retaining the merits of cubic spline interpolation, i.e., smoothness and continuity. One promising method for overcoming this flaw is the "tension spline" function developed by Schweikert [10] for curves and extended by Späth [11] for surfaces.

Schweikert proposed that in order to remove the unwanted inflection points in the interpolated curve, the hyperbolic functions (exponential function) should be used instead of

the cubic polynomial. Physically, the interpolating function used in his paper represents the solution of a simply supported beam in tension. It is from this that the hyperbolic interpolation function derived the name of "tension" spline [10]. In this chapter, the equations used to describe the tension spline function for curves are given first. These are then extended for surface interpolation.

2.1 Tension Spline for Curves

Consider n data points (x_i, y_i) in the (x, y) plane, where $1 \leq i \leq n$ and $a \leq x_i \leq b$, see Figure

2. The tension spline $g(x)$ used to interpolate the given data on the partition

$$a = x_1 < x_2 < \dots < x_i < \dots < x_{n-1} < x_n = b \quad (2.1)$$

of the interval $[a, b]$ is expressed as (see Appendix B for derivation)

$$g_i(x) = \sum_{k=1}^4 A_{i,k} \Phi_k(x) \quad (2.2)$$

for each subinterval i , where $x_i \leq x \leq x_{i+1}$. The functions Φ_k in (2.2) are defined by

$$\begin{aligned} \Phi_1(x, \alpha_i, x_i, x_{i+1}) &= x - x_i \\ \Phi_2(x, \alpha_i, x_i, x_{i+1}) &= x_{i+1} - x \\ \Phi_3(x, \alpha_i, x_i, x_{i+1}) &= \psi(x - x_i, \alpha_i, x_i, x_{i+1}) \\ \Phi_4(x, \alpha_i, x_i, x_{i+1}) &= \psi(x_{i+1} - x, \alpha_i, x_i, x_{i+1}) \end{aligned} \quad (2.3)$$

where

$$\psi(x, \alpha_i, x_i, x_{i+1}) = \frac{\Delta x_i \sinh(\alpha_i x) - x \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \quad (2.4)$$

and

$$\Delta x_i = x_{i+1} - x_i$$

Figure 2 Data Points in the (x,y) Plane Representing a Curve on Which a 1-D Tension Spline is to be Applied

The four coefficients $A_{i,k}$ in (2.2) represent the spline coefficients for the interval $x_i \leq x \leq x_{i+1}$ and are expressed as (see Appendix B)

$$A_{i,k} = C(\Delta x_i, \nu_i) K_i \quad (2.5)$$

where

$$C(\Delta x_i, \nu_i) = \begin{bmatrix} 0 & 0 & \frac{1}{\Delta x_i} & 0 \\ \frac{1}{\Delta x_i} & 0 & 0 & 0 \\ -\frac{1}{\Delta x_i} \frac{1}{1-\nu_i} & \frac{-1}{1-\nu_i^2} & \frac{1}{\Delta x_i} \frac{1}{1-\nu_i} & \frac{-\nu_i}{1-\nu_i^2} \\ \frac{1}{\Delta x_i} \frac{1}{1-\nu_i} & \frac{\nu_i}{1-\nu_i^2} & -\frac{1}{\Delta x_i} \frac{1}{1-\nu_i} & \frac{1}{1-\nu_i^2} \end{bmatrix} \quad (2.6)$$

$$K_i = \begin{bmatrix} y_i \\ y'_i \\ y_{i+1} \\ y'_{i+1} \end{bmatrix} \quad (2.7)$$

and

$$\nu_i = \left[\frac{d}{dx} \psi(x - x_i, \alpha_i, x_i, x_{i+1}) \right]_{x=x_{i+1}} \quad (2.8)$$

The coefficient α appearing in the above equations is called the tension factor and is a constant in each subinterval i . Superscript ' in (2.7) denotes the derivative of y with respect to x . It can be shown [5] that

$$\psi(x, 0, x_i, x_{i+1}) = \lim_{\alpha \rightarrow 0} \psi(x, \alpha, x_i, x_{i+1}) = \frac{x^3 - x}{\Delta x_i^2} \quad (2.9)$$

corresponds to the cubic spline interpolation. Also, notice that if the tension factor becomes very large, the interpolating curve is degraded to a piecewise linear interpolation.

The values of y'_i , appearing in (2.7), are determined by the following procedures:

- (1) For $i = 1$, the boundary value y'_1 is determined by the forward difference as

$$y'_1 = \frac{y_2 - y_1}{x_2 - x_1} \quad (2.10)$$

- (2) For $2 \leq i \leq n-1$, derivatives y'_i are determined by using the curvature continuity condition to set up a system consisting of the following $n-2$ equations (see Appendix C for derivation):

$$t_{i-1}y'_{i-1} + (t_{i-1}v_{i-1} + t_i v_i)y'_i + t_i y'_{i+1} = t_{i-1}(1 + v_{i-1})\frac{\Delta y_{i-1}}{\Delta x_{i-1}} + t_i(1 + v_i)\frac{\Delta y_i}{\Delta x_i} \quad (2.11)$$

where

$$t_i = \frac{\alpha_i^2 \Delta x_i \sinh(\alpha_i \Delta x_i)}{(v_i^2 - 1)(\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i)} \quad (2.12)$$

for $1 \leq i \leq n-1$ and v_i as in (2.8).

- (3) For $i = n$, the boundary value y'_n is solved for by the backward difference as

$$y'_n = \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \quad (2.13)$$

If the curvature at the boundary points are specified as zero, the corresponding spline is called a natural spline [12].

The system of equations (2.11) in matrix form (with the boundary derivative values included as known parameters) is

$$[CM][P] = [B] \quad (2.14)$$

here

$$CM = \begin{bmatrix} b_2 & c_2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ a_3 & b_3 & c_3 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_i & b_i & c_i & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_{n-2} & b_{n-2} & c_{n-2} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & a_{n-1} & b_{n-1} \end{bmatrix}$$

$$P = \begin{bmatrix} y'_2 \\ y'_3 \\ \vdots \\ y'_i \\ \vdots \\ y'_{n-2} \\ y'_{n-1} \end{bmatrix}; \quad B = \begin{bmatrix} B_2 \\ B_3 \\ \vdots \\ B_i \\ \vdots \\ B_{n-2} \\ B_{n-1} \end{bmatrix}$$

the coefficients a, b, and c in matrix CM are

$$a_i = t_{i+1}$$

$$b_i = t_i v_i + t_{i+1} v_{i+1}$$

$$c_i = t_{i+1}$$

and the coefficients in vector B are

$$B_2 = t_1(v_1 + 1) \frac{y_2 - y_1}{x_2 - x_1} + t_2(v_2 + 1) \frac{y_3 - y_2}{x_3 - x_2} + y'_1 t_1$$

$$B_i = t_i(v_i + 1) \frac{y_{i+1} - y_i}{x_{i+1} - x_i} + t_{i+1}(v_{i+1} + 1) \frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}}; \quad \text{for } 3 \leq i \leq n-2$$

$$B_{n-1} = t_{n-2}(v_{n-2} + 1) \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} + t_{n-1}(v_{n-1} + 1) \frac{y_n - y_{n-1}}{x_n - x_{n-1}} + y'_n t_{n-1}$$

The coefficient matrix CM of system (2.14) is a tridiagonal, symmetric, and diagonally dominant matrix with positive diagonal elements. It can be solved by the efficient Thomas algorithm [13] for the values of y'_i ($2 \leq i \leq n-1$) contained in P.

When the derivatives y'_i are known, the four spline coefficients $A_{i,k}$ for each subinterval i can be determined from (2.5). Equation (2.2) is then used to determine the interpolated y-value for any x in any subinterval of $\{a, b\}$.

The curve represented by the given n data points is now defined. Depending on the value chosen for the tension factor, α , the interpolating curve can be varied between a cubic polynomial and a piecewise linear polynomial.

2.2 Tension Spline for Surfaces

To extend the tension spline of curves to that for surface interpolation is straightforward. Let $u_{i,j} = u(\xi_i, \eta_j)$, $i = 1, 2, \dots, n$, and $j = 1, 2, \dots, m$ denote the values of a function given at the nodes of a rectangular computational domain grid with

$$\xi_1 < \xi_2 < \dots < \xi_n, \quad n \geq 2 \quad (2.16)$$

and

$$\eta_1 < \eta_2 < \dots < \eta_m, \quad m \geq 2 \quad (2.17)$$

in the (ξ, η) plane. The surface tension spline $f(\xi, \eta)$ used to interpolate the given data on the region described in (2.16) and (2.17) is expressed as [11,12]

$$f_{i,j}(\xi, \eta) = \sum_{k=1}^4 \sum_{l=1}^4 A_{i,j,k,l} \Phi_k(\xi, \alpha, \xi_i, \xi_{i+1}) \Phi_l(\eta, \beta, \eta_j, \eta_{j+1}) \quad (2.18)$$

for each rectangle $\{\xi_i \leq \xi \leq \xi_{i+1}; \eta_j \leq \eta \leq \eta_{j+1}\}$.

The nature of the spline function is such that the computational value of $f_{i,j}$ at the nodes of the rectangular grid is $u_{i,j}$, i.e.,

$$f_{i,j}(\xi_i, \eta_j) = u_{i,j}$$

The interpolation functions Φ in (2.18) take the same form as (2.3) and (2.4). The coefficients α and β in (2.18) are the tension parameters in the ξ and η directions, respectively.

The coefficient matrix $A_{i,j,k,l}$ (for $k, l = 1, 2, 3, 4$) in (2.18) is a 4-by-4 matrix at each rectangle and is formed by the matrix products

$$A_{i,j,k,l} = C(\Delta\xi_i, v_i) K_{i,j} [C(\Delta\eta_j, w_j)]^T \quad (2.19)$$

where

$$C(g, h) = \begin{bmatrix} 0 & 0 & \frac{1}{g} & 0 \\ \frac{1}{g} & 0 & 0 & 0 \\ -\frac{1}{g} \frac{1}{1-h} & \frac{-1}{1-h^2} & \frac{1}{g} \frac{1}{1-h} & \frac{-h}{1-h^2} \\ \frac{1}{g} \frac{1}{1-h} & \frac{h}{1-h^2} & -\frac{1}{g} \frac{1}{1-h} & \frac{1}{1-h^2} \end{bmatrix} \quad (2.20)$$

$$K_{i,j} = \begin{bmatrix} u_{i,j} & q_{i,j} & u_{i,j+1} & q_{i,j+1} \\ p_{i,j} & r_{i,j} & p_{i,j+1} & r_{i,j+1} \\ u_{i+1,j} & q_{i+1,j} & u_{i+1,j+1} & q_{i+1,j+1} \\ p_{i+1,j} & r_{i+1,j} & p_{i+1,j+1} & r_{i+1,j+1} \end{bmatrix} \quad (2.21)$$

$$v_i = \left[\frac{d}{d\xi} \psi(\xi - \xi_i, \alpha_i, \xi_i, \xi_{i+1}) \right]_{\xi=\xi_{i+1}} \quad (2.22)$$

$$w_j = \left[\frac{d}{d\eta} \psi(\eta - \eta_j, \beta_j, \eta_j, \eta_{j+1}) \right]_{\eta=\eta_{j+1}} \quad (2.23)$$

$$\Delta\xi_i = \xi_{i+1} - \xi_i$$

$$\Delta\eta_j = \eta_{j+1} - \eta_j$$

and the superscript T in (2.19) denotes the transpose. The elements $u_{i,j}$, $p_{i,j}$, $q_{i,j}$, and $r_{i,j}$ in matrix $K_{i,j}$ of (2.21) are the abbreviations of the following function values and its partial derivatives:

$$\begin{aligned}
u_{i,j} &= u(\xi_i, \eta_j) \\
p_{i,j} &= \frac{\partial u(\xi_i, \eta_j)}{\partial \xi} \\
q_{i,j} &= \frac{\partial u(\xi_i, \eta_j)}{\partial \eta} \\
r_{i,j} &= \frac{\partial^2 u(\xi_i, \eta_j)}{\partial \xi \partial \eta}
\end{aligned} \tag{2.24}$$

The partial derivatives $p_{i,j}$, $q_{i,j}$, and $r_{i,j}$ are determined as follows.

For $1 \leq j \leq m$, the boundary values $p_{1,j}$ and $p_{n,j}$ are solved for by forward and backward difference respectively according to

$$p_{1,j} = \frac{u_{2,j} - u_{1,j}}{\xi_2 - \xi_1} \tag{2.25}$$

$$p_{n,j} = \frac{u_{n,j} - u_{n-1,j}}{\xi_n - \xi_{n-1}} \tag{2.26}$$

For $1 \leq i \leq n$, the boundary values $q_{i,1}$ and $q_{i,m}$ are solved for by forward and backward difference respectively as

$$q_{i,1} = \frac{u_{i,2} - u_{i,1}}{\eta_2 - \eta_1} \tag{2.27}$$

$$q_{i,m} = \frac{u_{i,m} - u_{i,m-1}}{\eta_m - \eta_{m-1}} \tag{2.28}$$

For $j = 1$ and m , the boundary values $r_{1,j}$ and $r_{n,j}$ are solved for by forward and backward difference respectively as

$$r_{1,j} = \frac{q_{2,j} - q_{1,j}}{\xi_2 - \xi_1} \tag{2.29}$$

$$r_{n,j} = \frac{q_{n,j} - q_{n-1,j}}{\xi_n - \xi_{n-1}} \quad (2.30)$$

This completes the evaluation of all the required partial derivatives at the boundary nodes. In order to determine the partial derivative values of the internal nodes of the rectangular grid, the curvature continuity condition is used to set up the following $2n + m + 2$ linear systems of coupling equations:

For $j = 1, \dots, m$

$$t_{i-1}p_{i-1,j} + (t_{i-1}v_{i-1} + t_i v_i)p_{i,j} + t_i p_{i+1,j} =$$

$$t_{i-1}(1 + v_{i-1}) \frac{u_{i,j} - u_{i-1,j}}{\Delta \xi_{i-1}} + t_i(1 + v_i) \frac{u_{i+1,j} - u_{i,j}}{\Delta \xi_i} \quad (2.31)$$

where $2 \leq i \leq n-1$.

For $i = 1, \dots, n$

$$s_{j-1}q_{i,j-1} + (s_{j-1}w_{j-1} + s_j w_j)q_{i,j} + s_j q_{i,j+1} =$$

$$s_{j-1}(1 + w_{j-1}) \frac{u_{i,j} - u_{i,j-1}}{\Delta \eta_{j-1}} + s_j(1 + w_j) \frac{u_{i,j+1} - u_{i,j}}{\Delta \eta_j} \quad (2.32)$$

where $2 \leq j \leq m-1$.

For $j = 1$ and m

$$t_{i-1}r_{i-1,j} + (t_{i-1}v_{i-1} + t_i v_i)r_{i,j} + t_i r_{i+1,j} =$$

$$t_{i-1}(1 + v_{i-1}) \frac{q_{i,j} - q_{i-1,j}}{\Delta \xi_{i-1}} + t_i(1 + v_i) \frac{q_{i+1,j} - q_{i,j}}{\Delta \xi_i} \quad (2.33)$$

where $2 \leq i \leq n-1$.

For $i = 1, \dots, n$

$$s_{j-1}r_{i,j-1} + (s_{j-1}w_{j-1} + s_j w_j)r_{i,j} + s_j r_{i,j+1} =$$

$$s_{j-1}(1 + w_{j-1})\frac{p_{i,j} - p_{i,j-1}}{\Delta\eta_{j-1}} + s_j(1 + w_j)\frac{p_{i,j+1} - p_{i,j}}{\Delta\eta_j} \quad (2.34)$$

where $2 \leq j \leq m-1$.

In equations (2.31) to (2.34), the values of v_i and w_j are determined using (2.22) and (2.23) respectively, and the values of t_i and s_j are determined as

$$t_i = \frac{\alpha_i^2 \Delta\xi_i \sinh(\alpha_i \Delta\xi_i)}{(v_i^2 - 1)(\sinh(\alpha_i \Delta\xi_i) - \alpha_i \Delta\xi_i)} \quad (i = 1, \dots, n-1) \quad (2.35)$$

$$s_j = \frac{\beta_j^2 \Delta\eta_j \sinh(\beta_j \Delta\eta_j)}{(w_j^2 - 1)(\sinh(\beta_j \Delta\eta_j) - \beta_j \Delta\eta_j)} \quad (j = 1, \dots, m-1) \quad (2.36)$$

The above tridiagonal systems of equations, (2.31)-(2.34), is also solved by the Thomas algorithm for the interior values of $p_{i,j}$, $q_{i,j}$ and $r_{i,j}$ (for $i = 2, \dots, n-1$; $j = 2, \dots, m-1$).

When all the partial derivative quantities and hence $K_{i,j}$ are known, the coefficient matrix $A_{i,j,k,l}$ for each and every rectangle of the computational domain grid can be found from (2.19). Equation (2.18) is then used to determine the interpolated u -value for any ξ and η in the computational domain, where

$$\xi_1 \leq \xi \leq \xi_m \quad \text{and} \quad \eta_1 \leq \eta \leq \eta_m$$

In such a way the whole surface can be defined. Depending on the value chosen for the tension factors, α and β , the interpolation can be varied between a bi-cubic spline and a piecewise linear spline.

CHAPTER 3

CREATING THE BLADE FRAME

An aerodynamic blade is characterized not only by the types of airfoils used along its span, but also by their chord, thickness, and twist distributions. The frame of a blade can be created by placing scaled and twisted airfoil data at particular span stations along the blade. This frame, after transformation, may then be used as the basis for the bi-tension spline interpolation presented in Chapter 2.

Described next is the blade frame creation process, which involves the following four main steps:

- (1) Selection of span stations of the blade at which airfoils will be placed
- (2) Selection of the type of airfoil for placement at these span stations
- (3) Scaling the airfoils to the required chord and maximum thickness at their respective span stations
- (4) Twisting the airfoils according to the required angle of twist at their respective span stations.

3.1 Choice of Span Stations of the Blade at which to Place Airfoils

The frame of the blade is to be used as the basis for interpolation. In order that this interpolation be as accurate as possible, it is advised that as many airfoils as possible be placed along the span of the blade in order to form the blade frame. The manner in which these airfoils is placed depends on the nature of the chord, thickness, and twist distributions along the span

of the respective blade. More airfoils should be placed in regions of the span where the slope and/or curvature of the aforementioned distributions is high. This is so that the generated blades, chord, thickness, and twist distributions more closely match those of the desired blade. Conversely, it is not critical to have a high density of airfoils in regions along the span where all distributions are close to linear.

The availability of the correct airfoils is also a crucial factor in determining what span stations will have airfoils. In this work, results of [1] are used as the basis for placing the airfoils. The chosen span stations are represented by a Z-coordinate.

3.2 Choice of Airfoils

In a well-designed blade the airfoils that form it enhance its structural integrity and maximize its aerodynamic performance.

The structural integrity of the blade is enhanced by using thicker airfoils in the root region and gradually tapering the maximum thickness of the airfoils to a minimum at the tip of the blade. The blade's aerodynamic performance is optimized if its component airfoils are designed to maximize the desired forces, under given environmental conditions, in such a way that the structural integrity of the blade is not undermined.

This work utilizes already designed airfoils and places them at the chosen span stations. Figure 1 shows some existing low-speed airfoils [1] that could be used to form the frame of a blade for wind turbine rotors. Each airfoil has had its surface coordinates normalized with respect to its chord.

3.3 Scaling the Airfoils

For most of the airfoil data catalog, e.g., Abbott and Von Doenhoff [14], the X and Y coordinates are given in a form normalized with respect to the chord of the airfoil. To make the airfoil data conform to the required chord and thickness dimensions at the respective span stations, these data must be scaled. In the scaling process, the nondimensional coordinates of an airfoil are first scaled to the ones with the required maximum thickness, and then translated into the true dimensions with the desired chord-length. The modified Smoothing and Scaling code of Morgan [15] and Tu and Scott [16] was used in this work.

3.3.1 Scaling the Airfoils to the Required Maximum Thickness-to-Chord Ratio

The maximum thickness of an airfoil is the maximum straight-line distance between the upper and lower airfoil surfaces measured perpendicularly to the chord of the airfoil. In the case of a chord-normalized airfoil, this would represent the maximum thickness-to-chord ratio of the airfoil. The required maximum thickness-to-chord ratio at each blade frame span station is obtained from specification, or through interpolating an already designated maximum thickness-to-chord distribution at the particular span station.

The scaling code uses as input the camberline and thickness distributions generated by the smoothing code [16]. It first determines the maximum thickness of the airfoil and its position. The thickness distribution is then multiplied by a scale factor equal to the ratio of the new maximum thickness to the old maximum thickness. The camberline distribution and new thickness distribution are then combined to generate new thickness-scaled airfoil coordinates.

3.3.2 Scaling the Airfoils to the Required Chord Length

The chord length of an airfoil is the straight-line distance between the leading edge point and trailing edge point of the airfoil. In the case of chord-normalized airfoil data, the chord length is unity. For each blade frame span station airfoil, a required chord length can be obtained from specification or from an already designated chord distribution along the span of the blade. To scale the airfoil to the required chord length, the X and Y coordinates of the normalized airfoil are multiplied by the required chord length at the particular span station.

3.4 Twisting the Airfoils

To have better control and to obtain the desired aerodynamic force and moment distributions along the span of the blade, the blade must be twisted about its twist center. The prescribed twist angle for any particular airfoil can be obtained from an already designated twist distribution for the blade.

The center of twist for an airfoil is the aerodynamic center of the airfoil. This is because the moment coefficient about the aerodynamic center is independent of the angle of attack [17]. For most airfoils, the twist center is located close to the quarter-chord distance from the leading edge. In this work, the aerodynamic center can be set at the quarter-chord point, leading edge point, or any other point specified by the user. Before each airfoil is twisted, it must be offset by a distance equal to the distance between its leading edge and its twist center. The result is that the axis of twist centers for the component airfoils of the blade is the Z-axis. The translation representing the offset is

$$X_{os} = X - OS_x \quad (3.1)$$

$$Y_{os} = Y - OS_y \quad (3.2)$$

where OS_x and OS_y are the X and Y coordinates of the twist center, respectively.

The rotation transformation representing the twisting of the offset airfoil about its twist center is

$$X_{tw} = X_{os} \cos \theta - Y_{os} \sin \theta \quad (3.3)$$

$$Y_{tw} = X_{os} \sin \theta + Y_{os} \cos \theta \quad (3.4)$$

where θ is the twist angle.

Figures 3-5 summarize the blade frame creation processes. After placement of the scaled and twisted airfoils at their respective span stations, the resultant blade frame is shown in Figure 5.

Figure 3 Chord-Normalized Input Airfoils at Respective Span Stations

Figure 4 Input Airfoils That Have Been Scaled and Offset, and Then Placed at Their Respective Span Stations

Figure 5 Input Airfoils That Have Been Scaled, Offset, and Twisted, and Then Placed at Their Respective Span Stations

CHAPTER 4

TRANSFORMATION OF THE BLADE FRAME

Now that the blade frame has been formed, the tension spline method is applied to generate the whole blade surface. As shown in equations (2.16) and (2.17), the interpolation requires that the independent variables increase monotonically and that the computational domain be a rectangular mesh. However, the X, Y, and Z coordinates of the blade frame described in Chapter 3 do not conform to these requirements. It is therefore necessary to transform the coordinates into a computational domain coordinate system that is compatible with the interpolation scheme. In addition, to facilitate the surface interpolation scheme and the ease of programming, it is also desired that the computational domain be confined to a unit square region. The bi-tension spline interpolation will be carried out in the computational domain.

In this work the transformation from the physical domain to the computational domain consists of mathematical relationships mapping one domain to another, such that changes to the grid are direct and rapidly obtained and transformation data are readily available for use. To set up and use the computational-domain coordinates as the basis for surface interpolation, the following sequence of manipulations is performed:

- (1) The normalized Z-coordinate, η , of each airfoil's span station is computed.
 - (2) The polygonal arc length of each airfoil is computed and normalized.
 - (3) New X and Y coordinates for each airfoil are obtained by interpolation to form a regular grid net.
 - (4) Finally, the computational domain is defined and is ready to use as a basis for interpolation.
- These manipulations are now described.

4.1 Calculation of the Normalized Z-Coordinate

The normalized Z-coordinate is represented by

$$\eta_j = \frac{(Z_j - Z_1)}{(Z_{jmax} - Z_1)} \quad (4.1)$$

where $1 \leq j \leq jmax$, and $jmax$ is the total number of input airfoils used to define the blade frame. The transformation equation (4.1) transforms the Z-coordinate into a unit domain

$$0 \leq \eta_j \leq 1 \quad (4.2)$$

4.2 Computation and Normalization of the Polygonal Arc length for Each Input Airfoil

The polygonal arc length is the polygonal distance between the trailing edge of the airfoil and any point on the airfoil's surface. This distance is measured clockwise beginning at the lower surface trailing edge through all points on the surface prior to the point in question. The polygonal arc length, S , for each airfoil is determined as follows:

$$S_1 = 0$$

and

$$S_i = \sqrt{[(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2]} + S_{i-1} \quad (4.3)$$

for $2 \leq i \leq imax$, where $imax$ is the number of points used to define the airfoil surface.

To normalize the polygonal arc length, the following equation is used:

$$\bar{S}_i = \frac{S_i}{S_{imax}} \quad (4.4)$$

where $1 \leq i \leq imax$. Equation (4.4) transforms the S-coordinate into a unit domain

$$0 \leq \bar{S}_i \leq 1 \quad (4.5)$$

4.3 Interpolation of New X and Y Coordinates for Each Airfoil

The transformation relations stated above map the computational domain onto a unit square, but the grid net is not rectangular. It is therefore necessary to define a new set of data for each airfoil through interpolation such that a rectangular grid net is obtained. In the early stages of this algorithm's development, the use of an equally spaced set of normalized polygonal arc lengths was adequate for the interpolation. Because of the high gradient of the slope and curvature near the nose of the airfoils, that area could not be well defined. To overcome this discrepancy, a stretching function, which automatically concentrates the data points about the leading edge, is introduced.

Accordingly, the approach adopted is first to generate a set of equally spaced normalized polygonal arc lengths, $\bar{\xi}_i$; then to refine the equally spaced polygonal arc lengths about the nose of the airfoils through a stretching function; and ultimately to use the resulting set of polygonal arc lengths, ξ_i , to interpolate for new values of X and Y, for each airfoil, as functions of their respective normalized polygonal arc lengths computed by equation (4.4). Following are the relations representing the approach.

Let $imax$ be the number of desired computational grid points on each airfoil. Then for an equally spaced grid, the increment $\Delta\bar{\xi}$ would be

$$\Delta\bar{\xi} = \frac{1.0}{imax - 1} \quad (4.6)$$

and the new set of equally spaced normalized polygonal arc lengths would be

$$\bar{\xi}_i = \Delta\bar{\xi} \times (i - 1) \quad (4.7)$$

for $1 \leq i \leq imax$. To concentrate the data points about the leading edge automatically, the following stretching function is used [12]:

$$\xi_i = \bar{\xi}_{mid} \left\{ 1 + \frac{\sinh[\tau(\bar{\xi}_i - B)]}{\sinh(\tau B)} \right\} \quad (4.8)$$

where

$$B = \frac{1}{2\tau} \ln \left[\frac{1 + (e^\tau - 1)\bar{\xi}_{mid}}{1 + (e^{-\tau} - 1)\bar{\xi}_{mid}} \right] \quad (4.9)$$

and τ is the stretching parameter, which varies from zero (no stretching) to infinity (most refinement) about $\bar{\xi}_{mid}$. For each airfoil, $\bar{\xi}_{mid}$ represents the leading edge of the airfoil. This value varies among airfoils; experience shows that a value of 0.5 is close enough to the leading edge. Therefore, $\bar{\xi}_{mid}$ is set equal to 0.5 for all calculations. Figure 6 shows how the refinement (grid spacing in ξ) varies with different values of τ about $\bar{\xi}_{mid} = 0.5$. The same number of grid points is used for three different τ values. Experience shows that for the best result, τ should be set to a value between 4.0 and 6.0, depending on the value of $imax$ chosen.

New values of X and Y for each airfoil are then interpolated at the new set of the normalized polygonal arc lengths, represented by equation (4.8), using the 1-D tension spline described in Chapter 2.

4.4 Defining and Using the Computational Domain Coordinates.

As the result of the data processing steps presented above, the X and Y values become a function of ξ and η , i.e.,

$$X = X(\xi, \eta)$$

and

$$Y = Y(\xi, \eta)$$

The blade frame is now defined by (X_{ij}, ξ_i, η_j) and (Y_{ij}, ξ_i, η_j) .

By performing two separate 2-D interpolations on the computational domain of the blade frame, the physical coordinates (X, Y, Z) of any point on the surface of the blade can be determined.

Specifically, to find the X coordinate of a point on the surface of the blade, the surface tension spline method described in Chapter 2 is applied to the (X_{ij}, ξ_i, η_j) data set; see Figure 7.

(a) $\tau = 0.0001$

(b) $\tau = 4.5$

(c) $\tau = 10$

Figure 6 Variation in Grid Spacing with Three Different Stretching Parameters for an Airfoil Represented by 65 Points

Figure 7 Computational Domain with X as the Dependent Variable

Similarly, to find the corresponding Y coordinate at the same point, the (Y_{ij}, ξ_i, η_j) data set is used; see Figure 8. The Z coordinate of the point is determined by the inverse transformation of equation (4.1).

In such a way, the physical coordinates of any point in the computational domain of the blade frame can be determined. The whole blade surface therefore is defined.

Figure 8 Computational Domain with Y as the Dependent Variable

CHAPTER 5

PROGRAM DESCRIPTION

Presented in this chapter are the program structure, descriptions of the function of each subroutine, and a definition of all common block variables used.

5.1 Program Structure

In order that the whole program structure be appropriately described, the program is broken down into six modules: Input, Blade frame creation, Transformation, Interpolation, Partial derivatives (called by the Interpolation module), and Output. Figure 9 shows the order in which these modules are accessed by the main program.

Figure 9

Order of Access of Modules in the BLADE Code

5.1.1 Input Module

The input module's purpose is primarily to accept all necessary input data either from the terminal (subroutine TINPUT) or from the data file (subroutine DINPUT). This module also controls the scaling process of the input airfoils.

Subroutine INPUT is the driver of the input module from which either subroutine TINPUT or DINPUT is called. Both TINPUT and DINPUT call subroutines SPT11D and CURVE1 and function CURVE2. The smoothing and scaling codes, used for the scaling process, are run in batch mode from subroutines DINPUT and TINPUT. Figure 10 shows the structure of the input module.

Figure 10 Structure of the Input Module

5.1.2 Blade Frame Creation Module

The blade frame creation module's purpose is to create the frame of the blade from the scaled input data. Subroutine SCOSTW is the sole module's subroutine. In this module, each maximum-thickness-scaled input airfoil is scaled to the prescribed chord, offset to the desired twist center, twisted by the prescribed twist angle, and then placed at the desired span station.

5.1.3 Transformation Module

The transformation module's purpose is to transform the physical coordinates of the blade frame into computational coordinates. Subroutine TRANS is the driver of the transformation module. Subroutines SPT11D and CURVE1 and function CURVE2 are called by subroutine TRANS. Figure 11 shows the structure of the transformation module.

Figure 11 Structure of the Transformation Module

5.1.4 Partial Derivatives Module

The purpose of the partial derivatives module is to control and determine the partial derivatives of the dependent variable (X or Y) with respect to ξ , η , and ξ and η respectively. The partial derivatives module is called by the interpolation module. Subroutine PDERIV is the driver of the partial derivatives module whose structure is shown in Figure 12.

Figure 12 Structure of the Partial Derivatives Module

5.1.5 Interpolation Module

The purpose of the interpolation module is to control and perform the surface interpolation of the transformed blade frame, in the computational domain, so that the physical coordinates representing an airfoil at any span station of the blade can be determined. Subroutine CHOICE is the first routine called by the interpolation module. Subsequently, subroutine SECT is the driver of the interpolation procedure. Each time the coordinates of an airfoil are being determined in subroutine SECT, subroutines GETX and GETY are called. Both GETX and GETY call the partial derivatives module to provide required information. Figure 13 shows the structure of the interpolation module.

Figure 13 Structure of the Interpolation Module

5.1.6 Output Module

The purpose of the output module is to control and perform the viewing, plotting, and printing of input and generated graphs, tables, or data. The driver of the output module is subroutine PLOTPR from which subroutine HRDOUT is called. Subroutine HRDOUT controls the graphic output device options. Twenty-four programs were developed to facilitate the viewing, plotting, and printing of all the graphics in the program. These programs, which are run in batch mode from subroutine HRDOUT, call upon various GRAFMATIC, PLOTMATIC, and PRINTMATIC¹ subroutines to view, plot, and print the graphs. Figure 14 shows the structure of the output module.

5.2 Subroutine Functions

Following is a description of the function of each subroutine in the FORTRAN program developed:

- BLADE : Main program that controls overall program execution.
- CHOICE : Allows the user to make a choice of generating either a spanwise station airfoil or a three-dimensional blade defined by at least three airfoils.
- CURVE1 : Determines the parameters necessary to compute an interpolatory 1-D tension spline through a series of points.
- CURVE2 : Interpolates a curve at a given point using a spline-under-tension.
- CHZE : Determines the two 4-by-4 matrices, CHZY and CHZX equation (2.20) necessary for computations in subroutine COEFM.
- CKFILE : Checks and deletes an existing file.

¹ GRAFMATIC, PLOTMATIC, AND PRINTMATIC are trademarks of Microcompatibles, Inc. For those who do not have this graphic package, programs listed in Figure 14 should be replaced.

Subroutine PLOTPR		
HRDOUT		
VICPINP ² (run in batch)	PLCPINP ³ (run in batch)	PRCPINP ⁴ (run in batch)
VICPGEN (run in batch)	PLCPGEN (run in batch)	PRCPGEN (run in batch)
VICHNPL (run in batch)	PLCHNPL (run in batch)	PRCHNPL (run in batch)
VITOP (run in batch)	PLTOP (run in batch)	PRTOP (run in batch)
VIFRONT (run in batch)	PLFRONT (run in batch)	PRFRONT (run in batch)
VICHORD (run in batch)	PLCHORD (run in batch)	PRCHORD (run in batch)
VITWIST (run in batch)	PLTWIST (run in batch)	PRTWIST (run in batch)
VITHICK (run in batch)	PLTHICK (run in batch)	PRTHICK (run in batch)

Figure 14 Structure of the Output Module

-
- 2 Programs listed in this column are for viewing the graphic output on the screen.
 - 3 Programs listed in this column are for sending the graphic output to the plotter.
 - 4 Programs listed in this column are for sending the graphic output to the printer.

COEFM : Determines the 4-by-4 coefficients matrix equation (2.19) for each sector of the computational domain grid.

DINPUT : Called by INPUT to perform the "data file" input mode option. It also prepares all airfoil data for thickness scaling. Runs the smoothing and scaling codes in batch mode.

GETX : Determines the X-coordinate of all points on the surface of an airfoil at a particular span station of the blade.

GETY : Determines the Y-coordinate of all points on the surface of an airfoil at a particular span station of the blade.

GETU : Called by GETX and GETY to determine each interpolated dependent variable.

HRDOUT : Called by PLOTPR for controlling the graphic output device options.

INPUT : Accepts all major input necessary to successfully run the program.

KAY : Creates a 4-by-4 matrix, equation (2.21), containing all the values of u and its partial derivatives, at the four nodes of each sector in the computational domain grid.

PDERIV : Determines the boundary values of the partial derivatives of u (the dependent variable), with respect to ξ , η , and the cross derivative of u with respect to ξ and η according to equations (2.25)-(2.30). It then calls PUX, QUY, RUXY1, and RUXY2 to determine those partial derivatives it has not yet determined according to equations (2.13)-(2.34).

PLOTPR : Plots and prints input and generated data.

PHI : Determines the two interpolation function vectors, in the ξ and η directions, for each sector of the computational domain grid, by using equations (2.3) and (2.4).

- PUX** : Determines the partial derivative of u with respect to ξ at every internal node of the computational domain grid. These derivatives are contained in an array P . Uses equations (2.25) and (2.26) as boundary conditions to solve systems, equation (2.31), for P_{ij} .
- QUY** : Determines the partial derivative of u with respect to η at every internal node of the computational domain grid. These derivatives are contained in an array Q . Uses equations (2.27) and (2.28) as boundary conditions to solve systems, equation (2.32), for Q_{ij} .
- RUXY1** : Determines the cross derivative of u with respect to ξ and η at each boundary node of the computational domain grid where η has a value of 0 or 1 and ξ does not have values of 0 or 1. These derivatives are contained in an array R . Uses equations (2.29) and (2.30) as boundary conditions to solve systems, equations (2.33), for R_{ij} .
- RUXY2** : Determines the cross derivative of u with respect to ξ and η at each node of the computational domain grid where, η does not have values of 0 and 1. Uses equations (2.29) and (2.30) as boundary conditions to solve system, equation (2.34), for R_{ij} .
- SCOSTW** : Generates new X and Y coordinates of airfoils caused by chord scaling, offsetting, and then twisting the airfoils. See Chapter 3.
- SECT** : Controls the 2-D interpolation process of the points on the blade surface.
- SORT** : Finds the computational domain grid sector in which the point (ξ, η) lies. It returns the (ξ, η) coordinate of the bottom left corner node of the sector.
- SPT11D** : The main routine that calls **CURVE1** and **CURVE2** to perform a one-dimensional tension spline interpolation of a given curve at a given point.

THOMAS : Solves a tridiagonal system of equations using the Thomas algorithm.

TINPUT : Called by INPUT to perform the "terminal" input mode option. It also prepares all airfoil data for thickness scaling. Runs the smoothing and scaling codes in batch mode.

TRANS : Transforms the physical domain coordinates (X,Y,Z) into the computational coordinates (X, ξ , η) and (Y, ξ , η) by the method in Chapter 4.

TRANSPO : Finds the transpose of a matrix (See equation [2.19]).

5.3 Critical Variable Definitions

The critical variables used in the computer code are contained in twenty-nine named common blocks. In defining the variables we will therefore list them by common block as is done below:

C1 : TH1 - Vector containing twist angles of input airfoils.
 OSX - Vector containing the chord-normalized twist center X coordinates.
 OSY - Vector containing the chord-normalized twist center Y coordinates.
 CL - Vector containing chord lengths of input airfoils.

C2 : X - Array containing the X coordinates of thickness-scaled input airfoils.
 Y - Array containing the Y coordinates of thickness-scaled input airfoils.

C3 : X1 - Array containing the X coordinates of scaled, offset, and twisted input airfoils for all input span stations.
 Y1 - Array containing the Y coordinates of scaled, offset, and twisted input airfoils for all input span stations.

C4 : ZI - Vector containing the normalized polygonal arc length coordinates (ξ) for the input span stations.
 ETA- Vector containing the normalized Z coordinates (η) for the input span stations.

- C5 : XMIN - Vector containing the minimum X coordinates at each span station of X1.
XMAX - Vector containing the maximum X coordinates at each span station of X1.
- C6 : DZI - Vector containing magnitudes of increment of ξ .
DETA - Vector containing magnitudes of increment of η .
- C7 : P - Array containing partial derivatives of u with respect to ξ at each η .
Q - Array containing partial derivatives of u with respect to η at each ξ .
R - Array containing cross derivatives of u with respect to ξ and η at each ξ and η .
- C8 : IC - Logic variable indicating user's choice of blade generation.
ZIT - Z coordinate of span station at which user wants interpolation carried out.
- C9 : AL - Vector of normalized tension factors for each of the nodal intervals in the ξ direction.
BE - Vector of normalized tension factors for each of the nodal intervals in the η direction.
- C10 : V, W - Vectors containing necessary coefficients in the tridiagonal systems of equations (equations [2.31]-[2.34]) that are solved to find the required partial derivatives.
- C11 : ALPHA, BETA - Input values of tension factors in the ξ and η directions, respectively.
- C12 : Z - Vector containing the Z coordinates of the input span stations.
- C13 : U - Matrix containing the dependent variable, i.e., Y or X coordinates of the scaled, offset, and twisted airfoils at all input span stations.

- C14A : XU, XL - Matrices containing the X coordinates of the upper and lower surfaces, respectively, of all the input airfoils.
- C14B : YU, YL - Matrices containing the Y coordinates of the upper and lower surfaces, respectively, of all the input airfoils.
- C15 : S, T - Vectors containing necessary coefficients in the tridiagonal systems of equations (equations [2.31]-[2.34]) that are solved to find required partial derivatives.
- C16 : XIT, YIT - Internodal values of ξ and η , respectively, that are currently being used to interpolate for the Y- or X-coordinate.
- C17 : X3, Y3, Z3 - Rewound coordinates of the scaled, offset, twisted, and resampled input airfoils.
- C18A : XU1, YU1 - Matrices containing the X and Y coordinates, respectively, of the upper surfaces of the scaled, offset, and twisted input airfoils.
- C18B : XL1, YL1 - Matrices containing the X and Y coordinates, respectively, of the lower surfaces of the scaled, offset, and twisted input airfoils.
- C20 : U1, U2 - Matrices containing the values of X and Y, respectively, as functions of ξ and η .
- C22 : XZ, YZ - Vectors containing the X and Y values as interpolated by the surface tension spline at the current η station. XZ is computed in GETX and YZ in GETY.
- C25 : SLE - Vector containing the polygonal arc lengths to each airfoil's leading edge.
TAU - Refinement parameter.
- C30 : NOW - Number of rows of data in the chord/twist/thickness distribution file.
ROR - Z/span coordinate of the airfoil.

COR - Chord/span ratio of the airfoil.

TWI - Twist of the airfoil.

THK - Maximum thickness-to-chord ratio of the airfoil.

C31 : SIGMACH, SIGMATW, SIGMATH - Tension parameters needed for the one-dimensional interpolation of the chord, twist, and thickness distributions at the required span stations of the blade frame.

TRIDM : CL, CM, CN - Vectors containing lower, leading, and upper diagonal coefficients, respectively, of a tridiagonal matrix.

PQ - Vector that originally contains the right-hand-side values of the tridiagonal system, and that, after solution of the system, contains the solution.

AIJKIJ : AK - A 4-by-4 matrix that originally contains the function values of u and its partial derivatives for all four nodes of a sector; see equation (2.21). After manipulations, contains the spline coefficients for the particular sector concerned, see equation (2.19).

CHZXY : CHZX, CHZY - Two 4-by-4 coefficient matrices that contain values necessary in computing the spline coefficients for each sector (see equation [2.19]).

PHIXY : PHIX, PHIY - Vectors of length 4 that contain interpolation function values in the ξ and η directions, respectively (see equation [2.3]).

CHAPTER 6

HOW TO RUN THE CODE

Included in this chapter are the descriptions of

- (1) How to install the code
- (2) The necessary input data files
- (3) Options for running the program
- (4) A sample run of the program.

6.1 Program Installation

Before the code can be executed, the following must be done:

- (1) Make sure that the F77L¹ software is loaded on the system.
- (2) Make sure that the GRAFMATIC, PLOTMATIC, and PRINTMATIC² software libraries are loaded on the system.
- (3) Write down the paths to, and the directories into which, the F77L, GRAFMATIC, PLOTMATIC, and PRINTMATIC libraries have been loaded.
- (4) Make a directory called BLADE (or any other name), and make this directory the current directory, for example:

MD \BLADE

CD \BLADE

¹ F77L is the trademark of Lahey Computer Systems, Inc. For those who do not have the Lahey FORTRAN 77 compiler, all the "CALL SYSTEM (comnd)" statements in the BLADE source code must be modified. The SYSTEM subroutine is a F77L special user-callable system subroutine provided as an extension to the FORTRAN language. This subroutine passes a character expression "comnd" to DOS to be executed as if it had been typed at the console.

² See the footnote on page 37 for detail.

- (5) Copy all the source files from the BLADE diskette into the BLADE directory, for example:

```
COPY A:*. * \BLADE\*. *
```

- (6) Compile and link all the source files in the BLADE directory by typing the following at the DOS prompt:

```
COMPILE_PD1_PD2_PD3_PD4_PD5_PD6
```

where

PD1 is the path and directory containing the F77L execution and library files (F77L.EXE and F77L.LIB), e.g., C:\F77L\

PD2 is the path and directory containing the GRAFMATIC library for Lahey FORTRAN (GRAFEXLY.LIB), e.g., C:\GRAFMATIC\

PD3 is the path and directory containing the GRAFMATIC screen font library (QFONTLY.LIB), e.g., C:\GRAFMATIC\

PD4 is the path and directory containing the PLOTMATIC library for Lahey FORTRAN (PLOTLHP.LIB), e.g., C:\PLOTMATIC\

PD5 is the path and directory containing the PLOTMATIC plot font library for HP plotters (HPFONTLY.LIB), e.g., C:\PLOTMATIC\

PD6 is the path and directory containing the PRINTMATIC library (PRINTLY.LIB), e.g., C:\PRINTMATIC\

_ signifies a blank space.

- (7) Set up the system for sending graphics to the screen, plotter, and printer. This is done by typing **SETUP** at the DOS prompt.

NOTE: The set up step is especially important if one has just installed the code on the system, or if one intends to change the terminal, plotter, or printer hardware.

- (8) Make sure that all the necessary data files are present. If not, create them (see Sections 6.2 and 6.3).
- (9) Begin the execution of the Blade code by typing **BLADE** at the DOS prompt.

6.2 Input Data Files

Before the code is executed the user needs to prepare files for airfoil data, chord/twist/thickness distribution, and general input data (optional). Note that all data are read in free format by the program. Therefore, the data files do not have to be formatted in any specific way.

6.2.1 Airfoil Data File

The airfoil data file contains X and Y coordinates of the upper and lower surfaces of an airfoil. All data should already be normalized with respect to the chord length of the airfoil. Data are read from this file using free format.

Following is a line-by-line description of the data contained in the airfoil data file.

Line 1: NU

NU: Number of points on upper surface of airfoil.

Line 2: NL

NL: Number of points on lower surface of airfoil.

Lines 3 to line (2+NU): XU, YU

XU: X-coordinate of upper airfoil surface.

YU: Y-coordinate of upper airfoil surface.

Line (3+NU) to line (2+NU+NL): XL, YL

XL: X-coordinate of lower airfoil surface.

YL: Y-coordinate of lower airfoil surface.

Line (3+NU+NL), i.e., the last line:

Blank

An example airfoil data file, S805A7.DAT, is shown in Figure 15. The user should prepare one airfoil data file for each input airfoil.

Line 1	33		Number of points on upper airfoil surface.
Line 2	33		Number of points on lower airfoil surface.
Line 3	0.00000	0.00000	Lines 3 through 35 are the data of X and Y coordinates of upper airfoil surface.
Line 4	0.00270	0.00855	
Line 5	0.00997	0.01836	
Line 6	0.01992	0.02714	
Line 7	0.02985	0.03395	
Line 8	0.03996	0.03982	
Line 9	0.04990	0.04485	
Line 10	0.05996	0.04940	
Line 11	0.07998	0.05722	
Line 12	0.09994	0.06378	
Line 13	0.11995	0.06938	
Line 14	0.14996	0.07627	
Line 15	0.19998	0.08417	
Line 16	0.25000	0.08840	
Line 17	0.30000	0.09019	
Line 18	0.35001	0.09027	
Line 19	0.40001	0.08904	
Line 20	0.45001	0.08658	
Line 21	0.50002	0.08266	
Line 22	0.55001	0.07706	
Line 23	0.60001	0.07036	
Line 24	0.65000	0.06326	
Line 25	0.70000	0.05594	
Line 26	0.75001	0.04843	
Line 27	0.80001	0.04069	
Line 28	0.85001	0.03255	
Line 29	0.90001	0.02351	
Line 30	0.93001	0.01705	
Line 31	0.95000	0.01215	
Line 32	0.97000	0.00700	
Line 33	0.97999	0.00445	
Line 34	0.98999	0.00206	
Line 35	1.00000	0.00000	

Note: This data file is continued on the next page.

Figure 15 Sample Input Airfoil Data File (Example of S805A7.DAT)

Line 36	0.00000	0.00000	Lines 36 through 68 are the data of X and Y coordinates of lower airfoil surface.
Line 37	0.00297	-0.00772	
Line 38	0.01003	-0.01375	
Line 39	0.01997	-0.01916	
Line 40	0.02996	-0.02361	
Line 41	0.03994	-0.02727	
Line 42	0.04998	-0.03046	
Line 43	0.05996	-0.03333	
Line 44	0.07997	-0.03838	
Line 45	0.10000	-0.04268	
Line 46	0.11998	-0.04642	
Line 47	0.14997	-0.05120	
Line 48	0.20000	-0.05739	
Line 49	0.25000	-0.06166	
Line 50	0.30000	-0.06413	
Line 51	0.35001	-0.06455	
Line 52	0.40002	-0.06282	
Line 53	0.45002	-0.05920	
Line 54	0.50001	-0.05415	
Line 55	0.55000	-0.04826	
Line 56	0.60000	-0.04193	
Line 57	0.65000	-0.03531	
Line 58	0.70000	-0.02845	
Line 59	0.74999	-0.02130	
Line 60	0.79999	-0.01389	
Line 61	0.84998	-0.00679	
Line 62	0.89999	-0.00117	
Line 63	0.92999	0.00109	
Line 64	0.95000	0.00193	
Line 65	0.97000	0.00202	
Line 66	0.98001	0.00169	
Line 67	0.99000	0.00103	
Line 68	1.00000	0.00000	
Line 69			The last line should be blank.

Figure 15
(concluded)

Sample Input Airfoil Data File (Example of S805A7.DAT)

6.2.2 Chord/Twist/Thickness Distribution Data File

This data file is read in free format and contains data that define the chord, twist, and maximum thickness/chord ratio of an airfoil at specific span stations. The chord should be normalized with respect to the span of the blade.

Following is a line-by-line description of the data contained in the chord/twist/thickness distribution file.

Line 1: NOW

NOW: Number of span stations defining chord/twist/thickness distribution curve.

Line 2 to line (1+NOW): ROR, COR, TWI, THKN

ROR: The Z coordinate of the airfoil at the span station, normalized with respect to span.

COR: The chord of the airfoil at the span station, normalized with respect to span.

TWI: The twist angle of the airfoil at the span station.

THKN: The maximum thickness/chord ratio of the airfoil at the span station.

Line (2+NOW), i.e., the last line:

Blank

An example chord/twist/thickness distribution file, CTE2.DAT, is shown in Figure 16. At least one chord/twist/thickness distribution data file must be on hand before execution of the program, if the user intends to use a specific distribution to generate his/her own blade surface.

Line 1	16				Number of span stations representing the distributions.
Line 2	0.05	0.225	33.5	0.220	Lines 2 through 17 contain values of span coordinates (first column), chord at the current span station (second column), the twist angle in degrees at the current span station (third column), and the maximum thickness/chord ratio at the current span station (fourth column).
Line 3	0.10	0.210	26.8	0.210	
Line 4	0.15	0.188	21.0	0.200	
Line 5	0.20	0.167	15.2	0.190	
Line 6	0.30	0.133	7.80	0.180	
Line 7	0.35	0.118	5.00	0.178	
Line 8	0.45	0.092	1.00	0.172	
Line 9	0.50	0.080	0.00	0.168	
Line 10	0.55	0.072	-0.7	0.162	
Line 11	0.65	0.065	-1.0	0.150	
Line 12	0.70	0.065	-1.0	0.143	
Line 13	0.75	0.065	-1.0	0.135	
Line 14	0.80	0.065	-1.0	0.125	
Line 15	0.85	0.065	-1.0	0.115	
Line 16	0.90	0.065	-1.0	0.101	
Line 17	0.95	0.065	-1.0	0.088	
Line 18					The last line should be blank.

Figure 16 Sample Chord/Twist/Thickness Distribution Data File (Example of CTE2.DAT)

6.3 Running the Program

The code can accept input either from the terminal or from data files. Following are descriptions of the input and output options a user will encounter during the execution of the code.

6.3.1 Input Options

It is advisable for beginning users to enter input at the terminal, in order to acquaint themselves with the code. However, a tedious amount of data are requested, especially if many input airfoils are being used. For this reason the user is provided an option for entering data by use of an input data file. Most of the input data needed during the execution of the code are requested by subroutine INPUT.

When the program is executing, the user is first introduced to the package. After that, the user is asked for the way of input data as following:

The computer: You have two alternative ways of entering data:

Enter T for terminal input
Enter D for data file input

Enter your choice here ----->

The user's response:

T
or
D: If and only if an input data File
has been created beforehand.

If the user's response was D then:

The computer: Please enter the name of the data input file --->

The user's response: For example: JAYIN.DAT

The remaining data required for the blade surface generation are either input at the terminal or from the data file specified in the step above. An example input data file, JAYIN.DAT, is shown in Figure 17. Whether input is from the terminal or from an input data file, it is read sequentially using free format.

For simplicity, and yet in order to fully define the remaining of the input data requested during execution of the code, a step-by-step description will now be adhered to. This step-by-step description applies to both the terminal input option and the data file input option.

Step 1

(Input data requested by subroutine INPUT.)

Line 1: IDC

IDC: the desired choice of chord/twist/thickness distribution along the span of the blade.

Value of IDC is:

1. if self generated.
2. if example 1 is used.
3. if example 2 is used.
4. if distribution is from a data file supplied by the user.

If and only if IDC = 4, then enter line 2.

Line 2: DNAME

DNAME: the name of the data file containing the desired chord/twist/thickness distribution data.

Step 2

(Input data, necessary for creating the blade frame, requested by subroutine INPUT.)

Line 1: NAF

Line 1	3	An integer that indicates the method of input of the chord/twist/thickness distribution data. (In this case data from a data file, are being used)
Line 2	9	Number of input airfoils.
Line 3	30.5	Rotor radius, or span.
Line 4	Y	Yes, scale the input airfoils.
Line 5	6	Lines 5 through 9 respectively, represent (1) the desired type of input airfoil, (2) Z coordinate of the span station where this airfoil will be located, (3) the desired choice of specifying twist center X and Y coordinates, (4) the twist center X-coordinate, and (5) the twist center Y-coordinate. This is repeated nine times up to line 49, since the number of input airfoils is equal to nine.
Line 6	4.574	
Line 7	1	
Line 8	0.25	
Line 9	0.	
Line 10	6	
Line 11	7.625	
Line 12	1	
Line 13	0.25	
Line 14	0.	
Line 15	5	
Line 16	9.15	
Line 17	1	
Line 18	0.25	
Line 19	0.	
Line 20	5	
Line 21	13.75	
Line 22	1	
Line 23	0.25	
Line 24	0.	
Line 25	4	
Line 26	16.775	
Line 27	1	
Line 28	0.25	
Line 29	0.	
Line 30	4	
Line 31	19.825	
Line 32	1	
Line 33	0.25	
Line 34	0.	

Note: This data file is continued on the next page.

Figure 17 Sample Input Data File (Example of JAYIN.DAT)

Line 35	3	
Line 36	22.875	
Line 37	1	
Line 38	0.25	
Line 39	0.	
Line 40	2	
Line 41	25.925	
Line 42	1	
Line 43	0.25	
Line 44	0.	
Line 45	1	
Line 46	28.975	
Line 47	1	
Line 48	0.25	
Line 49	0.	
Line 50	119	The desired number of points to define the airfoil.
Line 51	1.	Tension factor in ξ direction.
Line 52	1.	Tension factor in η direction.
Line 53		Last line should be blank.

Figure 17
(concluded)

Sample Input Data File (Example of JAYIN.DAT)

NAF: the number of airfoil data sets to be input and interpolated in order to generate the blade surface.
Note: $1 < \text{NAF} < 11$

Line 2: SPAN

SPAN: the radius of rotor blade to be generated.

Line 3: SCALE

SCALE: A character specifying whether the scaling of the input airfoils is to be done.

The character of SCALE is: N for no scaling
Y for scaling

Step 3

(Input data, necessary for creating the blade frame, requested by subroutine INPUT. Note that this step is repeated sequentially NAF times.)

Line 1: NDF(K)

NDF(K): the desired airfoil type at station K.

Value of NDF(K) is:

1. for the S806A airfoil.
2. for the S805A/6A airfoil.
3. for the S805A airfoil.
4. for the S805A/7A airfoil.
5. for the S807 airfoil.
6. for the S808 airfoil.
7. for own entry of airfoil.

If and only if $\text{NDF}(K) = 7$, then enter line 2.

Line 2: NAME(K)

NAME(K): the name of the data file containing normalized coordinate data of desired airfoil at station K.

Line 3: ZZ(K)

ZZ(K): the Z coordinate at span station K.

If and only if $\text{IDC} = 1$, then enter lines 4, 5, and 6.

Line 4: TH1(K)

TH1(K): the twist angle, in degrees, of airfoil at station K.

Line 5: CL(K)

CL(K): the chord length of the airfoil at station K.

Line 6: THKN(K)

THKN(K): the maximum thickness/chord ratio of the airfoil at station K.

Line 7: TWCTR

TWCTR: the desired choice of specifying twist center X and Y coordinates.

Value of TWCTR is

1. to be input by the user.
2. to be determined by the program.

NOTE: Option 2 will place the twist center at the intersection of $X = 1/3$ and the airfoil meanline.

If and only if TWCTR = 1, then enter lines 8 and 9.

Line 8: OSX(K)

OSX(K): the X-coordinate of the twist center.

Line 9: OSY(K)

OSY(K): the Y-coordinate of the twist center.

Step 4

(Input data, necessary for the tension spline interpolation, as requested by subroutine INPUT.)

Line 1: N1

N1: the desired number of points to define airfoil. The value entered must be an odd integer number.

Line 2: ALPHA

ALPHA: the interpolation tension factor in the ξ direction. The standard value is 1.

Line 3: BETA

BETA: the interpolation tension factor in the η direction. The standard value is 1.

Step 5

(Stretching parameter requested by subroutine TRANS for concentrating data points about leading edge.)

Line 1: TAU

TAU: the stretching parameter used for concentrating the data points about the leading edge of the airfoil. The standard value is between 4.0 and 6.0, depending upon how many points (N1) are used to define the airfoil. Note: This parameter must be input at the terminal.

After step 4 of input, the program creates the frame of the blade according to the method of Chapter 3. On completion of the blade frame creation, the transformation module is entered. Step 5 reads the data needed to perform the transformations. There is no more required input, and after the program has performed the transformation (according to the methodology of Chapter 4), control passes to the interpolation module.

6.3.2 Output Options

In the interpolation module, the code gives the user two interpolation options:

- (1) Generating an airfoil at any user-prescribed station along the span of the blade.
- (2) Generating a three-dimensional blade, represented by a user-specified number of airfoils placed equally spaced along the span of the blade.

The output that the user can view on the screen, plot, and print is dependent upon which interpolation option the user selects.

Option 1 will furnish the user with

- (a) A concentric plot of the input airfoils forming the blade frame (normalized with respect to the span of the blade).
- (b) A span-normalized plot of the airfoil at the prescribed span station.
- (c) A chord-normalized plot of the airfoil at the prescribed span station, with the airfoil's twist and offset removed.

- (d) A printout of tables summarizing the input and generated airfoil data.

Option 2 will furnish the user with

- (a) A concentric plot of the input airfoils forming the blade frame (normalized with respect to the span of the blade).
- (b) A concentric plot of the user-specified number of airfoils forming the generated blade (normalized with respect to the span of the blade).
- (c) A perspective view of the span-normalized generated blade.
- (d) A top view of the span-normalized generated blade.
- (e) A front view of the span-normalized generated blade.
- (f) A plot comparing the chord distribution of the airfoils along the span of the generated blade with that along the span of the input data blade frame.
- (g) A plot comparing the twist distribution of the airfoils along the span of the generated blade with that along the span of the input data blade frame.
- (h) A plot comparing the maximum thickness/chord ratio distribution of the airfoils along the span of the generated blade with that along the span of the input data blade frame.
- (i) A printout of tables summarizing the input and generated airfoil data.

For each plot, the user is given the option of viewing the plot on the screen. The user is also offered hard-copy options, in which they can (1) turn down the offer; (2) send a hard-copy of the plot to the plotter; or (3) send a hard-copy of the plot to the printer.

The BLADE code then gives the user the option to interpolate the same input airfoils again. If so, the user is returned to the interpolation options menu. If not, the program asks whether the user wants to interpolate a new set of input airfoils.

An affirmative answer will loop the program back to the INPUT module, where the whole sequence of interaction described in Section 6.3 is re-initiated. A negative reply will stop the execution of the BLADE code and cause the program to exit to DOS.

6.4 Sample Run of the Program

A sample run of the program using the INPUT DATA option 1 (see Section 6.3.1) is now given below. In this example, three airfoils (S807, S805A, and S806A) were used as input to form a 60-meter-long blade. The airfoils and their respective Z-coordinates, twist angles, chord lengths, maximum thickness/chord ratios, and twist center X and Y coordinates, are shown in Figure 18.

SPAN STATION NUMBER	AIRFOIL TYPE	Z-Coord in meters	TWIST in degrees	CHORD in meters	MAX Th/Ch RATIO	TWIST CENTER (X,Y)
1	S807	0.0	25.0	6	.25	(0.25,0.0)
2	S805A	20.0	10.0	5	.2	(0.25,0.0)
3	S806A	60.0	0.0	2	.15	(0.25,0.0)

Figure 18 Data Used to Make the Sample Run

*Here is the sample run of the program. The numbers in **boldface and underline** are the input values.*

This program uses certain cross-sectional airfoil data as input, and proceeds to interpolate these input data by use of a bi-tension spline method.

As a result of this interpolation, the whole surface of the blade is definable.

This program will then generate airfoil data for any particular span stations of the blade.

*****READ AND FOLLOW INSTRUCTIONS CAREFULLY*****

Press Enter to Continue.

You have two alternative ways of entering data :

Enter T for terminal input

Enter D for data file input

Enter your choice here ---> **T**

Since this is a free nation , you are given a choice of how you would like to generate your blade.

Following are your options:

1.....GENERATE YOUR OWN TWISTED AND TAPERED BLADE.
(i.e., The user will input the chord/twist/thickness distribution at the terminal.)

2.....USE Example 1 chord/twist/thickness distribution.

3.....USE Example 2 chord/twist/thickness distribution.

4.....USE A chord/twist/thickness distribution that is not listed here.

NOTE: Option 4 can only be used if you have already created a chord/twist/thickness distribution data file.
(Please refer to the user's manual on how to generate this data file.)

Enter your choice here ---> **1**

How many airfoils do you want to input to form the blade?

*****Enter a number between 2 and 10 *****

---> **3**

Enter the span of the desired blade (i.e., the radius)

---> **60.0**

Do you want to scale the input airfoils ?

Enter : N for NO
 Y for YES

Enter your choice here ---> **Y**

Below are airfoil types to choose from. Data are contained in data files. They are already normalized with respect to chord.

If you do not desire any of the stated airfoil types, you can choose option 7.

- 1.....S806A Tangler Somers thin airfoil
- 2.....S805A/6A " " " "
- 3.....S805A " " " "
- 4.....S805A/7A " " " "
- 5.....S807 " " " "
- 6.....S808 " " " "
- 7....None of the above. I will enter my own.

Enter desired airfoil number for position 1 here ---> **5**

Reading data from data file S807.DAT

Data from file S807.DAT for airfoil 1 have been read

Press Enter to Continue.

The Z coordinate must be between 0.00 and 60.0

Enter Z coordinate for airfoil 1 here ---> **0.0**

Enter the twist angle (in degrees) ---> **25.0**

Enter the chord length of airfoil 1 ---> **6.0**

Enter the maximum thickness/chord ratio.

The value should be a decimal less than 1.0. ---> **0.25**

Enter the twist center X- and Y-coordinates.

Enter 1 if you want to input by yourself

Enter 2 if you want to determine by the program

NOTE: Option 2 only applies to the case where the twist center is located at the intersection of $X = 1/3$ and the meanline.

PLEASE ENTER YOUR CHOICE NUMBER HERE ---> **1**

Enter X-coordinate here ---> **0.25**

Enter Y-coordinate here ---> **0.0**

Please wait, scaling program is running!

Below are airfoil types to choose from. Data are contained in data files. They are already normalized with respect to chord.

If you do not desire any of the stated airfoil types, you can choose option 7.

1.....S806A Tangler Somers thin airfoil

2.....S805A/6A " " " "

3.....S805A " " " "

4.....S805A/7A " " " "

5.....S807 " " " "

6.....S808 " " " "

7....None of the above. I will enter my own.

Enter desired airfoil number for position 2 here ---> **3**

Reading data from data file S805A.DAT

Data from file S805A.DAT for airfoil 2 have been read

Press Enter to Continue.

The Z coordinate must be between 0.00 and 60.0

Enter Z coordinate for airfoil 2 here ---> **20.0**

Enter the twist angle (in degrees) ---> **10.0**

Enter the chord length of airfoil 2 ---> **5.0**

Enter the maximum thickness/chord ratio.
The value should be a decimal less than 1.0, ---> 0.2

Enter the twist center X- and Y-coordinates.

Enter 1 if you want to input by yourself

Enter 2 if you want to determine by the program

NOTE: Option 2 only applies to the case where the
twist center is located at the intersection
of $X = 1/3$ and the meanline.

PLEASE ENTER YOUR CHOICE NUMBER HERE ---> 1

Enter X-coordinate here ---> 0.25

Enter Y-coordinate here ---> 0.0

Please wait, scaling program is running!

Below are airfoil types to choose from. Data are
contained in data files. They are already normalized
with respect to chord.

If you do not desire any of the stated airfoil types,
you can choose option 7:

1.....S806A Tangler Somers thin airfoil

2.....S805A/6A " " " "

3.....S805A " " " "

4.....S805A/7A " " " "

5.....S807 " " " "

6.....S808 " " " "

7....None of the above. I will enter my own.

Enter desired airfoil number for position 3 here ---> 1

Reading data from data file S806A.DAT

Data from file S806A.DAT for airfoil 3 have been read

Press Enter to Continue.

The Z coordinate must be between 20.00 and 60.0

Enter Z coordinate for airfoil 3 here ---> 60.0

Enter the twist angle (in degrees) ---> 0.0

Enter the chord length of airfoil 3 ---> 2.0

Enter the maximum thickness/chord ratio.
The value should be a decimal less than 1.0. ---> 0.15

Enter the twist center X- and Y-coordinates.

Enter 1 if you want to input by yourself

Enter 2 if you want to determine by the program

PLEASE ENTER YOUR CHOICE NUMBER HERE ---> 1

Enter X-coordinate here ---> 0.25

Enter Y-coordinate here ---> 0.0

Please wait, scaling program is running!

ENTER the number of points you want per airfoil:

An ODD integer number between 50 and 120

==> 61

WE NOW NEED INPUT FOR BLADE SURFACE INTERPOLATION.

Enter the tension factors in X and Z directions,

respectively.

The numbers must be greater than 0 (zero)
and less than 1000.

ENTER X TENSION FACTOR (standard value is 1) ---> 1.0

ENTER Z TENSION FACTOR (standard value is 1) ---> 1.0

PLEASE WAIT.....

CALCULATIONS ARE TAKING PLACE

ENTER THE STRETCHING PARAMETER FOR CONCENTRATING
THE AIRFOIL DATA POINTS ABOUT THE LEADING EDGE.

THE STANDARD VALUE RANGES BETWEEN 4.0 AND 6.0
THE FEWER POINTS YOU USED TO REPRESENT THE
AIRFOIL CROSS SECTION, THE SMALLER
VALUE YOU SHOULD USE.

If no stretching is desired, enter 0.0001

Enter your choice here ---> 4.0

PLEASE WAIT.....

CALCULATIONS IN PROGRESS.....

Beyond this point are the graphic output options, which the user can choose interactively at the terminal.

This now follows on the next page.

YOU ARE NOW GIVEN THE FOLLOWING OPTIONS :
1.....GENERATE AN AIRFOIL AT ANY SPAN STATION.
2.....GENERATE A THREE-DIMENSIONAL BLADE.

NOTE :

Option 1 will furnish the user with an airfoil profile at the span station desired, as well as an airfoil normalized w.r.t. chord

Option 2 will interpolate over the whole blade and generate the coordinates for any specified number of span station airfoils.

Output plots you can create using option 2 are

- a) Concentric plots of generated airfoils.
- b) Perspective view of the generated blade.
- c) Top view of the generated blade.
- d) Front view of the generated blade.
- e) Chord distribution along the span of the blade
- f) Twist distribution along the span of the blade
- g) Maximum thickness distribution along the span of blade.

Enter your choice number here ---> **2**

You are now given the following three options on how to specify the output X-coordinates of the generated airfoil, which has its twist and offset removed, at the chosen span station.

Option 1....	The X-coordinates will be specified such that there are more points around the leading edge. This feature depends on the stretching parameter previously used.
Option 2....	The X-coordinates will be specified such that they correspond to the STANDARD 57 X-coordinates; see manual.
Option 3....	The X-coordinates will be specified such that they are regularly spaced. The spacing depends on the number of points (to be input next) used to label the X-coordinates.

Enter your choice number here ---> **1**

HOW MANY span stations do you want your blade
represented by ?
ENTER a number greater than 3
(less than 40 for viewing 3D plot) ---> **15**

PLEASE WAIT.....

CALCULATIONS UNDER WAY

The standard deviations of
Chord = 0.189267E+00
Twist angle = 0.815737E+00
Thickness = 0.621089E-01
Press Enter to Continue.

CONCENTRIC PLOT OF THE INPUT AIRFOILS.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> **1**

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> **1**

PLOTTING.....

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> **0**

A....Concentric plot of generated span station airfoils.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> **1**

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> 0

B....Normalized top view plot of the generated blade.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> 1

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> 0

C....Normalized front view plot of the generated blade.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> 1

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> 0

D....The CHORD distribution of the generated blade.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> 1

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> **0**

E....The TWIST distribution of the generated blade.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> **1**

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> **0**

F....The MAXIMUM THICKNESS distribution of the
 generated blade.
WOULD YOU LIKE TO VIEW THIS ?

ENTER ---> 0 NO.
 1 YES. ---> **1**

DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?

ENTER ---> 0 NO.
 1 YES, SEND TO PLOTTER.
 2 YES, SEND TO PRINTER.

ENTER YOUR CHOICE NUMBER HERE ---> **0**

Would you like a printout of the input and/or output data ?
If you want a printout, you need to have plenty of paper and
time, because the printout is several pages long.

ENTER ---> 0 NO.
 1 YES.

ENTER YOUR CHOICE NUMBER HERE ---> **1**

Which data set would you like to have a printout?

1. Input data.
2. Output of the generated airfoil at the chosen span stations with twist and taper.
3. Output of the generated airfoil at the chosen span stations without twist and taper.

Enter your choice number here ---> **2**

MAKE SURE THAT THE PRINTER IS READY.

Press Enter to Continue.

Tabular data output from each run of the code are found in the data files called INPUT.DAT, WTOUT.DAT, and NTOUT.DAT. The INPUT.DAT file contains most of the input data information. The WTOUT.DAT file contains the generated airfoil data with twist and taper at the chosen span stations. The generated airfoil data with twist and offset removed at the chosen span stations are given in the NTOUT.DAT file.

CHAPTER 7

RESULTS AND DISCUSSIONS

The computer code, as described in the previous chapter, generates output consisting of

- (1) Concentric plots of (a) the input airfoils forming the blade frame and (b) the generated airfoils for any number of span stations to define the blade.
- (2) Three-dimensional plot of the generated blade.
- (3) Top and front views of the generated blade.
- (4) Plots of the generated and the input twist, chord, and maximum thickness distributions of the airfoils along the span of the blade.
- (5) Chord-normalized and span-normalized plots of the profile of an airfoil at any span station of the generated blade.
- (6) Tabulated data for any span station of the generated blade, consisting of (a) the upper and lower surface coordinates of the chord-normalized and span-normalized airfoils, in standard airfoil data format and (b) the twist, chord-to-span ratio, maximum thickness/chord ratio, and twist center of the respective airfoil.

To demonstrate the feasibility and the usefulness of the computer code developed, the results of generating two wind turbine blade profiles are presented. The two examples are the CARTER and MICON wind turbine blades. For each example, the numerical results are presented in graphic form along with one set of generated airfoil data.

7.1 CARTER Blade

This example is an attempt to generate the profile of a twisted and tapered blade with a span of 30.5 ft. Nine airfoils were used to form the frame of the blade. The types of airfoils used and the Z-coordinates along the span of the blade at which they were placed are shown in Table 1.

Contained in Table 2 are the data that were used as input to the code to represent the chord, twist, and thickness distributions of the airfoils along the span of the blade. These data are used as basis for a one-dimensional tension spline interpolation to determine the chord, twist, and maximum thickness of each airfoil on the blade frame stated in Table 1.

Figure 19 shows a concentric plot of the scaled and twisted input airfoils forming the CARTER blade frame. The "R" in the axis labels of Figure 19 and all subsequent figures represents the span of the blade. Figure 20 shows a concentric plot of airfoils that were generated by applying the surface tension spline on the transformed blade frame. The three-dimensional plot of the generated blade is shown in Figure 21. Figures 22 and 23 present the top and front views of the generated blade. The vertical lines represent the locations of the generated airfoils. The surface of the generated blade appears smooth in the three-dimensional plot, as well as in both the top and front views.

Figure 24 compares the twist distribution along the span of the generated CARTER blade and that along the span of the desired blade. The generated twist distribution, like the desired one, varies between 33 deg at the base of the blade to -1 deg at the tip of the blade.

Table 1 Input Data to Form the CARTER Blade Frame

Span Station Number	Airfoil Type	Z-Coordinate in Feet	Z/Span Coordinate
1.	S808	4.575	0.15
2.	S808	7.625	0.25
3.	S807	9.150	0.30
4.	S807	13.750	0.45
5.	S805A/7A	16.775	0.55
6.	S805A/7A	19.825	0.65
7.	S805A	22.875	0.75
8.	S805A/6A	25.925	0.85
9.	S806A	28.975	0.95

Table 2 Input Chord/Twist/Thickness Data for the CARTER Blade

Z/Span Coordinate	Chord/Span Ratio	Twist Angle in Degrees	Maximum Thickness/ Chord Ratio
.05	.225	33.5	0.22
.10	.210	26.8	0.21
.15	.188	21.0	0.20
.20	.167	15.2	0.19
.30	.133	7.8	0.18
.35	.118	5.0	0.178
.45	.092	1.0	0.172
.50	.08	0.0	0.168
.55	.072	-0.7	0.162
.65	.065	-1.0	0.150
.70	.065	-1.0	0.143
.75	.065	-1.0	0.135
.80	.065	-1.0	0.125
.85	.065	-1.0	0.115
.90	.065	-1.0	0.101
.95	.065	-1.0	0.088

CONCENTRIC PLOT OF INPUT AIRFOILS

Figure 19 Concentric Plot of the Input Airfoils that Form the CARTER Blade Frame

CONCENTRIC PLOT OF GENERATED AIRFOILS

Figure 20 Concentric Plot of the Airfoils Forming the Generated CARTER Blade

Figure 21 Perspective View of the Generated CARTER Blade

Figure 22 Top View of the Generated CARTER Blade

Figure 23 Front View of the Generated CARTER Blade

TWIST DISTRIBUTION

Figure 24 Twist Distribution Along the Span of the CARTER Blade

In Figure 25 the distribution of the chord-to-span ratio along the span of the generated blade is compared to that along the span of the desired blade. The chord-to-span ratio of the airfoil decreases along the span of the blade from root to tip. This taper of the chord is shown in the top view of the blade (Figure 22).

In Figure 26 the distribution of the maximum thickness/chord ratio of the generated blade is compared to that of the desired blade. Like the chord-to-span ratio, the maximum thickness/chord ratio of the airfoils gradually decreases from the root to the tip of the blade. The taper of the blade due to the decrease in maximum thickness of the airfoil is illustrated in the front view of the blade (Figure 23).

The twist, chord, and maximum thickness distributions obtained show a close match between the characteristics of the generated CARTER blade and those of the desired blade. Different distribution characteristics are obtained when the tension parameters of the surface interpolating tension spline are varied. By appropriately varying the tension parameters, generated characteristics that more closely match the desired characteristics can be obtained.

Figures 27-29 represent span-normalized plots of airfoils at Z/span coordinates of 0.20, 0.55, and 0.90, respectively on the span of the generated CARTER blade. As an example of the data output by the code, the actual surface coordinate data of the airfoil with a Z/span coordinate of 0.55 are presented in Figure 30. Figures 31-33 represent chord-normalized plots of the airfoils in Figures 27-29, but without the twist and offset. Because of the removal of the twist and offset, the surface coordinate data of these airfoils can be presented in a form similar to that of standard airfoil data. As an example of this, in Figure 34, the data representing the surface coordinates of the airfoil with a Z/span coordinate of 0.55 are presented in standard form. Any number of airfoils at any span station of the CARTER blade can be generated and have their profiles and data displayed by the code in the manner of Figures 20-34.

CHORD DISTRIBUTION

Figure 25 Chord Distribution Along the Span of the CARTER Blade

THICKNESS DISTRIBUTION

Figure 26 Maximum Thickness Distribution Along the Span of the CARTER Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.200**

Figure 27 Span-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated CARTER Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.550**

Figure 28 Span-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.900**

Figure 29 Span-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated CARTER Blade

These are the actual coordinates of the airfoil
at span station 16.7750

Chord- Length	Twist angle	WITH		Z/R Coord.	Span R
		Twist Center X	Y		
0.0720	-0.6995	0.5490	0.0000	0.5500	30.5000

The upper and lower airfoil coordinates shown below have been
normalized w.r.t. span, R.

XU	YU	XL	YL
-0.01800	0.00022	-0.01800	0.00022
-0.01745	0.00147	-0.01716	-0.00086
-0.01640	0.00239	-0.01593	-0.00152
-0.01521	0.00314	-0.01463	-0.00204
-0.01393	0.00377	-0.01328	-0.00247
-0.01259	0.00431	-0.01188	-0.00285
-0.01119	0.00479	-0.01044	-0.00319
-0.00973	0.00521	-0.00896	-0.00349
-0.00820	0.00558	-0.00742	-0.00378
-0.00661	0.00590	-0.00583	-0.00403
-0.00495	0.00616	-0.00417	-0.00426
-0.00321	0.00637	-0.00244	-0.00447
-0.00138	0.00652	-0.00064	-0.00465
0.00053	0.00662	0.00126	-0.00480
0.00255	0.00667	0.00325	-0.00492
0.00466	0.00668	0.00534	-0.00499
0.00689	0.00664	0.00755	-0.00501
0.00925	0.00655	0.00988	-0.00496
0.01173	0.00641	0.01233	-0.00484
0.01436	0.00621	0.01493	-0.00464
0.01715	0.00593	0.01767	-0.00438
0.02010	0.00558	0.02057	-0.00405
0.02323	0.00515	0.02365	-0.00367
0.02655	0.00466	0.02692	-0.00323
0.03009	0.00410	0.03039	-0.00273
0.03385	0.00348	0.03408	-0.00219
0.03786	0.00280	0.03801	-0.00164
0.04213	0.00203	0.04220	-0.00110
0.04667	0.00114	0.04668	-0.00067
0.05400	-0.00066	0.05400	-0.00066

Figure 30 Span-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on
the Generated CARTER Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.200**

Figure 31 Chord-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated CARTER Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.550**

Figure 32 Chord-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.900**

**Figure 33 Chord-Normalized Airfoil at a Z/span Coordinate of 0.90 on the
Generated CARTER Blade**

Chord- Length	Twist angle	Thickness	Twist Center		Z/R Coord.	Span R
0.0720	-0.6995	0.1620	X 0.2500	Y 0.0000	0.5500	30.5000

The following upper and lower airfoil coordinates are for the airfoil, with no twist, no offset, and are normalized w.r.t. chord = 2.196000

XU	YU	XL	YL
0.00000	0.00000	0.00000	0.00000
0.00050	0.00216	0.00050	-0.00105
0.00100	0.00418	0.00100	-0.00206
0.00200	0.00778	0.00200	-0.00398
0.00500	0.01461	0.00500	-0.00879
0.01000	0.01996	0.01000	-0.01363
0.02000	0.02894	0.02000	-0.01952
0.03000	0.03601	0.03000	-0.02429
0.04000	0.04198	0.04000	-0.02825
0.05000	0.04718	0.05000	-0.03167
0.06000	0.05183	0.06000	-0.03477
0.07000	0.05603	0.07000	-0.03761
0.08000	0.05986	0.08000	-0.04022
0.09000	0.06337	0.09000	-0.04263
0.10000	0.06660	0.10000	-0.04487
0.15000	0.07940	0.15000	-0.05408
0.20000	0.08744	0.20000	-0.06073
0.25000	0.09170	0.25000	-0.06530
0.30000	0.09340	0.30000	-0.06793
0.35000	0.09328	0.35000	-0.06839
0.40000	0.09165	0.40000	-0.06656
0.45000	0.08854	0.45000	-0.06273
0.50000	0.08398	0.50000	-0.05735
0.55000	0.07823	0.55000	-0.05094
0.60000	0.07166	0.60000	-0.04389
0.65000	0.06456	0.65000	-0.03630
0.70000	0.05710	0.70000	-0.02842
0.75000	0.04932	0.75000	-0.02058
0.80000	0.04124	0.80000	-0.01308
0.85000	0.03272	0.85000	-0.00643
0.90000	0.02333	0.90000	-0.00130
0.95000	0.01162	0.95000	0.00142
0.98000	0.00432	0.98000	0.00116
0.99000	0.00212	0.99000	0.00067
1.00000	0.00000	1.00000	0.00000

Figure 34 Chord-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on the Generated CARTER Blade, with the Twist and Offset removed

7.2 MICON Blade

This example is an attempt to generate the profile of another twisted and tapered blade. Ten airfoils were used to form the frame of the 9 meter MICON blade. The types of airfoils and the Z-coordinates along the span of the blade at which they were placed are shown in Table 3.

Input data consisting of the chord, twist, and thickness distributions along the span of the MICON blade are presented in Table 4.

Figure 35 shows a concentric plot of the scaled and twisted input airfoils forming the MICON blade frame. Figure 36 shows a concentric plot of airfoils that were generated by applying the surface tension spline on the transformed blade frame. The three-dimensional plot of the generated blade is shown in Figure 37. Figures 38 and 39 present the top and front views of the generated blade. The surface of the generated blade appears smooth in both the top and front views, as well as in the three-dimensional plot. Figures 40, 41, and 42 compare the twist, chord, and maximum thickness distributions respectively, along the span of the generated blade with those along the span of the desired blade. Again the agreement is excellent.

The generated twist distribution, like the desired one, varies from 16 deg at the base of the blade to 0 deg at the tip of the blade. The chord-to-span ratio of the airfoils decreases in an almost linear manner along the span of the blade from root to tip. This taper of the chord is illustrated well in the top view of the blade (Figure 38). Like the chord-to-span ratio, the maximum thickness/chord ratio of the airfoils gradually decreases from the root to tip of the blade. The taper of the blade due to the decrease in maximum thickness of the airfoils is illustrated in the front view of the blade (Figure 39).

Figures 43-45 represent span-normalized plots of airfoils at Z/span coordinates of 0.20, 0.55, and 0.90, respectively, on the span of the generated MICON blade. The actual surface coordinate data of the airfoil in Figure 44 are presented in Figure 46. Figures 47-49 represent

chord-normalized plots of the airfoils in Figures 33-35, but with the twist and offset removed. The chord-normalized surface coordinates of the airfoil in Figure 48 are presented in Figure 50 in a form similar to that of standard airfoil data. Any number of airfoils at any span station of the generated MICON blade can have their profiles and data displayed in the manner of Figures 43-50.

Table 3 Input Data to Form the MICON Blade Frame

Span Station Number	Airfoil Type	Z-Coordinate in Meters	Z/Span Coordinate
1.	S808	1.35	0.15000
2.	S807	2.7	0.30000
3.	S807	3.5	0.38889
4.	S807	4.0	0.44444
5.	S805A/7A	4.5	0.50000
6.	S805A/7A	5.5	0.61111
7.	S805A	6.75	0.75000
8.	S805A/6A	7.5	0.83333
9.	S806A	8.55	0.95000
10.	S806A	8.9	0.98889

Table 4 Input Chord/Twist/Thickness Data for the MICON Blade

Z/Span Coordinate	Chord/Span Ratio	Twist Angle in Degrees	Maximum Thickness/ Chord Ratio
0.144444	0.132	15.9	0.25
0.233333	0.125	12.7	0.20
0.322222	0.117	10.1	0.171
0.414444	0.109	7.5	0.152
0.505556	0.102	5.6	0.137
0.594444	0.093	3.9	0.127
0.683333	0.085	2.7	0.122
0.772222	0.078	1.8	0.118
0.861111	0.069	0.9	0.118
0.925556	0.063	0.45	0.118
0.988889	0.057	0.0	0.118

CONCENTRIC PLOT OF INPUT AIRFOILS

Figure 35 Concentric Plot of the Input Airfoils that Form the MICON Blade Frame

CONCENTRIC PLOT OF GENERATED AIRFOILS

Figure 36 Concentric Plot of the Airfoils Forming the Generated MICON Blade

Figure 37 **Perspective View of the Generated MICON Blade**

Figure 38 Top View of the Generated MICON Blade

Figure 39 Front View of the Generated MICON Blade

TWIST DISTRIBUTION

Figure 40 Twist Distribution Along the Span of the MICON Blade

CHORD DISTRIBUTION

Figure 41 Chord Distribution Along the Span of the MICON Blade

THICKNESS DISTRIBUTION

Figure 42 Maximum Thickness Distribution Along the Span of the MICON Blade

PLOT OF GENERATED AIRFOIL AT Z/SPAN coordinate = 0.200

Figure 43 Span-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated MICON Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.550**

Figure 44 Span-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.900**

Figure 45 Span-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated MICON Blade

These are the actual coordinates of the airfoil
at span station 4.9500

WITH					
Chord- Length	Twist angle	Twist Center		Z/R Coord.	Span R
		X	Y		
0.0976	4.7321	0.2635	0.1065	0.5500	9.0000

The upper and lower airfoil coordinates shown below have been
normalized w.r.t. span, R.

XU	YU	XL	YL
-0.02820	-0.01421	-0.02820	-0.01421
-0.02782	-0.01311	-0.02741	-0.01514
-0.02611	-0.01149	-0.02519	-0.01581
-0.02410	-0.01023	-0.02286	-0.01622
-0.02303	-0.00968	-0.02167	-0.01637
-0.02080	-0.00869	-0.01925	-0.01662
-0.01964	-0.00824	-0.01801	-0.01671
-0.01723	-0.00740	-0.01547	-0.01686
-0.01598	-0.00702	-0.01417	-0.01691
-0.01337	-0.00630	-0.01147	-0.01698
-0.01201	-0.00597	-0.01008	-0.01700
-0.00917	-0.00537	-0.00719	-0.01701
-0.00769	-0.00510	-0.00569	-0.01699
-0.00458	-0.00461	-0.00257	-0.01692
-0.00295	-0.00439	-0.00094	-0.01686
0.00048	-0.00399	0.00248	-0.01668
0.00228	-0.00381	0.00426	-0.01656
0.00607	-0.00350	0.00801	-0.01623
0.01013	-0.00325	0.01202	-0.01578
0.01450	-0.00309	0.01631	-0.01519
0.01681	-0.00305	0.01856	-0.01484
0.02170	-0.00306	0.02332	-0.01405
0.02697	-0.00318	0.02844	-0.01312
0.03267	-0.00340	0.03395	-0.01205
0.03569	-0.00353	0.03685	-0.01146
0.04211	-0.00384	0.04301	-0.01017
0.04906	-0.00423	0.04968	-0.00877
0.05275	-0.00447	0.05322	-0.00807
0.06059	-0.00511	0.06078	-0.00680
0.06906	-0.00616	0.06906	-0.00616

Figure 46 Span-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on
the Generated MICON Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.200**

Figure 47 Chord-Normalized Airfoil at a Z/span Coordinate of 0.20 on the Generated MICON Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.550**

Figure 48 Chord-Normalized Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade

**PLOT OF GENERATED AIRFOIL
AT Z/SPAN coordinate = 0.900**

Figure 49 Chord-Normalized Airfoil at a Z/span Coordinate of 0.90 on the Generated MICON Blade

Chord- Length	Twist angle	Thickness	Twist Center		Z/R Coord.	Span R
0.0976	4.7321	0.1320	X 0.3000	Y 0.1213	0.5500	9.0000

The following upper and lower airfoil coordinates are for the airfoil, with no twist, no offset, and are normalized w.r.t. chord = 0.878370

XU	YU	XL	YL
0.00000	0.00000	0.00000	0.00000
0.00050	0.00135	0.00050	-0.00088
0.00100	0.00265	0.00100	-0.00173
0.00200	0.00512	0.00200	-0.00337
0.00500	0.01120	0.00500	-0.00762
0.01000	0.01668	0.01000	-0.01203
0.02000	0.02337	0.02000	-0.01516
0.03000	0.02973	0.03000	-0.01912
0.04000	0.03469	0.04000	-0.02191
0.05000	0.03920	0.05000	-0.02433
0.06000	0.04317	0.06000	-0.02658
0.07000	0.04675	0.07000	-0.02861
0.08000	0.05003	0.08000	-0.03049
0.09000	0.05304	0.09000	-0.03221
0.10000	0.05581	0.10000	-0.03382
0.15000	0.06687	0.15000	-0.04047
0.20000	0.07404	0.20000	-0.04534
0.25000	0.07816	0.25000	-0.04879
0.30000	0.08014	0.30000	-0.05090
0.35000	0.08053	0.35000	-0.05145
0.40000	0.07958	0.40000	-0.05036
0.45000	0.07723	0.45000	-0.04783
0.50000	0.07345	0.50000	-0.04417
0.55000	0.06850	0.55000	-0.03970
0.60000	0.06276	0.60000	-0.03460
0.65000	0.05654	0.65000	-0.02889
0.70000	0.05002	0.70000	-0.02276
0.75000	0.04324	0.75000	-0.01646
0.80000	0.03620	0.80000	-0.01028
0.85000	0.02878	0.85000	-0.00472
0.90000	0.02058	0.90000	-0.00049
0.95000	0.01029	0.95000	0.00158
0.98000	0.00402	0.98000	0.00108
0.99000	0.00200	0.99000	0.00062
1.00000	0.00000	1.00000	0.00000

Figure 50 Chord-Normalized Airfoil Data for Airfoil at a Z/span Coordinate of 0.55 on the Generated MICON Blade, with the Twist and Offset removed

CHAPTER 8

CONCLUSIONS

A numerical interpolation scheme has been developed that generates the three-dimensional surface of a twisted and tapered wind turbine blade. The scheme involves creating the frame of the blade (Chapter 3); transforming the blade frame from its physical domain to a computational domain (Chapter 4); and performing a bi-tension spline interpolation in the computational domain to determine the physical coordinates of any point on the blade surface (Chapter 2). A FORTRAN computer program has been written to implement the scheme. The program is able to

- (1) Input two or more normalized airfoils at some specific spanwise stations as the basis for interpolation.
- (2) Scale the input airfoil data according to the prescribed chord and thickness.
- (3) Generate the desired three-dimensional blade geometry for linear or nonlinear spanwise chord, thickness, and/or twist distributions.
- (4) Graphically output the generated three-dimensional blade profiles, which include the concentric plots of any number of spanwise station airfoils; and a perspective view, top view, and front view, respectively, of the generated blade.
- (5) Graphically output plots of the chord, twist, and thickness distribution curves of the generated and input airfoils.
- (6) Output in tabular form of all the interpolated airfoil sections specified by the user.

To demonstrate the feasibility, usefulness, and versatility of the numerical scheme, two different aerodynamic blade profiles were generated and presented in Chapter 7. The results

of this demonstration indicate that the numerical interpolation scheme can be a powerful tool for generating the surface coordinates for any highly twisted and tapered three-dimensional blade.

REFERENCES

1. Tangler, J. L. and Somers, D. M., "Advanced Airfoils for HAWTs," Wind Power '85 Proceedings, San Francisco, CA, August 1985.
2. Eppler, R., and Somers, D. M., "A Computer Program for the Design and Analysis of Low-Speed Airfoils," NASA TM-80210, 1980.
3. Tangler, J. L., Private Communication, Solar Energy Research Institute, Wind Energy Research Branch, Golden, Colorado.
4. Stoddard, F. S., Private Communication, Pioneer Wind Power, Amherst, Massachusetts.
5. Andrews, B. C., and Van Doren, K. R., "A Method for Geometric Modeling of Complex Wind Turbine Blades," Seventh ASME Wind Energy Symposium, SED-Vol. 5, p. 115, 1988.
6. Cline, A. K., "Scalar- and Planar-Valued Curve Fitting Using Splines Under Tension," Communications of the ACM, Vol. 17, pp. 218-220, 1974.
7. Pruess, S., "Properties of Splines in Tension," Journal of Approximating Theory, Vol. 17, pp. 86-96, 1976.
8. James, M. L., Smith, G. M. and Walford, J. C., Applied Numerical Methods for Digital Computation, 2nd edition, Harper and Row Publishers, New York, 1977.
9. Conte, S. D. and Boor, C., Elementary Numerical Analysis, An Algorithmic Approach, 3rd edition, McGraw Hill Book Company, New York, 1980.
10. Schweikert, D. G., "An Interpolation Curve Using A Spline in Tension," Journal of Mathematics and Physics, Vol. 45, pp. 312-317, 1966.

11. Späth, H., "Algorithm 16: Two-Dimensional Exponential Splines," Computing, Vol. 7, pp. 364-369, 1971.
12. Späth, H., Spline Algorithm for Curves and Surfaces, Utilitas Mathematica Publishing Inc., Winnipeg, Canada, 1974.
13. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Corporation, Washington, 1984.
14. Abbott, I. H., and Von Doenhoff, A. E., Theory of Wing Sections - Including a Summary of Airfoil Data, Dover Publications, Inc., New York, 1959.
15. Morgan, H. L., Jr., "Computer Programs for Smoothing and Scaling Airfoil Coordinates," NASA TM-84666, 1983.
16. Tu, P. K. C., and Scott, G. N., "Smoothing and Scaling Airfoil Coordinates on a Personal Computer," SERI/TR-257-3372, Solar Energy Research Institute, Wind Research Branch, Golden, Colorado, 1989.
17. Kuethe, A. M. and Chow, C. Y., Foundations of Aerodynamics: Bases of Aerodynamic Design, 4th edition, John Wiley & Sons Inc., New York, 1986.

APPENDIX A

PROGRAM LISTING

```

1      PROGRAM BLADE
2      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
3      PARAMETER(IX = 120 , IY = 10)
4      C
5      DIMENSION NDF(IY),NNN(IY),NMID(IY)
6      COMMON /C1/TH1(IY),OSX(IY),OSY(IY),CL(IY)/C2/X(IX,IY),Y(IX,IY)
7      COMMON /C3/X1(IX,IY),Y1(IX,IY)/C4/ZI(IX),ETA(IY)
8      COMMON /C5/XMIN(IY),XMAX(IY)/C6/DZI(IX),DETA(IY)
9      COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C8/IC,ZIT
10     COMMON /C9/AL(IX),BE(IY)/C10/V(IX),W(IY)
11     COMMON /C11/ALPHA,BETA/C12/Z(IY)/C13/U(IX,IY)
12     COMMON /C14A/XU(IX,IY),XL(IX,IY)
13     COMMON /C14B/YU(IX,IY),YL(IX,IY)
14     COMMON /C15/S(IY),T(IX)/C16/XIT,YIT
15     COMMON /C18A/XU1(IX,IY),YU1(IX,IY)
16     COMMON /C18B/XL1(IX,IY),YL1(IX,IY)
17     COMMON /C20/U1(IX,IY),U2(IX,IY)
18     COMMON /C30/NOW,ROR(IX),COR(IX),TWI(IX),THK(IX)
19     COMMON /C31/SIGMACH,SIGMATW,SIGMATH
20     C
21     1  CALL INPUT(NAF,NDF,NNN,NMID,SPAN,IDC,N1,SIGMA)
22     CALL SCOSTW(NAF,NNN,NMID,SPAN)
23     CALL TRANS(N1,NNN,NMID,NAF,SIGMA)
24     C
25     IAN = 0
26     C
27     5  CALL CHOICE(NAF)
28     CALL SECT(N1,NAF,ICT,SPAN)
29     CALL PLOTPR(ICT,IAN)
30     CALL SYSTEM('CLS')
31     WRITE(6,100)
32     READ(5,*) IAN
33     C
34     IF (IAN .EQ. 1) GOTO 5
35     C
36     CALL SYSTEM('CLS')
37     WRITE(6,200)
38     READ(5,*) IAN
39     C
40     IF (IAN .EQ. 1) GOTO 1
41     C
42     STOP
43     100 FORMAT(1X,///,
44     1      " Do you want to interpolate the 'SAME' input airfoils"
45     2      ,/, ' again ? ENTER ---> 0      No '
46     3      ,/, '                    1      Yes  ---> ')

```

PROGRAM BLADE Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

47 200 FORMAT(1X,/,/,/,
48 1      ' Do you want to interpolate a',/,/,
49 2      'NEW SET' ",/,/,
50 3      ' of input airfoils?',/,/,
51 4      ' ENTER ---> 0 No ',/,/,
52 5      ' 1 Yes ---> ')
53      END

```

SUBROUTINE CHOICE Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX

Source file Listing

```

54 C
55 C-----
56 C
57 SUBROUTINE CHOICE(NAF)
58 C
59 C*****
60 C The purpose of this subroutine is to allow the user to choose *
61 C plots of desired sections. *
62 C*****
63 C
64 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
65 PARAMETER(IX = 120 , IY = 10)
66 C
67 COMMON /C8/IC,ZIT/C12/Z(IY)/C3/X(IX,IY),Y(IX,IY)
68 C
69 CALL SYSTEM('CLS')
70 WRITE(6,90)
71 READ(5,*)IC
72 C
73 IF (IC .EQ. 1) THEN
74 1 WRITE(6,100)Z(1),Z(NAF)
75 READ(5,*)ZIT
76 C
77 IF (ZIT .LT. Z(1) .OR. ZIT .GT. Z(NAF)) THEN
78 WRITE(6,200)Z(1),Z(NAF)
79 GOTO 1
80 ENDIF
81 C
82 ENDIF
83 C
84 RETURN
85 90 FORMAT(1X,
86 1 ' YOU ARE NOW GIVEN THE FOLLOWING OPTIONS : '
87 2 ,/, ' 1.....GENERATE AN AIRFOIL AT ANY SPAN STATION.'
88 3 ,/, ' 2.....GENERATE A THREE-DIMENSIONAL BLADE.'
89 4 ,/, ' NOTE : '
90 5 ,/, ' Option 1 will furnish the user with an airfoil'
91 6 ,/, ' profile at the span station desired, as'
92 7 ,/, ' well as an airfoil normalized w.r.t. chord'
93 8 ,/, ' Option 2 will interpolate over the whole blade'
94 9 ,/, ' and generate the coordinates for any'
95 1 ,/, ' specified number of span station airfoils.'
96 2 ,/, ' Output plots you can create using option 2 are '
97 3 ,/, ' a) Concentric plots of the generated airfoils.'
98 4 ,/, ' b) Perspective view of the generated blade.'
99 5 ,/, ' c) Top view of the generated blade.'
100 6 ,/, ' d) Front view of the generated blade.'
101 7 ,/, ' e) Chord distribution along the span of the blade.'
102 8 ,/, ' f) Twist distribution along the span of the blade.'
103 9 ,/, ' g) Maximum thickness distribution along the span'
104 1 ' of blade.'
105 2 ,/, ' Enter your choice number here ---> ')

```

SUBROUTINE CHOICE Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
106 100 FORMAT(' The Z Coordnate must be between ',F7.2,' and ',F7.2,/,
107 1      ' Enter the Z coordinate of the desired span station '
108 2      ' here ---> ')
109 200 FORMAT(' The Z Coordnate must be between ',F7.2,' and ',F7.2,/,
110 1      ' You just entered Z = ',F7.2,' Please try AGAIN ! ')
111      END
```

SUBROUTINE CURVE1 Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file listing

```

112 C
113 C-----
114 C
115     SUBROUTINE CURVE1(N,X,Y,SLP1,SLPN,YP,TEMP,SIGMA)
116 C
117 C*****
118 C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO COMPUTE *
119 C AN INTERPOLATORY SPLINE UNDER TENSION THROUGH A SEQUENCE OF *
120 C FUNCTIONAL VALUES. THE SLOPES AT THE TWO ENDS OF THE CURVE MAY *
121 C BE SPECIFIED OR OMITTED. FOR ACTUAL COMPUTATION OF POINTS ON THE *
122 C CURVE IT IS NECESSARY TO CALL THE SUBROUTINE CURVE2. *
123 C *
124 C N = THE NUMBER OF VALUES TO BE INTERPOLATED (N.GE.2). *
125 C X = AN ARRAY OF THE N INCREASING ABSCISSAE OF THE FUNCTIONAL *
126 C VALUES. *
127 C Y = AN ARRAY OF THE N ORDINATES OF THE VALUES (I.E. Y(K) IS *
128 C THE FUNCTIONAL VALUE CORRESPONDING TO X(K)). *
129 C SLP1 AND SLPN = CONTAIN THE DESIRED VALUES FOR THE FIRST *
130 C DERIVATIVE OF THE CURVE AT X(1) AND X(N), *
131 C RESPECTIVELY. IF THE QUANTITY SIGMA IS *
132 C NEGATIVE THESE VALUES WILL BE DETERMINED *
133 C INTERNALLY AND THE USER NEED ONLY FURNISH *
134 C PLACE-HOLDING PARAMETERS FOR SLP1 AND SLPN. *
135 C SUCH PLACE-HOLDING PARAMETERS WILL BE IGNORED *
136 C BUT NOT DESTROYED *
137 C YP = AN ARRAY OF LENGTH AT LEAST N. *
138 C TEMP = AN ARRAY OF LENGTH AT LEAST N WHICH IS USED FOR *
139 C SCRATCH STORAGE, AND *
140 C SIGMA = THE TENSION FACTOR. THIS IS NON-ZERO AND INDICTES THE *
141 C CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO (E.G. *
142 C 0.01) THE RESULTING CURVE IS APPROXIMATELY A CUBIC *
143 C SPLINE. IF ABS(SIGMA) IS LARGE (E.G. 50.) THE *
144 C RESULTING CURVE IS NEARLY A POLYGONAL LINE. THE SIGN *
145 C OF SIGMA INDICATES WHETHER THE DERIVATIVE INFORMATION *
146 C HAS BEEN INPUT OR NOT. IF SIGMA IS NEGATIVE THE END- *
147 C POINT DERIVATIVES WILL BE DETERMINED INTERNALLY. A *
148 C STANDARD VALUES FOR SIGMA IS APPROXIMATELY 1, IN *
149 C ABSOLUTE VALUE. *
150 C *
151 C ON OUTPUT ---- *
152 C *
153 C YP = VALUES PROPORTIONAL TO THE 2ND DERIVATIVE OF THE CURVE *
154 C AT THE GIVEN NODES. *
155 C N, X, Y, SLP1, SLPN, AND SIGMA ARE UNALTERED. *
156 C*****
157 C
158     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
159     PARAMETER(IX = 120 , IY = 10)
160 C
161     DIMENSION X(N),Y(N),YP(IX),TEMP(IX)
162 C
163     NM1 = N - 1

```


SUBROUTINE CURVE1 Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

164      NP1 = N + 1
165      DELX1 = X(2) - X(1)
166      DX1 = (Y(2) - Y(1)) / DELX1
167      C
168      C*****
169      C DETERMINE SLOPES IF NECESSARY      *
170      C*****
171      C
172      IF (SIGMA .LT. 0.D0) GOTO 50
173      C
174      SLPP1 = SLP1
175      SLPPN = SLPN
176      C
177      C*****
178      C DENORMALIZE TENSION FACTOR      *
179      C*****
180      C
181      10  SIGMAP = DABS(SIGMA) * FLOAT(N-1) / (X(N)-X(1))
182      C
183      C*****
184      C SET UP RIGHT HAND SIDE AND TRIGIAGONAL SYSTEM FOR YP AND      *
185      C PERFORM FORWARD ELIMINATION      *
186      C*****
187      C
188      DELS = SIGMAP * DELX1
189      EXPS = DEXP(DELS)
190      SINHS = 0.5D0 * (EXPS - 1.D0/EXPS)
191      SINHIN = 1.D0 / (DELX1*SINHS)
192      DIAG1 = SINHIN * (DELS * 0.5D0 * (EXPS+1/EXPS) - SINHS)
193      DIAGIN = 1.D0/DIAG1
194      YP(1) = DIAGIN * (DX1-SLPP1)
195      SPDIAG = SINHIN * (SINHS-DELS)
196      TEMP(1) = DIAGIN * SPDIAG
197      IF (N .EQ. 2) GOTO 30
198      C
199      DO 20 I = 2,NM1
200      DELX2 = X(I+1) - X(I)
201      DX2 = (Y(I+1)-Y(I)) / DELX2
202      DELS = SIGMAP * DELX2
203      EXPS = DEXP(DELS)
204      SINHS = 0.5D0 * (EXPS-1.D0/EXPS)
205      SINHIN = 1.D0 / (DELX2*SINHS)
206      DIAG2 = SINHIN * (DELS * (0.5D0 * (EXPS+1.D0/EXPS)) - SINHS)
207      DIAGIN = 1.D0 / (DIAG1+DIAG2-SPDIAG*TEMP(I-1))
208      YP(I) = DIAGIN * (DX2-DX1-SPDIAG*YP(I-1))
209      SPDIAG = SINHIN * (SINHS-DELS)
210      TEMP(I) = DIAGIN * SPDIAG
211      DX1 = DX2
212      DIAG1 = DIAG2
213      20  CONTINUE
214      C
215      30  DIAGIN = 1.D0 / (DIAG1-SPDIAG*TEMP(NM1))

```

SUBROUTINE CURVE1 Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

216 C
217     YP(N) = DIAGIN * (SLPPN-DX2-SPDIAG*YP(NM1))
218 C
219 C*****
220 C PERFORM BACK SUBSTITUTION      *
221 C*****
222 C
223     DO 40 I = 2,N
224         IBAK = NP1-I
225         YP(IBAK) = YP(IBAK) - TEMP(IBAK) * YP(IBAK+1)
226 40    CONTINUE
227 C
228     RETURN
229 50    IF (N .EQ. 2) GOTO 60
230 C
231 C*****
232 C IF NO DERIVATIVES ARE GIVEN USE SECOND ORDER POLYNOMIAL      *
233 C INTERPOLATION ON INPUT DATA FOR VALUES AT ENDPOINTS.      *
234 C*****
235 C
236     DELX2 = X(3)-X(2)
237     DELX12 = X(3)-X(1)
238     C1 = -(DELX12+DELX1) / DELX12 / DELX1
239     C2 =  DELX12        / DELX1  / DELX2
240     C3 = - DELX1        /DELX12 /DELX2
241     SLPP1 = C1 * Y(1) + C2 * Y(2) + C3 * Y(3)
242     DELN  = X(N)  - X(NM1)
243     DELNM1 = X(NM1) - X(N-2)
244     DELNN  = X(N)  - X(N-2)
245     C1 = (DELNN+DELN) / DELNN / DELN
246     C2 = -DELNN      / DELN  / DELNM1
247     C3 =  DELN       / DELNN / DELNM1
248     SLPPN = C3 * Y(N-2) + C2 * Y(NM1) + C1 * Y(N)
249     GOTO 10
250 C
251 C*****
252 C IF ONLY TWO POINTS AND NO DERIVATIVES ARE GIVEN, USE STRAIGHT *
253 C LINE FOR CURVE      *
254 C*****
255 C
256 60    YP(1) = 0.D0
257        YP(2) = 0.D0
258 C
259     RETURN
260     END

```

FUNCTION CURVE2 Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

261 C
262 C-----
263 C
264 FUNCTION CURVE2(T,N,X,Y,YP,SIGMA,IT)
265 C
266 C*****
267 C THIS SUBROUTINE INTERPOLATES A CURVE AT A GIVEN POINT USING A *
268 C SPLINE UNDER TENSION. THE SUBROUTINE CURVE1 SHOULD BE CALLED *
269 C EARLIER TO DETERMINE CERTAIN NECESSARY PARAMETERS. *
270 C *
271 C ON INPUT ---- *
272 C *
273 C T = A REAL VALUE TO BE MAPPED ONTO THE INTERPOLATING CURVE *
274 C N = THE NUMBER OF POINTS WHICH WERE INTERPOLATED TO DETERMINE *
275 C THE CURVE *
276 C X AND Y = ARRAYS CONTAINING THE ORDINATES AND ABCISSAS OF THE *
277 C POINTS *
278 C YP = AN ARRAY WITH VALUES PROPORTIONAL TO THE SECOND *
279 C DERIVATIVE OF THE CURVE AT THE NODES *
280 C SIGMA = THE TENSION FACTOR (ITS SIGN IS IGNORED). *
281 C IT = IS AN INTEGER SWITCH. IF IT IS NOT 1 THIS INDICATES THAT *
282 C THE SUBROUTINE HAS BEEN CALLED PREVIOUSLY (WITH N, X, Y, *
283 C YP, AND SIGMA UNALTERED) AND THAT THIS VALUE OF T EXCEEDS *
284 C THE PREVIOUS VALUE. WITH SUCH INFORMATION THE SUBROUTINE *
285 C IS ABLE TO PERFORM THE INTERPOLATION MUCH MORE RAPIDLY. *
286 C IF A USER SEEKS TO INTERPOLATE AT A SEQUENCE OF POINTS, *
287 C EFFICIENCY IS GAINED BY ORDERING THE VALUES INCREASING *
288 C AND SETTING IT TO THE INDEX OF THE CALL. IF IT IS 1 THE *
289 C SEARCH FOR THE INTERVAL (X(K),X(K+1)) CONTAINING T STARTS *
290 C WITH K=1. *
291 C *
292 C THE PARAMETERS N, X, Y, YP, AND SIGMA SHOULD BE INPUT UNALTERED *
293 C FROM THE OUTPUT OF CURVE1. *
294 C *
295 C ON OUTPUT ---- *
296 C *
297 C YT = THE INTERPOLATED VALUE. FOR T LESS THAN, X(1) YT=Y(1). *
298 C FOR T GREATER THAN X(N), YT=Y(N). *
299 C YTDX = DERIVATIVE OF YT WITH RESPECT TO X. *
300 C *
301 C NONE OF THE INPUT PARAMETERS ARE ALTERED. *
302 C*****
303 C
304 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
305 PARAMETER(IX = 120 , IY = 10)
306 C
307 DIMENSION X(N),Y(N),YP(IX)
308 C
309 S = X(N)-X(1)
310 C
311 C*****
312 C DENORMALIZE SIGMA *
313 C*****

```

FUNCTION CURVE2 Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

314 C
315     SIGMAP = DABS(SIGMA) * FLOAT(N-1) / S
316 C
317 C*****
318 C IF IT.NE.1 START SEARCH WHERE PREVIOUSLY TERMINATED, OTHERWISE *
319 C START FROM BEGINNING *
320 C*****
321 C
322     IF (IT .EQ. 1) I1 = 2
323 C
324 C*****
325 C SERACH FOR INTERVAL *
326 C*****
327 C
328 10    DO 20 I = I1,N
329         IF (X(I)-T)20,20,30
330 20    CONTINUE
331 C
332     I = N
333 C
334 C*****
335 C CHECK TO INSURE CORRECT INTERVAL *
336 C*****
337 C
338 30    IF (X(I-1) .LE. T .OR. T .LE. X(I)) GOTO 40
339 C
340 C*****
341 C RESTART SERACH AND RESET I1 *
342 C (INPUT " IT " WAS INCORRECT) *
343 C*****
344 C
345     I1 = 2
346     GOTO 10
347 C
348 C*****
349 C SET UP AND PERFORM INTERPOLATION *
350 C*****
351 C
352 40    DEL1 = T - X(I-1)
353         DEL2 = X(I) - T
354         DEL3 = X(I) - X(I-1)
355         EXPS1 = DEXP(SIGMAP*DEL1)
356         SINHD1 = 0.5D0 * (EXPS1-1.D0/EXPS1)
357         EXPS = DEXP(SIGMAP*DEL2)
358         SINHD2 = 0.5D0 * (EXPS-1.D0/EXPS)
359         EXPS = EXPS1 * EXPS
360         SINHS = 0.5D0 * (EXPS-1.D0/EXPS)
361         A1 = (YP(I) * SINHD1 + YP(I-1) * SINHD2) / SINHS
362         A2 = ((Y(I)-YP(I)) * DEL1+(Y(I-1)-YP(I-1)) * DEL2) / DEL3
363         CURVE2 = A1 + A2
364         I1 = 1
365 C
366     RETURN

```

F77L - Lahey FORTRAN 77, Version 2.22 03/21/91 22:23:03
Page 10

FUNCTION CURVE2 Compiling Options:/NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

367 END

SUBROUTINE CHZE Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

368 C
369 C-----
370 C
371 SUBROUTINE CHZE(I,J)
372 C
373 C*****
374 C This subroutine determines the two 4x4 coefficient matrices, i.e. *
375 C CHZY and CHZX in equation (2.20) of manual. These are then *
376 C input in SUBROUTINE COEFM. *
377 C*****
378 C
379 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
380 PARAMETER(IX = 120 , IY = 10)
381 C
382 COMMON /CHZXY/CHZX(4,4),CHZY(4,4)
383 COMMON /C10/V(IX),W(IY)/C6/DX(IX),DY(IY)
384 C
385 DIMENSION CHZ(4,4,2)
386 DIMENSION Z(2),H(2)
387 C
388 H(1) = DX(I)
389 Z(1) = V(I)
390 H(2) = DY(J)
391 Z(2) = W(J)
392 C
393 DO 100 K = 1,2
394 WON = 1.D0 / H(K)
395 WOZ = 1.D0 / (1.D0-Z(K))
396 WOP = 1.D0 / (1.D0-Z(K)*Z(K))
397 CHZ(1,1,K) = 0.D0
398 CHZ(1,2,K) = 0.D0
399 CHZ(1,3,K) = WON
400 CHZ(1,4,K) = 0.D0
401 CHZ(2,1,K) = WON
402 CHZ(2,2,K) = 0.D0
403 CHZ(2,3,K) = 0.D0
404 CHZ(2,4,K) = 0.D0
405 CHZ(3,1,K) = -1.D0 * WON * WOZ
406 CHZ(3,2,K) = -1.D0 * WOP
407 CHZ(3,3,K) = WON * WOZ
408 CHZ(3,4,K) = -1.D0 * Z(K) * WOP
409 CHZ(4,1,K) = WON * WOZ
410 CHZ(4,2,K) = Z(K) * WOP
411 CHZ(4,3,K) = -1.D0 * WON * WOZ
412 CHZ(4,4,K) = WOP
413 100 CONTINUE
414 C
415 DO 200 K = 1,4
416 C
417 DO 300 L = 1,4
418 CHZX(K,L) = CHZ(K,L,1)
419 CHZY(K,L) = CHZ(K,L,2)

```

F77L - Lahey FORTRAN 77, Version 2.22 03/21/91 22:23:03

Page 12

SUBROUTINE CHZE Compiling Options:/NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX

Source file Listing

```
420 300 CONTINUE
421 C
422 200 CONTINUE
423 C
424 RETURN
425 END
```

Source file Listing

```
426 C
427 C-----
428 C
429 SUBROUTINE CKFILE(FNAME)
430 C
431 C*****
432 C This subroutine checks/deletes an existing file. *
433 C*****
434 C
435 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
436 CHARACTER FNAME*(*),DELCMD*44
437 LOGICAL*1 FEXIST
438 C
439 INQUIRE (FILE = FNAME , EXIST = FEXIST)
440 C
441 IF (FEXIST) THEN
442 WRITE (DELCMD,10) FNAME
443 CALL SYSTEM (DELCMD)
444 ENDIF
445 C
446 RETURN
447 10 FORMAT('DEL ',A40)
448 END
```


SUBROUTINE COEFM Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
449 C
450 C-----
451 C
452 C      SUBROUTINE COEFM
453 C
454 C*****
455 C This subroutine determines the 4x4 coefficient matrix, i.e.      *
456 C equation (2.19) of manual, for each sector of the computational  *
457 C domain.                                                           *
458 C*****
459 C
460 C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
461 C
462 C      COMMON /AIJKIJ/AK(4,4)/CHZXY/CHZX(4,4),CHZY(4,4)
463 C
464 C      DIMENSION CHZYT(4,4)
465 C
466 C      CALL TRANSPO(4,4,CHZY,CHZYT)
467 C      CALL MATM(4,4,4,AK,CHZYT,CHZY)
468 C      CALL MATM(4,4,4,CHZX,CHZY,AK)
469 C
470 C      RETURN
471 C      END
```

```

472 C
473 C-----
474 C
475 SUBROUTINE DINPUT(NAF,NDF,NNN,NMID,SPAN,IDC,N1,SIGMA,IU)
476 C
477 C*****
478 C This subroutine performs the "DATA FILE" input mode option *
479 C*****
480 C
481 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
482 PARAMETER(IX = 120 , IY = 10)
483 C
484 CHARACTER*1 SCALE
485 CHARACTER*12 NAME(IY),FNAME,DNAME
486 C
487 COMMON /C1/TH1(IY),OSX(IY),OSY(IY),CL(IY)/C2/XX(IX,IY),YY(IX,IY)
488 COMMON /C12/ZZ(IY)/C11/ALPHA,BETA
489 COMMON /C14A/XU(IX,IY),XL(IX,IY)
490 COMMON /C14B/YU(IX,IY),YL(IX,IY)
491 COMMON /C30/NOW,ROR(IX),COR(IX),TWI(IX),THK(IX)
492 COMMON /C31/SIGMACH,SIGMATW,SIGMATH
493 C
494 DIMENSION NDF(IY),NNN(IY)
495 DIMENSION ZE(IX),CHORD(IX),VAR(8),THKN(IX),NMID(IY)
496 C
497 DATA VAR/ 300.0D0,0.0D0,5.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0/
498 C
499 CALL SYSTEM('CLS')
500 WRITE(6,121)
501 C
502 READ(IU,*)IDC
503 C
504 IF (IDC .EQ. 2) DNAME = 'CTE1.DAT'
505 C
506 IF (IDC .EQ. 3) DNAME = 'CTE2.DAT'
507 C
508 IF (IDC .EQ. 4) THEN
509 READ(IU,*)DNAME
510 ENDIF
511 C
512 IF (IDC .GT. 1) THEN
513 OPEN(UNIT = 10 , FILE = DNAME , STATUS = 'OLD')
514 ENDIF
515 C
516 CALL CKFILE('CTEX1.DAT')
517 OPEN(UNIT = 20 , FILE = 'CTEX1.DAT', STATUS = 'NEW')
518 C
519 C Step 2
520 C
521 READ(IU,*)NAF
522 C
523 READ(IU,*)SPAN

```

SUBROUTINE DINPUT Compiling Options: /NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

524 C
525     READ(IU,*)SCALE
526 C
527     IF (IDC .GT. 1) THEN
528         READ(10,*)NOW
529 C
530         DO 8 I = 1,NOW
531             READ(10,*)ROR(I),COR(I),TWI(I),THK(I)
532             WRITE(20,205)ROR(I),COR(I),TWI(I),THK(I) * 100.0D0
533             ZE(I) = ROR(I) * SPAN
534             CHORD(I) = COR(I) * SPAN
535 8      CONTINUE
536 C
537         CLOSE(UNIT = 10)
538     ENDIF
539 C
540     IF (IDC.EQ.1) NOW = NAF
541     MAX = 0
542     OPEN(UNIT = 8 , FILE = 'A10A.DAT' , STATUS = 'UNKNOWN')
543 C
544 C Step 3
545 C
546     DO 100 K = 1,NAF
547 C
548         READ(IU,*)NDF(K)
549 C
550         IF (NDF(K) .EQ. 1) NAME(K) = 'S806A.DAT'
551         IF (NDF(K) .EQ. 2) NAME(K) = 'S805A6.DAT'
552         IF (NDF(K) .EQ. 3) NAME(K) = 'S805A.DAT'
553         IF (NDF(K) .EQ. 4) NAME(K) = 'S805A7.DAT'
554         IF (NDF(K) .EQ. 5) NAME(K) = 'S807.DAT'
555         IF (NDF(K) .EQ. 6) NAME(K) = 'S808.DAT'
556         IF (NDF(K) .EQ. 7) THEN
557             READ(IU,*)NAME(K)
558         ENDIF
559 C
560         FNAME = NAME(K)
561         LUN = 10
562         OPEN(UNIT = LUN,FILE = FNAME , STATUS = 'OLD' )
563 C
564         READ(LUN,*)NU
565         READ(LUN,*)NL
566         N = NU + NL - 1
567         NNN(K) = N
568 C
569         DO 50 I = 1,NU
570             READ(LUN,*)XU(I,K),YU(I,K)
571 50      CONTINUE
572 C
573         DO 60 I = 1,NL
574             READ(LUN,*)XL(I,K),YL(I,K)
575 60      CONTINUE

```

SUBROUTINE DINPUT Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

576 C
577      CLOSE(UNIT = LUN)
578 C
579      READ(IU,*)ZZ(K)
580 C
581      IF (IDC .GT. 1) THEN
582          ZIT = ZZ(K)
583          SIGMACH = -1.D0
584          SIGMATW = -1.D0
585          SIGMATH = -1.D0
586          CALL SPT11D(NOW , ZE , CHORD , SIGMACH , ZIT , CHL )
587          CALL SPT11D(NOW , ZE , TWI , SIGMATW , ZIT , TWA )
588          CALL SPT11D(NOW , ZE , THK , SIGMATH , ZIT , THICK)
589          CL(K) = CHL
590          TH1(K) = TWA
591          THKN(K) = THICK
592      ELSE
593 C
594          READ(IU,*)TH1(K)
595 C
596          READ(IU,*)CL(K)
597 C
598          READ(IU,*)THKN(K)
599 C
600          THICK = THKN(K)
601 C
602          ROR(K) = ZZ(K)/SPAN
603          COR(K) = CL(K)/SPAN
604          TWI(K) = TH1(K)
605          THK(K) = THKN(K)
606 C
607          WRITE(20,205)ROR(K) , COR(K) , TH1(K) , THKN(K)*100.0D0
608 C
609      ENDIF
610 C
611      READ(IU,*)TWCTR
612 C
613      IF (TWCTR .EQ. 1) THEN
614          READ(IU,*)OSX(K)
615          READ(IU,*)OSY(K)
616      ENDIF
617 C
618      IF (TWCTR .EQ. 2) THEN
619          TWCTRX = 0.3D0
620          SIG = -1.D0
621          CALL SPT11D(NU,XU,YU,SIG,TWCTRX,TWCTYU)
622          CALL SPT11D(NL,XL,YL,SIG,TWCTRX,TWCTYL)
623          OSX(K) = TWCTRX
624          OSY(K) = 0.5D0 * (TWCTYU-TWCTYL)
625      ENDIF
626 C
627 C SET UP FOR SCALING
628 C

```

SUBROUTINE DINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

629      IF (SCALE .EQ. 'Y' .OR. SCALE .EQ. 'y') THEN
630          CALL CKFILE('AFXYIN')
631          OPEN (15 , FILE = 'AFXYIN' , STATUS = 'NEW')
632          WU = 1.0D0
633          NU1 = NU-1
634          VAR(1) = 0.D0
635          WRITE(15,400)CHARNB(NAME(K))
636          WRITE(15,401)(VAR(I),I = 1,8)
637          WRITE(15,402)NU
638          YU(NU,K) = (YU(NU,K) + YU(NU1,K)) / 2.0D0
639      C
640          DO 66 I = 1,NU
641              WRITE(15,403)XU(I,K),YU(I,K),WU
642      66      CONTINUE
643      C
644          WRITE(15,402)NL
645      C
646          DO 67 I = 1,NL
647              WRITE(15,403)XL(I,K),YL(I,K),WU
648      67      CONTINUE
649      C
650          CLOSE(UNIT = 15)
651      C
652          CALL CKFILE('SCXYIN')
653      C
654          CALL SYSTEM('SMOOTH2')
655      C
656          OPEN (UNIT = 16 , FILE = 'SCALEINP.DAT' , ACCESS = 'APPEND'
657      /      , STATUS = 'OLD')
658      C
659          XNT = 1.0D0
660          WRITE(16,402)XNT
661          WRITE(16,402)THICK
662          CLOSE(UNIT = 16)
663      C
664          CALL SYSTEM('RENAME SCALEINP.DAT SCXYIN')
665      C
666          CALL SYSTEM('SCALE1')
667      C
668      C CONCLUSION OF SCALING
669      C
670      C
671      C -- RETRIEVE SCALED DATA FROM DATA FILES --
672      C
673          OPEN (UNIT = 17 , FILE = 'XUXLSS.DAT' , STATUS = 'OLD')
674          READ(17,*)NU
675          READ(17,*)(XU(I,K),YU(I,K),I = 1,NU)
676          READ(17,*)NL
677          READ(17,*)(XL(I,K),YL(I,K),I = 1,NL)
678          CLOSE(UNIT = 17)
679      C
680      ENDIF

```

SUBROUTINE DINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
681 C
682 C   CONTINUE AND FORM THE AIRFOIL
683 C
684       N = NU + NL - 1
685       NNN(K) = N
686       NMID(K) = NL
687       YU(NU,K) = 0.0D0
688       YL(NL,K) = 0.0D0
689 C
690       DO 70 I = 1,N
691           J = NL - I + 1
692           L = I - NL + 1
693 C
694           IF (I .LT. NL) THEN
695 C
696               XX(I,K) = XL(J,K)
697               YY(I,K) = YL(J,K)
698 C
699           ELSE
700 C
701               XX(I,K) = XU(L,K)
702               YY(I,K) = YU(L,K)
703 C
704           ENDIF
705 C
706           WRITE(8,*)XX(I,K),YY(I,K),ZZ(K)/SPAN
707 C
708 70    CONTINUE
709 C
710       IF (NNN(K) .GT. MAX) MAX = NNN(K)
711 C
712 100   CONTINUE
713 C
714       CLOSE (UNIT = 8)
715 C
716 C   Step 4
717 C
718       SIGMA = -1.D0
719       READ(IU,*)N1
720 C
721       READ(IU,*)ALPHA
722 C
723       READ(IU,*)BETA
724 C
725       CLOSE(UNIT = IU)
726 C
727       OPEN(UNIT = 100 , FILE = 'INPUT.DAT' , ACCESS = 'APPEND'
728       /      , FORM = 'FORMATTED',CARRIAGE CONTROL = 'FORTRAN'
729       /      , STATUS = 'UNKNOWN')
730 C
731       WRITE(100,105)NAF,ALPHA,BETA
732 C
733       DO 200 K = 1,NAF
```

SUBROUTINE DINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

734 C
735     WRITE(100,106)K,NAME(K)
736     WRITE(100,203)
737     WRITE(100,220)CL(K),TH1(K),OSX(K),OSY(K),ZZ(K),THKN(K)
738     WRITE(100,107)
739     IA = (NNN(K) + 1) / 2
740 C
741     IF (NMID(K) .GE. IA) THEN
742         NREF = NMID(K)
743     ELSE
744         NREF = NNN(K) - NMID(K) + 1
745     ENDIF
746 C
747     DO 120 I = 1,NREF
748         WRITE(100,210)XU(I,K),YU(I,K),XL(I,K),YL(I,K)
749 C
750 120    CONTINUE
751 C
752 200    CONTINUE
753 C
754     N = MAX
755 C
756     CLOSE(UNIT = 20)
757     CLOSE(UNIT = 100)
758 C
759     RETURN
760 105    FORMAT(/,10X,' Number of input airfoils = ',I2,/,10X,
761 /      ' Tension parameters: Alpha = ',F7.2,', and Beta = ',F7.2)
762 106    FORMAT(1H1,/,10X,' Input data file name for airfoil',I2,' is '
763 /      ',A12,' and the input data are:',/)
764 107    FORMAT(/,17X,'XU',12X,'YU',13X,'XL',11X,'YL',/)
765 121    FORMAT(1X,/, ' Please wait, program is running!')
766 201    FORMAT(1X,I2)
767 202    FORMAT(1X,2F8.2)
768 203    FORMAT(7X,'Chord-',7X,'Twist',10X,'Twist Center',9X,'Span',
769 /      9X,'max',/,7X,'Length',7X,'Angle',10X,'X',10X,'Y',8X,'Coord.'
770 /      ',5X,'thickness')
771 205    FORMAT(1X,4(F9.5,1X))
772 210    FORMAT(10X,4(2X,E12.5))
773 220    FORMAT(2X,6(2X,F10.4))
774 400    FORMAT(A)
775 401    FORMAT(8F10.5)
776 402    FORMAT(F10.5)
777 403    FORMAT(3E15.6)
778     END

```

SUBROUTINE GETX Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

779 C
780 C-----
781 C
782     SUBROUTINE GETX(N,M,TAU,SMID)
783 C
784 C*****
785 C This subroutine determines the X-coordinate of all points on the *
786 C surface of an airfoil at a particular span station of the blade. *
787 C*****
788 C
789     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
790     PARAMETER(IX = 120 , IY = 10)
791 C
792     COMMON /C4/ZI(IX),ETA(IY)/C6/DZI(IX),DETA(IY)
793     COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
794     COMMON /C10/V(IX),W(IY)/C9/AL(IX),BE(IY)/C16/XIT,YIT
795     COMMON /C20/U1(IX,IY),U2(IX,IY)
796     COMMON /C22/XZ(IX),YZ(IX)
797 C
798     DO 10 J = 1,M
799 C
800         DO 20 I = 1,N
801             U(I,J) = U1(I,J)
802         20 CONTINUE
803 C
804     10 CONTINUE
805 C
806     CALL PDERIV(N,M)
807 C
808     IA = (N + 1) / 2
809     DZL = SMID / FLOAT(IA-1)
810     DZU = (1.0D0 - SMID) / FLOAT(IA-1)
811     B1 = 1.0D0 + (DEXP(TAU) - 1.0D0) * SMID
812     B2 = 1.0D0 + (DEXP(-TAU) - 1.0D0) * SMID
813     B = DLOG(B1/B2) * 0.5D0 / TAU
814 C
815     DO 100 I = 1,N
816         IF (I .EQ. IA) THEN
817             XIT = SMID
818         ELSE
819             IF (I .LT. IA) ZIT = DZL * FLOAT(I-1)
820             IF (I .GT. IA) ZIT = DZU * FLOAT(I-IA) + SMID
821             XIT = SMID * (1.0D0 + (DSINH(TAU*(ZIT-B)) / DSINH(TAU*B)))
822         ENDIF
823         CALL GETU(N,M,UIT)
824         XZ(I) = UIT
825     100 CONTINUE
826 C
827     RETURN
828     END

```


SUBROUTINE GETY Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

829 C
830 C-----
831 C
832       SUBROUTINE GETY(N,M,TAU,SMID)
833 C
834 C*****
835 C This subroutine determines the Y-coordinate of all points on the *
836 C surface of an airfoil at a particular span station of the blade. *
837 C*****
838 C
839       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
840       PARAMETER(IX = 120 , IY = 10)
841 C
842       COMMON /C4/ZI(IX),ETA(IY)/C6/DZI(IX),DETA(IY)
843       COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
844       COMMON /C10/V(IX),W(IY)/C9/AL(IX),BE(IY)/C16/XIT,YIT
845       COMMON /C20/U1(IX,IY),U2(IX,IY)
846       COMMON /C22/XZ(IX),YZ(IX)
847 C
848       DO 10 J = 1,M
849 C
850           DO 20 I = 1,N
851               U(I,J) = U2(I,J)
852           20 CONTINUE
853 C
854       10 CONTINUE
855 C
856       CALL PDERIV(N,M)
857 C
858       IA = (N + 1) / 2
859       DZL = SMID / FLOAT(IA-1)
860       DZU = (1.0D0 - SMID) / FLOAT(IA-1)
861       B1 = 1.0D0 + (DEXP(TAU) - 1.0D0) * SMID
862       B2 = 1.0D0 + (DEXP(-TAU) - 1.0D0) * SMID
863       B = DLOG(B1/B2) * 0.5D0 / TAU
864 C
865       DO 100 I = 1,N
866           IF (I .EQ. IA) THEN
867               XIT = SMID
868           ELSE
869               IF (I.LT.IA) ZIT = DZL * FLOAT(I-1)
870               IF (I.GT.IA) ZIT = DZU * FLOAT(I-IA) + SMID
871               XIT = SMID * (1.0D0 + (DSINH(TAU*(ZIT-B)) / DSINH(TAU*B)))
872           ENDIF
873       CALL GETU(N,M,UIT)
874       YZ(I) = UIT
875       100 CONTINUE
876 C
877       RETURN
878       END

```

SUBROUTINE GETU Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
879 C
880 C-----
881 C
882 SUBROUTINE GETU(N,M,UIT)
883 C
884 C-----
885 C The purpose of this routine is to determine the ORDINATE value at
886 C the points xit and yit in the computational domain.The value of the
887 C ordinate is passed out as UIT.
888 C-----
889 C
890 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
891 PARAMETER(IX = 120 , IY = 10)
892 C
893 COMMON /C4/ZI(IX),ETA(IY)/C6/DZI(IX),DETA(IY)
894 COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
895 COMMON /C10/V(IX),W(IY)/C9/AL(IX),BE(IY)/C16/XIT,YIT
896 COMMON /AIJKIJ/AK(4,4)/CHZXY/CHZX(4,4),CHZY(4,4)
897 COMMON /PHIXY/PHIX(4),PHIY(4)
898 C
899 CALL SORT(N,M,IL,JL)
900 CALL KAY(IL,JL)
901 CALL CHZE(IL,JL)
902 CALL COEFM
903 CALL PHI(IL,JL)
904 SUM = 0.D0
905 C
906 DO 100 K = 1,4
907 C
908 DO 200 L = 1,4
909 SUM = SUM + AK(K,L) * PHIX(K) * PHIY(L)
910 200 CONTINUE
911 C
912 100 CONTINUE
913 C
914 UIT = SUM
915 C
916 RETURN
917 END
```

SUBROUTINE HRDOUT

Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX

Source file Listing

```

918 C
919 C-----
920 C
921 SUBROUTINE HRDOUT(SCREEN,PLOTTER,PRINTR)
922 C
923 C*****
924 C This subroutine controls graphic output device options.  *
925 C*****
926 C
927 CHARACTER*7 SCREEN, PLOTTER, PRINTR
928 C
929 C IF THE USER WANTS TO VIEW OR PLOT OUTPUT DATA HE/SHE HAS THE OPTION
930 C
931 PRINT, ' WOULD YOU LIKE TO VIEW THIS ?'
932 PRINT, ' ENTER ---> 0 NO. '
933 PRINT, ' 1 YES. ---> '
934 C
935 READ(5,*) IANS
936 C
937 IF (IANS .EQ. 1) CALL SYSTEM(SCREEN)
938 C
939 10 PRINT, ' DO YOU WANT TO SEND THIS VIEW TO A HARD-COPY DEVICE ?'
940 PRINT, ' ENTER ---> 0 NO.'
941 PRINT, ' 1 YES, SEND TO PLOTTER.'
942 PRINT, ' 2 YES, SEND TO PRINTER.'
943 PRINT, ' ENTER YOUR CHOICE NUMBER HERE ---> '
944 C
945 READ(5,*) IANS
946 C
947 IF (IANS .EQ. 1) THEN
948 PRINT *
949 PRINT, ' PLOTTING.....'
950 CALL SYSTEM(PLOTTER)
951 GO TO 10
952 ENDIF
953 C
954 IF (IANS .EQ. 2) THEN
955 PRINT *
956 PRINT, ' PRINTER FILE GENERATING.....PLEASE WAIT.....'
957 CALL SYSTEM(PRINTR)
958 PRINT, ' PRINTING.....'
959 CALL SYSTEM('DPRINT DUMFIL')
960 GO TO 10
961 ENDIF
962 C
963 RETURN
964 END

```

SUBROUTINE INPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

965 C
966 C-----
967 C
968     SUBROUTINE INPUT(NAF,NDF,NNN,NMID,SPAN,IDC,N1,SIGMA)
969 C
970 C*****
971 C This subroutine performs all major input necessary to run the  *
972 C program.                                                         *
973 C*****
974 C
975     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
976     PARAMETER(IX = 120 , IY = 10)
977 C
978     CHARACTER*1 CHOICE
979     CHARACTER*12 INAME
980 C
981     COMMON /C1/TH1(IY),OSX(IY),OSY(IY),CL(IY)/C2/XX(IX,IY),YY(IX,IY)
982     COMMON /C12/ZZ(IY)/C11/ALPHA,BETA
983     COMMON /C14A/XU(IX,IY),XL(IX,IY)
984     COMMON /C14B/YU(IX,IY),YL(IX,IY)
985     COMMON /C30/NOW,ROR(IX),COR(IX),TWI(IX),THK(IX)
986     COMMON /C31/SIGMACH,SIGMATW,SIGMATH
987 C
988     CALL CKFILE('INPUT.DAT')
989     CALL CKFILE('WTOUT.DAT')
990     CALL CKFILE('NTOUT.DAT')
991     CALL SYSTEM('CLS')
992 C
993     WRITE(6,90)
994     PAUSE
995 C
996     CALL SYSTEM('CLS')
997 C
998     WRITE(6,91)
999     READ(5,*)CHOICE
1000    CALL SYSTEM('CLS')
1001 C
1002    IF (CHOICE .EQ. 'D' .OR. CHOICE .EQ. 'd') THEN
1003        WRITE(6,92)
1004        READ(5,*)INAME
1005        IU = 50
1006        OPEN(UNIT = IU , FILE = INAME , STATUS = 'OLD')
1007        CALL DINPUT(NAF,NDF,NNN,NMID,SPAN,IDC,N1,SIGMA,IU)
1008    ELSE
1009        IU = 5
1010        CALL TINPUT(NAF,NDF,NNN,NMID,SPAN,IDC,N1,SIGMA,IU)
1011    ENDIF
1012 C
1013    RETURN
1014 C
1015 90  FORMAT(1X,/,
1016 1      '      This program uses certain cross-sectional airfoil'
1017 2      ',/, ' data as input, and proceeds to interpolate these'

```

SUBROUTINE INPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
1018      3  ,/, ' input data by use of a bi-tension spline method.'
1019      4  ,/, '   As a result of this interpolation, the whole surface'
1020      5  ,/, ' of the blade is definable.'
1021      6  ,/, ' This program will then generate airfoil data for'
1022      7  ,/, ' any particular span stations of the blade. '
1023      8  ,/, ' *** READ AND FOLLOW INSTRUCTIONS CAREFULLY ***')
1024  91  FORMAT(1X
1025      1  ,/, ' You have two alternative ways of entering data : '
1026      2  ,/, '           Enter T for the terminal input'
1027      3  ,/, '           Enter D for the data file input'
1028      4  ,/, ' Enter your choice here ---> ')
1029  92  FORMAT(1X,/,
1030      1  '           Please enter the name of the data input file ---> ')
1031      END
```

SUBROUTINE KAY Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1032 C
1033 C-----
1034 C
1035      SUBROUTINE KAY(I,J)
1036 C
1037 C*****
1038 C This subroutine creates the 4x4 coefficient matrix, i.e.      *
1039 C equation (2.21) of the manual, which contains all the values of *
1040 C U and its partial derivatives at the four nodes of each sector *
1041 C in the computational domain.                                     *
1042 C*****
1043 C
1044      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1045      PARAMETER(IX = 120 , IY = 10)
1046 C
1047      COMMON /C13/U(IX,IY)/C7/P(IX,IY),Q(IX,IY),R(IX,IY)
1048      COMMON /AIJKIJ/AK(4,4)
1049 C
1050      JP1 = J + 1
1051      IP1 = I + 1
1052 C
1053      AK(1,1) = U(I,J)
1054      AK(1,2) = Q(I,J)
1055      AK(1,3) = U(I,JP1)
1056      AK(1,4) = Q(I,JP1)
1057      AK(2,1) = P(I,J)
1058      AK(2,2) = R(I,J)
1059      AK(2,3) = P(I,JP1)
1060      AK(2,4) = R(I,JP1)
1061      AK(3,1) = U(IP1,J)
1062      AK(3,2) = Q(IP1,J)
1063      AK(3,3) = U(IP1,JP1)
1064      AK(3,4) = Q(IP1,JP1)
1065      AK(4,1) = P(IP1,J)
1066      AK(4,2) = R(IP1,J)
1067      AK(4,3) = P(IP1,JP1)
1068      AK(4,4) = R(IP1,JP1)
1069 C
1070      RETURN
1071      END

```

SUBROUTINE MATM Compiling Options: /NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
1072 C
1073 C-----
1074 C
1075 SUBROUTINE MATM(N,M,L,A,B,C)
1076 C
1077 C*****
1078 C This subroutine calculates the product of two matrices, *
1079 C A(I,J)*B(J,K), and stores the results in matrix C(I,J). *
1080 C*****
1081 C
1082 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1083 C
1084 DIMENSION A(N,M),B(M,L),C(N,L)
1085 C
1086 DO 60 I = 1,N
1087 C
1088 DO 50 K = 1,L
1089 SUM = 0.00
1090 C
1091 DO 40 J = 1,M
1092 SUM = SUM + A(I,J) * B(J,K)
1093 40 CONTINUE
1094 C
1095 C(I,K) = SUM
1096 50 CONTINUE
1097 C
1098 60 CONTINUE
1099 C
1100 RETURN
1101 END
```

SUBROUTINE PDERIV Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1102 C
1103 C-----
1104 C
1105       SUBROUTINE PDERIV(N,M)
1106 C
1107 C*****
1108 C This routine receives the transformed coordinates as input data. *
1109 C It then proceeds to find the partial derivatives of Y and X with *
1110 C respect to ZI and ETA. *
1111 C *
1112 C NOTE: X, Y, and U represent ZI, ETA and X or Y respectively. *
1113 C *
1114 C DX and DY (ie. dxi and dxi) are computed first. *
1115 C*****
1116 C First we will normalize the tension factors AL and BE. *
1117 C Simultaneously we find DZI and DETA , represented by DX and DY. *
1118 C*****
1119 C
1120       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1121       PARAMETER(IX = 120 , IY = 10)
1122 C
1123       COMMON /C4/X(IX),Y(IY)/C6/DX(IX),DY(IY)/C11/ALPHA,BETA
1124       COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
1125       COMMON /C15/S(IY),T(IX)/C10/V(IX),W(IY)/C9/AL(IX),BE(IY)
1126       COMMON /TRIDM/CL(IX),CM(IX),CN(IX),PQ(IX)
1127 C
1128       NM1 = N-1
1129       MM1 = M-1
1130 C
1131       DO 10 I = 1,NM1
1132           IP1 = I + 1
1133           AL(I) = ALPHA * FLOAT(NM1)
1134           DX(I) = X(IP1) - X(I)
1135 10      CONTINUE
1136 C
1137       DO 20 J = 1,MM1
1138           JP1 = J + 1
1139           BE(J) = BETA * FLOAT(MM1)
1140           DY(J) = Y(JP1) - Y(J)
1141 20      CONTINUE
1142 C
1143 C Next the coefficients V and T are computed
1144 C
1145       DO 30 I = 1,NM1
1146           ADX = AL(I) * DX(I)
1147 C
1148           IF (ADX .GT. 0.1D0 .AND. ADX .LT. 0.1D0) THEN
1149               ADX2 = ADX * ADX
1150               XAN = 1.0D0 + ADX2 * (0.1D0 + ADX2 / 280.0D0)
1151               XAD = 1.0D0 + ADX2 * (0.05D0 + ADX2 / 840.0D0)
1152               V(I) = 2.0D0 * XAN / XAD
1153               SINHDX = DSINH(ADX)

```


SUBROUTINE PDERIV Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1154         SAME = SINHDX - ADX
1155     ELSE
1156         SINHDX = DSINH(ADX)
1157         SAME = SINHDX - ADX
1158         V(I) = (ADX * DCOSH(ADX) - SINHDX) / SAME
1159     ENDIF
1160 C
1161         T(I) = ADX * AL(I) * SINHDX / ((V(I) * V(I) - 1.D0) * SAME)
1162 C
1163     30 CONTINUE
1164 C
1165 C Here the coefficients W and S are computed
1166 C
1167     DO 40 J = 1,MM1
1168         BDY = BE(J) * DY(J)
1169 C
1170         IF (BDY .GT. 0.D0 .AND. BDY .LT. .0.1D0) THEN
1171             BDY2 = BDY * BDY
1172             YAN = 1.D0 + BDY2 * (0.1D0 + BDY2 / 280.D0)
1173             YAD = 1.D0 + BDY2 * (0.05D0 + BDY2 / 840.D0)
1174             W(J) = 2.D0 * YAN / YAD
1175             SINHDY = DSINH(BDY)
1176             SAME = SINHDY - BDY
1177         ELSE
1178             SINHDY = DSINH(BDY)
1179             SAME = SINHDY - BDY
1180             W(J) = (BDY * DCOSH(BDY) - SINHDY) / SAME
1181         ENDIF
1182 C
1183         S(J) = BDY * BE(J) * SINHDY / ((W(J) * W(J) - 1.D0) * SAME)
1184 C
1185     40 CONTINUE
1186 C
1187 C*****
1188 C The following procedure is meant to determine the mesh point slopes *
1189 C P,Q, AND R. *
1190 C First the boundary slopes are determined by: *
1191 C 1....Forward difference for the first point, 1. *
1192 C 2....Backward difference for the last point, N/M. *
1193 C Next routines to set up the internal mesh point equations are *
1194 C called (i.e. PUX,QUY). These routines return the partial *
1195 C derivatives. The whole job is done below for the three necessary *
1196 C partial derivatives: P,Q AND R. *
1197 C*****
1198 C
1199     DO 50 J = 1,M
1200         P(1,J) = (U(2,J) - U(1,J)) / (X(2) - X(1))
1201         P(N,J) = (U(N,J) - U(NM1,J)) / (X(N) - X(NM1))
1202     50 CONTINUE
1203 C
1204     CALL PUX(N,M)
1205 C
1206     DO 60 I = 1,N

```

SUBROUTINE PDERIV Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
1207      Q(I,1) = (U(I,2) - U(I,1)) / (Y(2) - Y(1))
1208      Q(I,M) = (U(I,M) - U(I,MM1)) / (Y(M) - Y(MM1))
1209  60    CONTINUE
1210  C
1211      IF (M .GT. 2) THEN
1212          CALL QUY(N,M)
1213      ENDIF
1214  C
1215      DO 70 J = 1,M,MM1
1216          R(1,J) = (Q(2,J) - Q(1,J)) / (X(2) - X(1))
1217          R(N,J) = (Q(N,J) - Q(NM1,J)) / (X(N) - X(NM1))
1218  70    CONTINUE
1219  C
1220      CALL RUXY1(N,M)
1221  C
1222      IF (M .GT. 2) THEN
1223          CALL RUXY2(N,M)
1224      ENDIF
1225  C
1226      RETURN
1227      END
```

SUBROUTINE PLOTFR Compiling Options: /NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1228 C
1229 C-----
1230 C
1231 C      SUBROUTINE PLOTFR(ICT, IAN)
1232 C
1233 C*****
1234 C This subroutine plots and prints input and generated data      *
1235 C*****
1236 C
1237 C      CHARACTER*7 SCREEN(9), PLOTFR(9), PRINTR(9)
1238 C
1239 C      DATA SCREEN / 'VICPINP' , 'VICPGEN' , 'VICHNPL' , 'VI3D' ,
1240 1      , 'VITOP' , 'VIFRONT' , 'VICHORD' , 'VITWIST'
1241 2      , 'VITHICK' /
1242 C
1243 C      DATA PLOTFR / 'PLCPINP' , 'PLCPGEN' , 'PLCHNPL' , 'PL3D' ,
1244 1      , 'PLTOP' , 'PLFRONT' , 'PLCHORD' , 'PLTWIST'
1245 2      , 'PLTHICK' /
1246 C
1247 C      DATA PRINTR / 'PRCPINP' , 'PRCPGEN' , 'PRCHNPL' , 'PR3D' ,
1248 1      , 'PRTOP' , 'PRFRONT' , 'PRCHORD' , 'PRTWIST'
1249 2      , 'PRTHICK' /
1250 C
1251 C      IF (IAN .EQ. 0) THEN
1252 C
1253 C          CALL SYSTEM('CLS')
1254 C          PRINT, ' CONCENTRIC PLOT OF THE INPUT AIRFOILS.'
1255 C          CALL HRDOUT (SCREEN(1), PLOTFR(1), PRINTR(1))
1256 C
1257 C      ENDIF
1258 C
1259 C Plots for ICT = 1
1260 C
1261 C      IF (ICT .EQ. 1) THEN
1262 C
1263 C          CALL SYSTEM('CLS')
1264 C          PRINT, ' PLOT OF GENERATED AIRFOIL AT CHOSEN SPAN STATION.'
1265 C          CALL HRDOUT (SCREEN(2), PLOTFR(2), PRINTR(2))
1266 C
1267 C          CALL SYSTEM('CLS')
1268 C          PRINT, ' PLOT OF GENERATED AIRFOIL AT CHOSEN SPAN STATION.'
1269 C          PRINT, ' that has had the twist and offset removed, and'
1270 C          PRINT, ' been normalized w.r.t. its chord.'
1271 C          CALL HRDOUT (SCREEN(3), PLOTFR(3), PRINTR(3))
1272 C
1273 C      ENDIF
1274 C
1275 C Plots for ICT = 2
1276 C
1277 C      IF (ICT .EQ. 2) THEN
1278 C
1279 C          CALL SYSTEM('CLS')

```

SUBROUTINE PLOTPR Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1280      PRINT, ' A....Concentric plot of generated span station'
1281      PRINT, '      airfoils.'
1282      CALL HRDOUT (SCREEN(2),PLOT(2),PRINTR(2))
1283  C
1284      CALL SYSTEM('CLS')
1285      PRINT, ' B....3-D Plot of the generated blade.'
1286      PRINT, '      ** RESTRICTION : Total generated blade stations '
1287      PRINT, '                        MUST less than 40'
1288      CALL HRDOUT (SCREEN(4),PLOT(4),PRINTR(4))
1289  C
1290      CALL SYSTEM('CLS')
1291      PRINT, ' C....Normalized TOP view plot of the generated blade.'
1292      CALL HRDOUT (SCREEN(5),PLOT(5),PRINTR(5))
1293  C
1294      CALL SYSTEM('CLS')
1295      PRINT, ' D....Normalized FRONT view plot of the generated blade.'
1296      CALL HRDOUT (SCREEN(6),PLOT(6),PRINTR(6))
1297  C
1298      CALL SYSTEM('CLS')
1299      PRINT, ' E....The CHORD distribution of the generated blade.'
1300      CALL HRDOUT (SCREEN(7),PLOT(7),PRINTR(7))
1301  C
1302      CALL SYSTEM('CLS')
1303      PRINT, ' F....The TWIST distribution of the generated blade.'
1304      CALL HRDOUT (SCREEN(8),PLOT(8),PRINTR(8))
1305  C
1306      CALL SYSTEM('CLS')
1307      PRINT, ' G....The MAXIMUM THICKNESS distribution of the'
1308      PRINT, '                        generated blade.'
1309      CALL HRDOUT (SCREEN(9),PLOT(9),PRINTR(9))
1310  C
1311      ENDIF
1312  C
1313  C Printed Output
1314  C
1315      WRITE(6,100)
1316      READ(5,*)IA
1317  C
1318      IF (IA .EQ. 1) THEN
1319  C
1320      11      CALL SYSTEM('CLS')
1321              WRITE(6,101)
1322              READ(5,*)IWANT
1323              CALL SYSTEM('CLS')
1324              WRITE(6,102)
1325              PAUSE
1326              IF (IWANT .EQ. 1) CALL SYSTEM('COPY INPUT.DAT PRN')
1327              IF (IWANT .EQ. 2) CALL SYSTEM('COPY WTOUT.DAT PRN')
1328              IF (IWANT .EQ. 3) CALL SYSTEM('COPY NTOUT.DAT PRN')
1329              CALL SYSTEM('CLS')
1330              WRITE(6,103)
1331              READ(5,*)IAGAIN

```

SUBROUTINE PLOTPR Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1332          IF (IAGAIN .EQ. 1) GOTO 11
1333 C
1334          ENDIF
1335 C
1336          RETURN
1337 100  FORMAT(1X,,
1338      1      ' Would you like a printout of the input and/or output data?'
1339      2 ,/, ' If you want a printout, you need to have plenty of paper'
1340      3 ,/, ' and time, because the printout is several pages long.'
1341      4 ,/, ' ENTER --->      0      NO.'
1342      5 ,/, '                      1      YES.'
1343      6 ,/, ' ENTER YOUR CHOICE NUMBER HERE ---> ')
1344 101  FORMAT(1X,,
1345      1      'Which data set you would like to have a printout?'
1346      2 ,/, ' 1. Input data.'
1347      3 ,/, ' 2. Output of the generated airfoil at the chosen span'
1348      4 ,/, '    stations with twist and taper.'
1349      5 ,/, ' 3. Output of the generated airfoil at the chosen span'
1350      6 ,/, '    stations without twist and taper.'
1351      7 ,/, ' Enter your choice number here ---> ')
1352 102  FORMAT(' MAKE SURE THAT THE PRINTER IS READY.')
1353 103  FORMAT(1X,,
1354      1      ' WOULD YOU LIKE TO HAVE ANOTHER SET OF PRINTOUT?'
1355      2 ,/, ' ENTER --->      0      NO.'
1356      3 ,/, '                      1      YES.',/)
1357          END

```

SUBROUTINE PHI Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
1358 C
1359 C-----
1360 C
1361 SUBROUTINE PHI(IL,JL)
1362 C
1363 C*****
1364 C This subroutine determines the two interpolation function *
1365 C vectors, in the xi and eta directions, for each sector of *
1366 C the computational domain, see equations (2.3) and (2.4) of *
1367 C of the manual. *
1368 C*****
1369 C
1370 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1371 PARAMETER(IX = 120 , IY = 10)
1372 C
1373 COMMON /C4/X(IX),Y(IY)/C9/AL(IX),BE(IY)/C6/DX(IX),DY(IY)
1374 COMMON /C16/XIT,YIT
1375 COMMON /PHIXY/PHIX(4),PHIY(4)
1376 C
1377 IU = IL + 1
1378 JU = JL + 1
1379 PHIX(1) = XIT - X(IL)
1380 PHIY(1) = YIT - Y(JL)
1381 PHIX(2) = X(IU) - XIT
1382 PHIY(2) = Y(JU) - YIT
1383 ADX = AL(IL) * DX(IL)
1384 BDY = BE(JL) * DY(JL)
1385 QUX = DSINH(ADX) - ADX
1386 QUY = DSINH(BDY) - BDY
1387 PHIX(3) = (DX(IL)*DSINH(AL(IL)*PHIX(1))-PHIX(1)*DSINH(ADX))/QUX
1388 PHIY(3) = (DY(JL)*DSINH(BE(JL)*PHIY(1))-PHIY(1)*DSINH(BDY))/QUY
1389 PHIX(4) = (DX(IL)*DSINH(AL(IL)*PHIX(2))-PHIX(2)*DSINH(ADX))/QUX
1390 PHIY(4) = (DY(JL)*DSINH(BE(JL)*PHIY(2))-PHIY(2)*DSINH(BDY))/QUY
1391 C
1392 RETURN
1393 END
```

SUBROUTINE PUX Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1394 C
1395 C-----
1396 C
1397       SUBROUTINE PUX(N,M)
1398 C
1399 C*****
1400 C This subroutine determines the partial derivative of U with *
1401 C respect to xi at every internal node of the computational *
1402 C domain. These derivatives are stored in array P. *
1403 C*****
1404 C
1405       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1406       PARAMETER(IX = 120 , IY = 10)
1407 C
1408       COMMON /C6/DX(IX),DY(IY)
1409       COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
1410       COMMON /C15/S(IY),T(IX)/C10/V(IX),W(IY)
1411       COMMON /TRIDM/CL(IX),CM(IX),CN(IX),PQ(IX)
1412 C
1413       NM1 = N-1
1414       NM2 = N-2
1415 C
1416       DO 80 J = 1,M
1417 C
1418           DO 60 I = 2,NM1
1419               IM1 = I - 1
1420               IP1 = I + 1
1421               CM(IM1) = T(IM1) * V(IM1) + T(I) * V(I)
1422               CL(IM1) = T(IM1)
1423               CN(IM1) = T(I)
1424               BA = T(IM1) * (V(IM1)+1.D0) * (U(I,J)-U(IM1,J)) / DX(IM1)
1425               BB = T(I) * (V(I) +1.D0) * (U(IP1,J)-U(I,J)) / DX(I)
1426               PQ(IM1) = BA + BB
1427       60      CONTINUE
1428 C
1429               PQ(1) = PQ(1) - T(1) * P(1,J)
1430               PQ(NM2) = PQ(NM2) - T(NM1) * P(N,J)
1431               CALL THOMAS(NM2)
1432 C
1433           DO 70 I = 2,NM1
1434               IM1 = I - 1
1435               P(I,J) = PQ(IM1)
1436       70      CONTINUE
1437 C
1438       80      CONTINUE
1439 C
1440       RETURN
1441       END

```

SUBROUTINE QUY Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1442 C
1443 C-----
1444 C
1445 SUBROUTINE QUY(N,M)
1446 C
1447 C*****
1448 C This subroutine determines the partial derivative of U with *
1449 C respect to eta at every internal node of the computational *
1450 C domain. These derivatives are stored in array Q. *
1451 C*****
1452 C
1453 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1454 PARAMETER(IX = 120 , IY = 10)
1455 C
1456 COMMON /C6/DX(IX),DY(IY)
1457 COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
1458 COMMON /C15/S(IY),T(IX)/C10/V(IX),W(IY)
1459 COMMON /TRIDM/CL(IX),CM(IX),CN(IX),PQ(IX)
1460 C
1461 MM1 = M-1
1462 MM2 = M-2
1463 C
1464 DO 80 I = 1,N
1465 C
1466 DO 60 J = 2,MM1
1467 JM1 = J - 1
1468 JP1 = J + 1
1469 CM(JM1) = S(JM1) * W(JM1) + S(J) * W(J)
1470 CL(JM1) = S(JM1)
1471 CN(JM1) = S(J)
1472 BA = S(JM1) * (W(JM1)+1.D0) * (U(I,J)-U(I,JM1)) / DY(JM1)
1473 BB = S(J) * (W(J)+1.D0) * (U(I,JP1)-U(I,J)) / DY(J)
1474 PQ(JM1) = BA + BB
1475 60 CONTINUE
1476 C
1477 PQ(1) = PQ(1) - S(1) * Q(I,1)
1478 PQ(MM2) = PQ(MM2) - S(MM1) * Q(I,M)
1479 CALL THOMAS(MM2)
1480 C
1481 DO 70 J = 2,MM1
1482 JM1 = J - 1
1483 Q(I,J) = PQ(JM1)
1484 70 CONTINUE
1485 C
1486 80 CONTINUE
1487 C
1488 RETURN
1489 END

```


SUBROUTINE RUXY1 Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1490 C
1491 C-----
1492 C
1493       SUBROUTINE RUXY1(N,M)
1494 C
1495 C*****
1496 C This subroutine determines the mixed-derivative of U with *
1497 C respect to xi and eta at each boundary grid node of the *
1498 C computational domain, where eta has a value of 0 or 1 and *
1499 C xi does not have values of 0 or 1. These derivatives are *
1500 C stored in array R. *
1501 C*****
1502 C
1503       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1504       PARAMETER(IX = 120 , IY = 10)
1505 C
1506       COMMON /C6/DX(IX),DY(IY)
1507       COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
1508       COMMON /C15/S(IY),T(IX)/C10/V(IX),W(IY)
1509       COMMON /TRIDM/CL(IX),CM(IX),CN(IX),PQ(IX)
1510 C
1511       NM1 = N-1
1512       MM1 = M-1
1513       NM2 = N-2
1514 C
1515       DO 80 J = 1,M,MM1
1516 C
1517           DO 60 I = 2,NM1
1518               IM1 = I - 1
1519               IP1 = I + 1
1520               CM(IM1) = T(IM1) * V(IM1) + T(I) * V(I)
1521               CL(IM1) = T(IM1)
1522               CN(IM1) = T(I)
1523               BA = T(IM1) * (V(IM1)+1.D0) * (Q(I,J)-Q(IM1,J)) / DX(IM1)
1524               BB = T(I) * (V(I)+1.D0) * (Q(IP1,J)-Q(I,J)) / DX(I)
1525               PQ(IM1) = BA + BB
1526       60 CONTINUE
1527 C
1528       PQ(1) = PQ(1) - T(1) * R(1,J)
1529       PQ(NM2) = PQ(NM2) - T(NM1) * R(N,J)
1530       CALL THOMAS(NM2)
1531 C
1532       DO 70 I = 2,NM1
1533           IM1 = I - 1
1534           R(I,J) = PQ(IM1)
1535       70 CONTINUE
1536 C
1537       80 CONTINUE
1538 C
1539       RETURN
1540       END

```

SUBROUTINE RUXY2 Compiling Options:/NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1541 C
1542 C-----
1543 C
1544       SUBROUTINE RUXY2(N,M)
1545 C
1546 C*****
1547 C This subroutine determines the mixed-derivative of U with      *
1548 C respect to xi and eta at each boundary grid node of the      *
1549 C computational domain, where eta does not have the values of   *
1550 C 0 and 1. These derivatives are stored in array R.             *
1551 C*****
1552 C
1553       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1554       PARAMETER(IX = 120 , IY = 10)
1555 C
1556       COMMON /C6/DX(IX),DY(IY)
1557       COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
1558       COMMON /C15/S(IY),T(IX)/C10/V(IX),W(IY)
1559       COMMON /TRIDM/CL(IX),CM(IX),CN(IX),PQ(IX)
1560 C
1561       MM1 = M-1
1562       MM2 = M-2
1563 C
1564       DO 80 I = 1,N
1565 C
1566           DO 60 J = 2,MM1
1567               JM1 = J - 1
1568               JP1 = J + 1
1569               CM(JM1) = S(JM1) * W(JM1) + S(J) * W(J)
1570               CL(JM1) = S(JM1)
1571               CN(JM1) = S(J)
1572               BA = S(JM1) * (W(JM1)+1.D0) * (P(I,J)-P(I,JM1)) / DY(JM1)
1573               BB = S(J) * (W(J)+1.D0) * (P(I,JP1)-P(I,J)) / DY(J)
1574               PQ(JM1) = BA + BB
1575       60      CONTINUE
1576 C
1577       PQ(1) = PQ(1) - S(1) * R(I,1)
1578       PQ(MM2) = PQ(MM2) - S(MM1) * R(I,M)
1579       CALL THOMAS(MM2)
1580 C
1581       DO 70 J = 2,MM1
1582           JM1 = J - 1
1583           R(I,J) = PQ(JM1)
1584       70      CONTINUE
1585 C
1586       80      CONTINUE
1587 C
1588       RETURN
1589       END

```

SUBROUTINE SCOSTW Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1590 C
1591 C-----
1592 C
1593       SUBROUTINE SCOSTW(NAF,NNN,NMID,SPAN)
1594 C
1595 C*****
1596 C This subroutine generates new X and Y coordinates caused *
1597 C by scaling, off-setting, and twisting. *
1598 C*****
1599 C
1600       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1601       PARAMETER(IX = 120 , IY = 10)
1602 C
1603       COMMON /C2/X(IX,IY),Y(IX,IY)/C1/TH1(IY),OSX(IY),OSY(IY),CL(IY)
1604       COMMON /C3/XI(IX,IY),YI(IX,IY)/C12/Z(IY)
1605       COMMON /C18A/XU1(IX,IY),YU1(IX,IY)
1606       COMMON /C18B/XL1(IX,IY),YL1(IX,IY)
1607 C
1608       DIMENSION NNN(NAF),NMID(IY)
1609 C
1610       OPEN(UNIT = 100,FILE = 'INPUT.DAT',ACCESS = 'APPEND'
1611 1       ,FORM='FORMATTED',CARRIAGE CONTROL = 'FORTRAN',STATUS = 'OLD')
1612 C
1613       CALL CKFILE('AI1.DAT')
1614 C
1615       OPEN(UNIT = 200,FILE = 'AI1.DAT' ,STATUS = 'NEW')
1616       OPEN(UNIT = 11 ,FILE = 'AI0B.DAT',STATUS = 'UNKNOWN')
1617 C
1618       PI = DACOS(-1.D0)
1619 C
1620       DO 40 J = 1,NAF
1621           NIX = NNN(J)
1622 C
1623 C Setup the twist center.
1624 C
1625           CX = OSX(J)
1626           CY = OSY(J)
1627 C
1628 C The twist angle in degrees is converted to twist in radians.
1629 C
1630           TAU = PI * TH1(J) / 180.D0
1631           TCOS = DCOS(TAU)
1632           TSIN = DSIN(TAU)
1633 C
1634 C Offsetting the airfoil AND scaling it by multiplying by C.L.
1635 C
1636           DO 20 I = 1,NIX
1637               XI(I,J) = (X(I,J)-CX) * CL(J)
1638               YI(I,J) = (Y(I,J)-CY) * CL(J)
1639               WRITE(11,*)XI(I,J)/SPAN,YI(I,J)/SPAN,Z(J)/SPAN
1640 C
1641 C Next we will TWIST the airfoil about the offset center.
1642 C

```

SUBROUTINE SCOSTW Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1643          XR = XI(I,J)
1644          YR = YI(I,J)
1645 C
1646 C Find the new coordinates after twisting.
1647 C
1648          XI(I,J) = XR * TCOS - YR * TSIN
1649          YI(I,J) = XR * TSIN + YR * TCOS
1650 C
1651 C Next in case the user wants to view the offsetted and twisted airfoils
1652 C then we must store the airfoil data in data file AII.DAT
1653 C
1654          WRITE(200,100)XI(I,J)/SPAN,YI(I,J)/SPAN,Z(J)/SPAN
1655 C
1656 20      CONTINUE
1657 C
1658 40      CONTINUE
1659 C
1660          CLOSE(UNIT = 11)
1661 C
1662          DO 70 J = 1,NAF
1663              IM = NMID(J)
1664              NIX = NNN(J)
1665 C
1666 C Next the airfoil coordinates are put into upper and lower airfoil format.
1667 C
1668          DO 50 I = 1,NIX
1669 C
1670              IF (I .LE. IM) THEN
1671                  K = IM - I + 1
1672                  XL1(K,J) = XI(I,J)
1673                  YL1(K,J) = YI(I,J)
1674              ENDIF
1675 C
1676              IF (I .GE. IM) THEN
1677                  K = I - IM + 1
1678                  XU1(K,J) = XI(I,J)
1679                  YU1(K,J) = YI(I,J)
1680              ENDIF
1681 C
1682 50      CONTINUE
1683 C
1684 C Also the upper and lower airfoil coordinates of the airfoil,
1685 C as well as twist offset and chord data are output to a file
1686 C named INPUT.DAT
1687 C
1688          WRITE(100,105)J,NNN(J)
1689          WRITE(100,106)
1690          WRITE(100,110)CL(J),TH1(J),OSX(J),OSY(J),Z(J)
1691          WRITE(100,107)
1692 C
1693          DO 60 I = 1,IM
1694              WRITE(100,120)XU1(I,J),YU1(I,J),XL1(I,J),YL1(I,J) ~

```

SUBROUTINE SCOSTW Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
1695 60      CONTINUE
1696 C
1697 70      CONTINUE
1698 C
1699      CLOSE(UNIT = 200)
1700      CLOSE(UNIT = 100)
1701 C
1702      RETURN
1703      100 FORMAT(3(G18.9E3,2X))
1704      105 FORMAT(1H1,/,10X,'Scaled, offset, and twisted data for airfoil',I2
1705          1 ,/,10X,'
1706          2 ,/,10X,'Number of original input data points are:',I3)
1707      106 FORMAT(//,10X,'Chord-',7X,'Twist',10X,'Twist center',9X,'Span'
1708          1 ,/,10X,'Length',7X,'Angle',10X,'X',10X,'Y',8X,'Coord.')
```

1709 107 FORMAT(//,17X,'XU',12X,'YU',13X,'XL',11X,'YL',/)

1710 110 FORMAT(5X,5(2X,F10.4))

1711 120 FORMAT(10X,4(2X,E12.5))

1712 END

SUBROUTINE SECT Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1713 C
1714 C-----
1715 C
1716 SUBROUTINE SECT(N,M,ICT,SPAN)
1717 C
1718 C*****
1719 C This subroutine generates the desired cross-sectional airfoil *
1720 C data for the user. *
1721 C*****
1722 C
1723 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
1724 PARAMETER(IX = 120 , IY = 10)
1725 C
1726 CHARACTER*14 ZTEMP
1727 CHARACTER*16 ZCOORD
1728 C
1729 COMMON /C4/ZI(IX),ETA(IY)/C6/DZI(IX),DETA(IY)
1730 COMMON /C7/P(IX,IY),Q(IX,IY),R(IX,IY)/C13/U(IX,IY)
1731 COMMON /C10/V(IX),W(IY)/C9/AL(IX),BE(IY)/C12/Z(IY)
1732 COMMON /C5/XMIN(IY),XMAX(IY)/C8/IC,ZIT/C16/XIT,YIT
1733 COMMON /C20/U1(IX,IY),U2(IX,IY)
1734 COMMON /C22/XZ(IX),YZ(IX)/C25/SLE(IY),TAU
1735 COMMON /C30/NOW,RORINP(IX),CORINP(IX),TWIINP(IX),THKINP(IX)
1736 COMMON /C31/SIGMACH,SIGMATW,SIGMATH
1737 C
1738 DIMENSION XL2(IX),YL2(IX),XU2(IX),YU2(IX)
1739 DIMENSION XL3(IX),YL3(IX),XU3(IX),YU3(IX),XSTAND(57)
1740 C
1741 C Standard coordinate interpolation values
1742 C
1743 DATA (XSTAND(I),I = 1,57)/
1744 1 0.D0, .00025D0, .0005D0, .00075D0, .001D0, .0015D0,
1745 2 .002D0, .0025D0, .005D0, .01D0, .02D0, .03D0,
1746 3 .04D0, .05D0, .06D0, .07D0, .08D0, .09D0,
1747 4 .1D0, .125D0, .15D0, .175D0, .2D0, .225D0,
1748 5 .25D0, .275D0, .3D0, .325D0, .35D0, .375D0,
1749 6 .4D0, .425D0, .45D0, .475D0, .5D0, .525D0,
1750 7 .55D0, .575D0, .6D0, .625D0, .65D0, .675D0,
1751 8 .7D0, .725D0, .75D0, .775D0, .8D0, .825D0,
1752 9 .85D0, .875D0, .9D0, .925D0, .95D0, .97D0,
1753 1 .98D0, .99D0, 1.D0/
1754 C
1755 OPEN(UNIT = 700,FILE = 'WTOUT.DAT',ACCESS = 'APPEND'
1756 1 ,FORM = 'FORMATTED',CARRIAGE CONTROL = 'FORTRAN'
1757 2 ,STATUS = 'UNKNOWN')
1758 OPEN(UNIT = 750,FILE = 'NTOUT.DAT',ACCESS = 'APPEND'
1759 1 ,FORM = 'FORMATTED',CARRIAGE CONTROL = 'FORTRAN'
1760 2 ,STATUS = 'UNKNOWN')
1761 C
1762 CALL CKFILE('AIRFOIL.DAT')
1763 CALL CKFILE('NAIRFOIL.DAT')
1764 CALL CKFILE('NZCT.DAT')

```

SUBROUTINE SECT Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1765      CALL CKFILE('FVBOUND.DAT')
1766      CALL CKFILE('TVBOUND.DAT')
1767      CALL CKFILE('ZSPAN.DAT')
1768      CALL CKFILE('ZNSPAN.DAT')
1769  C
1770      OPEN(UNIT = 200, FILE = 'AIRFOIL.DAT' , STATUS = 'NEW')
1771      OPEN(UNIT = 250, FILE = 'NAIRFOIL.DAT', STATUS = 'NEW')
1772      OPEN(UNIT = 300, FILE = 'NZCT.DAT' , STATUS = 'NEW')
1773      OPEN(UNIT = 400, FILE = 'FVBOUND.DAT' , STATUS = 'NEW')
1774      OPEN(UNIT = 450, FILE = 'TVBOUND.DAT' , STATUS = 'NEW')
1775      OPEN(UNIT = 500, FILE = 'ZSPAN.DAT' , STATUS = 'NEW')
1776      OPEN(UNIT = 550, FILE = 'ZNSPAN.DAT' , STATUS = 'NEW')
1777  C
1778      CALL SYSTEM('CLS')
1779      WRITE(6,401)
1780      READ(5,*)LABEL
1781  C
1782      IF (LABEL .EQ. 3) THEN
1783          IBOUND = IX/2
1784      246      WRITE(6,402)IBOUND
1785          READ(5,*)ILABEL
1786  C
1787  C Check to see if the number entered falls within dimensional bounds.'
1788  C
1789          IF (ILABEL .LT. 24 .OR. ILABEL .GE. IBOUND) THEN
1790              WRITE(6,403)ILABEL
1791              GOTO 246
1792          ENDIF
1793  C
1794  C Check to see if the number entered is an odd number.
1795  C
1796          IF (MOD(ILABEL,2) .EQ. 0) THEN
1797              WRITE(6,404)
1798              GOTO 246
1799          ENDIF
1800  C
1801      ENDIF
1802  C
1803  C If the user wants to generate a blade
1804  C
1805          IB = 1
1806          ICT = IC
1807  C
1808  C If ICT .EQ. 1 ----> the user wants to generate an airfoil at the
1809  C                      ZIT span station.
1810  C
1811  C      ICT .EQ. 2 ----> the user wants to generate a 3D blade which
1812  C                      is represented by "NSS" equally spaced span
1813  C                      stations.
1814  C
1815          IF (ICT .EQ. 1) THEN
1816              WRITE(200,*)N,IB

```

SUBROUTINE SECT Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1817      XPL = 0.11D0
1818      YPL = 0.15D0
1819      DPL = ZIT/SPAN
1820      WRITE(ZTEMP, '(F8.4)') DPL
1821      ZCOORD = ' '//ZTEMP//''
1822      WRITE(500, *) XPL, YPL, ZCOORD
1823      XPL = 0.95D0
1824      YPL = 0.6D0
1825      WRITE(550, *) XPL, YPL, ZCOORD
1826  C
1827      ELSE
1828  C
1829      100      WRITE(6, 405)
1830      READ(5, *) NSS
1831  C
1832      IF (NSS .LT. 3) THEN
1833          WRITE(6, 406)
1834          GOTO 100
1835      ENDIF
1836  C
1837      ZIT = Z(1)
1838      DZIT = (Z(M) - Z(1)) / FLOAT(NSS - 1)
1839      WRITE(200, *) N, NSS
1840  C
1841      ENDIF
1842  C
1843  C Generate the airfoil section data at the ZIT span station
1844  C
1845      WRITE(6, 104)
1846  C
1847  C Initialize sums of the standard deviations for the chord, twist,
1848  C and thickness distributions, respectively.
1849  C
1850      SUMCH = 0.0D0
1851      SUMTW = 0.0D0
1852      SUMTH = 0.0D0
1853  C
1854  C First determine ETA at particular ZIT span station
1855  C
1856      130      YIT = (ZIT - Z(1)) / (Z(M) - Z(1))
1857  C
1858  C Also determine the leading edge S value at this ZIT span station
1859  C
1860      SIGMAL = 20.0D0
1861      CALL SPT11D(M, ETA, SLE, SIGMAL, YIT, SMID)
1862  C
1863  C Next determine all the X values, N of them, that represent
1864  C the ZIT span station
1865  C
1866      CALL GETX(N, M, TAU, SMID)
1867  C
1868  C Next determine all the Y values, N of them, that represent
1869  C the ZIT span station

```


SUBROUTINE SECT Compiling Options: /NO/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1870 C
1871     CALL GETY(N,M,TAU,SMID)
1872 C
1873 C Determine the chord and twist of this ZIT span station. Also
1874 C determine the chord/span ratio and the r/R or Z/SPAN ratio.
1875 C These values are dumped into a data file named NZCT.DAT.
1876 C
1877 C First compute midpoint of trailing-edge base
1878 C
1879     XTE = 0.5D0 * (XZ(1) + XZ(N))
1880     YTE = 0.5D0 * (YZ(1) + YZ(N))
1881 C
1882 C Next, find the most forward leading-edge point and the longest
1883 C chord.
1884 C
1885     CHORD = 0.D0
1886 C
1887     DO 5 I = 1,N
1888         DX = XZ(I) - XTE
1889         DY = YZ(I) - YTE
1890         DIST = DSQRT(DX * DX + DY * DY)
1891 C
1892         IF (DIST .GT. CHORD) THEN
1893             CHORD = DIST
1894             IA = I
1895             XLE = XZ(I)
1896             YLE = YZ(I)
1897         ENDIF
1898 C
1899     5 CONTINUE
1900 C
1901     DX = XTE - XLE
1902     DY = YTE - YLE
1903     TRAD = DATAN2 (DY , DX)
1904     TWIST = TRAD * 180.D0 / DACOS(-1.D0)
1905     ROR = ZIT / SPAN
1906     CHORDN = CHORD / SPAN
1907 C
1908 C Determine the twist center (TWISTX , TWISTY) which will make the
1909 C leading-edge coordinate (0,0).
1910 C
1911     CT = DCOS (TRAD)
1912     ST = DSIN (TRAD)
1913     TWISTX = -XZ(IA) * CT - YZ(IA) * ST
1914     TWISTY = XZ(IA) * ST - YZ(IA) * CT
1915 C
1916     XMA = XZ(IA)
1917     XMI = XZ(IA)
1918     YMIN = YZ(IA)
1919     YMAX = YZ(IA)
1920 C
1921 C The following calculations are to arrange the upper and lower
1922 C airfoil x and y coordinates into a manner similar to that of

```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

1923 C airfoil tables. Also output to data files NZCT.DATA,
1924 C AIRFOIL.DAT, WTOUT.DAT, and NTOUT.DAT
1925 C
1926 C      G = SPAN
1927 C
1928 C      DO 210 I = 1,N
1929 C
1930 C          IF (I .LE. IA) THEN
1931 C              K = IA - I + 1
1932 C              XL2(K) = XZ(I)
1933 C              YL2(K) = YZ(I)
1934 C          ENDIF
1935 C
1936 C          IF (I .GE. IA) THEN
1937 C              K = I - IA + 1
1938 C              XU2(K) = XZ(I)
1939 C              YU2(K) = YZ(I)
1940 C          ENDIF
1941 C
1942 C          IF (XZ(I) .GT. XMA) XMA = XZ(I)
1943 C          IF (XZ(I) .LT. XMI) XMI = XZ(I)
1944 C          IF (YZ(I) .GT. YMAX) YMAX = YZ(I)
1945 C          IF (YZ(I) .LT. YMIN) YMIN = YZ(I)
1946 C
1947 C          WRITE(200,500)XZ(I)/G, YZ(I)/G, ZIT/G
1948 C
1949 C      210 CONTINUE
1950 C
1951 C      IF (ICT .EQ. 2) THEN
1952 C          WRITE(400,500)ROR, YMAX/G, YMIN/G
1953 C          WRITE(450,500)ROR, XMA /G, XMI/G
1954 C      ENDIF
1955 C
1956 C      WRITE(700,105)ZIT, ZIT
1957 C      WRITE(700,510)CHORDN, TWIST, TWISTX, TWISTY, ZIT/SPAN, SPAN
1958 C      WRITE(700,106)
1959 C
1960 C      DO 220 I = 1,IA
1961 C          WRITE(700,520)XU2(I)/G, YU2(I)/G, XL2(I)/G, YL2(I)/G
1962 C      220 CONTINUE
1963 C
1964 C      Also here the airfoil has its twist and offset removed so that
1965 C      the airfoil data without offset and twist can be output.
1966 C
1967 C          OFFX = TWISTX/CHORD
1968 C          OFFY = TWISTY/CHORD
1969 C
1970 C      DO 230 I = 1,N
1971 C          XNT = ( XZ(I) * CT + YZ(I) * ST) / CHORD + OFFX
1972 C          YNT = (-XZ(I) * ST + YZ(I) * CT) / CHORD + OFFY
1973 C          XZ(I)=XNT
1974 C          YZ(I)=YNT

```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
1975 230 CONTINUE
1976 C
1977      DO 231 I = 1, IA
1978          IBACK = IA - I + 1
1979          XL2(IBACK) = XZ(I)
1980          YL2(IBACK) = YZ(I)
1981 231 CONTINUE
1982 C
1983      NPTL = IA
1984      IUPPER = 1
1985 C
1986      DO 232 I = IA, N
1987          XU2(IUPPER) = XZ(I)
1988          YU2(IUPPER) = YZ(I)
1989          IUPPER = IUPPER + 1
1990 232 CONTINUE
1991 C
1992      NPTU = N - IA + 1
1993 C
1994      IF (LABEL .EQ. 1) THEN
1995          NUPPER = NPTU
1996 C
1997          DO 248 I = 1, NUPPER
1998              XU3(I) = XU2(I)
1999              YU3(I) = YU2(I)
2000 248 CONTINUE
2001 C
2002          NLOWER = NPTL
2003 C
2004          DO 249 I = 1, NLOWER
2005              XL3(I) = XL2(I)
2006              YL3(I) = YL2(I)
2007 249 CONTINUE
2008 C
2009      ELSE IF (LABEL .EQ. 2) THEN
2010          NUPPER = 57
2011          NLOWER = 57
2012          SIG = -1.D0
2013 C
2014          DO 245 I = 1, 57
2015              CALL SPT11D(NPTU, XU2, YU2, SIG, XSTAND(I), YSTAND)
2016              XU3(I) = XSTAND(I)
2017              YU3(I) = YSTAND
2018              CALL SPT11D(NPTL, XL2, YL2, SIG, XSTAND(I), YSTAND)
2019              XL3(I) = XSTAND(I)
2020              YL3(I) = YSTAND
2021 245 CONTINUE
2022 C
2023      ELSE IF (LABEL .EQ. 3) THEN
2024          NUPPER = ILABEL
2025          NLOWER = ILABEL
2026          SIG = -1.D0
```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2027      DXOUT = 1.D0/FLOAT(ILABEL-1)
2028      XXX = 0.D0
2029 C
2030      DO 247 I = 1,ILABEL
2031          IF (I .EQ. ILABEL) XXX = 1.D0
2032          CALL SPT11D(NPTU,XU2,YU2,SIG,XXX,YYY)
2033          XU3(I) = XXX
2034          YU3(I) = YYY
2035          CALL SPT11D(NPTL,XL2,YL2,SIG,XXX,YYY)
2036          XL3(I) = XXX
2037          YL3(I) = YYY
2038          XXX = XXX + DXOUT
2039      247 CONTINUE
2040 C
2041      ENDIF
2042 C
2043 C Determine the max. thickness THMAX
2044 C
2045      THMAX = 0.0D0
2046      IXDATA = MAX0(NUPPER,NLOWER)
2047 C
2048      DO 240 I = 1,IXDATA
2049          GTH = DABS(YU3(I) - YL3(I))
2050          IF (GTH .GT. THMAX) THMAX = GTH
2051      240 CONTINUE
2052 C
2053      WRITE(750,107)ZIT
2054      WRITE(750,511)CHORDN, TWIST, THMAX, OFFX, OFFY, ZIT/SPAN, SPAN
2055      WRITE(750,108)CHORD
2056 C
2057      DO 242 I = 1,IXDATA
2058          WRITE(750,520)XU3(I), YU3(I), XL3(I), YL3(I)
2059      242 CONTINUE
2060 C
2061 C Write airfoil data to data file NAIRFOIL.DAT
2062 C
2063      DO 251 I = 1,NLOWER
2064          IBACK = NLOWER - I + 1
2065          XZ(I) = XL3(IBACK)
2066          YZ(I) = YL3(IBACK)
2067      251 CONTINUE
2068 C
2069      ISTART = 1
2070      IF (YL3(1) .EQ. YU3(1)) ISTART = 2
2071      IP1 = NLOWER
2072 C
2073      DO 252 J = ISTART,NUPPER
2074          IP1 = IP1 + 1
2075          XZ(IP1) = XU3(J)
2076          YZ(IP1) = YU3(J)
2077      252 CONTINUE
2078 C
2079      DO 253 I = 1,IP1

```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2080          WRITE(250,500)XZ(I), YZ(I), ZIT/G
2081 253 CONTINUE
2082 C
2083 C IF the choice is to generate a blade then
2084 C
2085          IF (ICT .EQ. 2) THEN
2086 C
2087 C Also here the standard deviations of the chord, twist, and
2088 C thickness are determined.
2089 C
2090          WRITE(300,500)ROR, CHORDN, TWIST, THMAX * 100.0D0
2091 C
2092          IF (NOW .GT. 3) THEN
2093              CALL SPT11D(NOW,RORINP,CORINP,SIGMACH,ROR,CORACT)
2094              CALL SPT11D(NOW,RORINP,TWIINP,SIGMATW,ROR,TWIACT)
2095              CALL SPT11D(NOW,RORINP,THKINP,SIGMATH,ROR,THKACT)
2096 C
2097          ELSE
2098 C
2099              CALL SPT11D(NOW,RORINP,CORINP,-10. ,ROR,CORACT)
2100              CALL SPT11D(NOW,RORINP,TWIINP,-10. ,ROR,TWIACT)
2101              CALL SPT11D(NOW,RORINP,THKINP,-10. ,ROR,THKACT)
2102 C
2103          ENDIF
2104 C
2105          DCH = (CHORDN - CORACT) * 100.0D0
2106          DTW = TWIST-TWIACT
2107          DTH = (THMAX * CHORDN - THKACT * CORACT) * 100.0D0
2108          SUMCH = SUMCH + DCH * DCH
2109          SUMTW = SUMTW + DTW * DTW
2110          SUMTH = SUMTH + DTH * DTH
2111 C
2112 C Increment ZIT and IB, the span station counter,
2113 C and continue with the loop.
2114 C
2115          ZIT = ZIT + DZIT
2116          IB = IB + 1
2117          IF (IB .LE. NSS) GOTO 130
2118 C
2119 C Next interpolate input chord, twist, maximum
2120 C thickness distribution for graphic output
2121 C
2122          CALL CKFILE('CTTINP.DAT')
2123          OPEN(UNIT = 900,FILE = 'CTTINP.DAT',STATUS = 'NEW')
2124          ZIT = Z(1)
2125          DZIT = (Z(M) - Z(1))/100.D0
2126 C
2127          DO 300 I = 1 , 101
2128              ROR = ZIT/SPAN
2129 C
2130              IF (NOW .GT. 3) THEN
2131 C
2132                  CALL SPT11D(NOW,RORINP,CORINP,SIGMACH,ROR,CORACT)

```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2133          CALL SPT11D(NOW,RORINP,TWIINP,SIGMATW,ROR,TWIACT)
2134          CALL SPT11D(NOW,RORINP,THKINP,SIGMATH,ROR,THKACT)
2135 C
2136          ELSE
2137 C
2138          CALL SPT11D(NOW,RORINP,CORINP,-10. ,ROR,CORACT)
2139          CALL SPT11D(NOW,RORINP,TWIINP,-10. ,ROR,TWIACT)
2140          CALL SPT11D(NOW,RORINP,THKINP,-10. ,ROR,THKACT)
2141 C
2142          ENDIF
2143 C
2144          WRITE(900,*)ROR, CORACT, TWIACT, THKACT * 100.DO
2145          ZIT = ZIT + DZIT
2146 300      CONTINUE
2147 C
2148          CLOSE(UNIT = 900)
2149 C
2150          CALL CKFILE('CTTIN.DAT')
2151          OPEN(UNIT = 900,FILE = 'CTTIN.DAT',STATUS = 'NEW')
2152 C
2153          DO 301 I = 1,NOW
2154              WRITE(900,*)RORINP(I),CORINP(I),TWIINP(I),THKINP(I)*100.DO
2155 301      CONTINUE
2156 C
2157          CLOSE(UNIT = 900)
2158 C
2159 C Next the standard deviations are determined.
2160 C
2161          SDCH = DSQRT(SUMCH/NSS)
2162          SDTW = DSQRT(SUMTW/NSS)
2163          SDTH = DSQRT(SUMTH/NSS)
2164          WRITE(6,407)SDCH, SDTW, SDTH
2165          PAUSE
2166          ENDIF
2167 C
2168          CLOSE(UNIT = 200)
2169          CLOSE(UNIT = 250)
2170          CLOSE(UNIT = 300)
2171          CLOSE(UNIT = 400)
2172          CLOSE(UNIT = 450)
2173          CLOSE(UNIT = 500)
2174          CLOSE(UNIT = 550)
2175          CLOSE(UNIT = 600)
2176          CLOSE(UNIT = 610)
2177          CLOSE(UNIT = 620)
2178          CLOSE(UNIT = 700)
2179          CLOSE(UNIT = 750)
2180 C
2181          RETURN
2182 104      FORMAT(1X,/,
2183 1          ' PLEASE WAIT.....'
2184 2          ,//, ' CALCULATIONS UNDER WAY .....')
```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2185 105 FORMAT(1H1,/,14X,'***** OUTPUT DATA *****'
2186 1 ,/,10X,'DATA FOR SPAN STATION = ',F7.4,' ARE:'
2187 2 ,/,10X,'
2188 3 ,/,10X,'These are the actual coordinates of the airfoil'
2189 4 ,/,10X,'at span station',F7.4
2190 5 ,/,10X,' WITH'
2191 6 ,/,8X,'Chord-',6X,'Twist',10X,'Twist Center',10X,'Z/R',8X,'Span'
2192 7 ,/,8X,'Length',6X,'Angle',9X,'X',12X,'Y',8X,'Coord.',7X,'R')
2193 C
2194 106 FORMAT(/,10X,'The upper and lower airfoil coordinates shown'
2195 1 , ' below have been',/,10X,'normalized w.r.t. span, R.'
2196 2 ,/,20X,'XU',12X,'YU',13X,'XL',11X,'YL',/)
2197 C
2198 107 FORMAT(1H1,/,14X,'***** OUTPUT DATA *****'
2199 1 ,/,10X,'DATA FOR SPAN STATION',F7.4,' ARE:'
2200 2 ,/,10X,'
2201 3 ,/,6X,'Chord-',4X,'Twist',4X,'Thickness',5X,'Twist Center'
2202 4 ,6X,'Z/R',6X,'Span'
2203 5 ,/,6X,'Length',4X,'Angle',16X,'X/C',8X,'Y/C',5X,'Coord.',6X,'R')
2204 C
2205 108 FORMAT(/,10X,'The following upper and lower airfoil coordinates'
2206 1 , ' are for the',/,10X,'airfoil, with no twist, no offset,'
2207 2 , ' and are normalized w.r.t.',/,10X,'chord =',F10.6
2208 3 ,/,20X,'XU',12X,'YU',12X,'XL',11X,'YL',/)
2209 C
2210 401 FORMAT(
2211 1 ' You are now given the following three options on how to'
2212 2 ,/, ' specify the output X-coordinates of the generated airfoil,'
2213 3 ,/, ' which has its twist and offset removed, at the chosen span'
2214 4 ,/, ' station.'
2215 5 ,/, ' Option 1.... The X-coordinates will be specified'
2216 6 ,/, ' such that there are more points around the'
2217 7 ,/, ' leading edge. This feature depends on the'
2218 8 ,/, ' stretching parameter previously used.'
2219 9 ,/, ' Option 2.... The X-coordinates will be specified'
2220 1 ,/, ' such that they correspond to the'
2221 2 ,/, ' STANDARD 57 X coordinates; see manual.'
2222 3 ,/, ' Option 3.... The X-coordinates will be specified'
2223 4 ,/, ' such that they are regularly spaced. The'
2224 5 ,/, ' spacing depends on the number of points'
2225 6 ,/, ' (to be input next) used to label the X-'
2226 7 ,/, ' coordinates.'
2227 8 ,/, ' Enter your choice number here ---> ')
2228 C
2229 402 FORMAT(
2230 1 ' Enter the number of points you want to specify the'
2231 2 ,/, ' X-coordinates.'
2232 3 ,/, ' An ODD integer number between 24 and 'I3,' ---> ')
2233 C
2234 403 FORMAT(' The number you just typed is out of bound.',/,
2235 1 ' Please try AGAIN !')
2236 C
2237 404 FORMAT('.....YOU ENTERED AN EVEN NUMBER.....')

```

SUBROUTINE SECT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
2238      1      ,/, ' **** YOU DID NOT ENTER AN ODD NUMBER. TRY AGAIN ! ****' )
2239 C
2240 405 FORMAT(' HOW MANY span stations do you want your blade'
2241      1      ,/, ' represented by ? '
2242      2      ,/, ' ENTER a number greater than 3 '
2243      3      ,/, ' (less than 40 for viewing 3D plot) ---> ')
2244 C
2245 406 FORMAT(' The number of span stations for representing the'
2246      /      ,/, ' blade must be greater than 3. Please try AGAIN !')
2247 C
2248 407 FORMAT(' The standard deviations of ',/,
2249      /      ' Chord      = ',D15.6,/,
2250      /      ' Twist  angle = ',D15.6,/,
2251      /      ' Thickness   = ',D15.6)
2252 C
2253 500 FORMAT(4(F9.5,2X))
2254 510 FORMAT(2X,6(2X,F10.4))
2255 511 FORMAT(2X,7(2X,F8.4))
2256 520 FORMAT(10X,4(2X,F12.5))
2257      END
```


SUBROUTINE SORT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2258 C
2259 C-----
2260 C
2261 C          SUBROUTINE SORT(N,M,IL,JL)
2262 C
2263 C*****
2264 C This subroutine finds the computational domain grid sector in *
2265 C which the point (xi,eta) lies. It returns the (xi,eta) *
2266 C coordinates of the bottom left corner node of the sector. *
2267 C*****
2268 C
2269 C          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2270 C          PARAMETER(IX = 120 , IY = 10)
2271 C
2272 C          COMMON /C4/X(IX),Y(IY)/C16/XIT,YIT
2273 C
2274 C          NM1 = N - 1
2275 C          MM1 = M - 1
2276 C
2277 10      DO 20 I = 2,N
2278         IM1 = I - 1
2279         IF (X(I)-XIT) 20,20,30
2280 20      CONTINUE
2281 C
2282 C          IL = NM1
2283 C
2284 30      IF (X(IM1) .LE. XIT .OR. XIT .LE. X(I)) GOTO 40
2285 C
2286 C          WRITE(6,200)
2287 C          PAUSE
2288 C          GOTO 50
2289 C
2290 40      IL = IM1
2291 C
2292 50      DO 60 J = 2,M
2293         JM1 = J - 1
2294         IF (Y(J)-YIT) 60,60,70
2295 60      CONTINUE
2296 C
2297 C          JL = MM1
2298 C
2299 70      IF (Y(JM1) .LE. YIT .OR. YIT .LE. Y(J)) GOTO 80
2300 C
2301 C          WRITE(6,201)
2302 C          PAUSE
2303 C          GOTO 90
2304 C
2305 80      JL = JM1
2306 C
2307 90      RETURN
2308 200      FORMAT(1X,' ZI COORDINATE OF POINT FALLS OUTSIDE THE BOUNDS.')
```

F77L - Lahey FORTRAN 77, Version 2.22 03/21/91 22:23:03

Page 55

SUBROUTINE SORT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

2310

END

```
2311 C
2312 C-----
2313 C
2314 C      SUBROUTINE SPT11D(M,X,Y,SIGMA,XIT,YIT)
2315 C
2316 C*****
2317 C This subroutine calls SUBROUTINE CURVE1 and FUNCTION CURVE2 to *
2318 C perform a 1D tension spline interpolation of a given curve at a *
2319 C given point. *
2320 C*****
2321 C
2322 C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2323 C      PARAMETER(IX = 120 , IY = 10)
2324 C
2325 C      DIMENSION X(M),Y(M),YP(IX),TEMP(IX)
2326 C
2327 C      CALL CURVE1(M,X,Y,SLP1,SLPN,YP,TEMP,SIGMA)
2328 C      IT = 1
2329 C      YIT = CURVE2(XIT,M,X,Y,YP,SIGMA,IT)
2330 C
2331 C      RETURN
2332 C      END
```

SUBROUTINE THOMAS Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2333 C
2334 C-----
2335 C
2336 SUBROUTINE THOMAS(N)
2337 C
2338 C*****
2339 C This subroutine solves tridiagonal system by the THOMAS' *
2340 C algorithm. *
2341 C*****
2342 C
2343 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2344 PARAMETER(IX = 120)
2345 C
2346 COMMON /TRIDM/CL(IX),CM(IX),CN(IX),PQ(IX)
2347 C
2348 NM1 = N - 1
2349 C
2350 IF (N .LE. 1) THEN
2351 PQ(1) = PQ(1) / CM(1)
2352 RETURN
2353 ENDIF
2354 C
2355 C*****
2356 C Establish upper triangular matrix *
2357 C*****
2358 C
2359 DO 20 I = 2,N
2360 IM1 = I - 1
2361 CL(I) = CL(I) / CM(IM1)
2362 CM(I) = CM(I) - CL(I) * CN(IM1)
2363 PQ(I) = PQ(I) - CL(I) * PQ(IM1)
2364 20 CONTINUE
2365 C
2366 C*****
2367 C Back substitution *
2368 C*****
2369 C
2370 PQ(N) = PQ(N) / CM(N)
2371 C
2372 DO 30 I = NM1,1,-1
2373 PQ(I) = (PQ(I) - CN(I) * PQ(I+1)) / CM(I)
2374 30 CONTINUE
2375 C
2376 RETURN
2377 END

```

SUBROUTINE TINPUT Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2378 C
2379 C-----
2380 C
2381 SUBROUTINE TINPUT(NAF,NDF,NNN,NMID,SPAN,IDC,N1,SIGMA,IU)
2382 C
2383 C*****
2384 C This subroutine performs the "TERMINAL" input mode option *
2385 C*****
2386 C
2387 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2388 PARAMETER(IX = 120 , IY = 10)
2389 C
2390 CHARACTER*1 SCALE
2391 CHARACTER*12 NAME(IY),FNAME,DNAME
2392 C
2393 COMMON /C1/TH1(IY),OSX(IY),OSY(IY),CL(IY)/C2/XX(IX,IY),YY(IX,IY)
2394 COMMON /C12/ZZ(IY)/C11/ALPHA,BETA
2395 COMMON /C14A/XU(IX,IY),XL(IX,IY)
2396 COMMON /C14B/YU(IX,IY),YL(IX,IY)
2397 COMMON /C30/NOW,ROR(IX),COR(IX),TWI(IX),THK(IX)
2398 COMMON /C31/SIGMACH,SIGMATW,SIGMATH
2399 C
2400 DIMENSION NDF(IY),NNN(IY)
2401 DIMENSION ZE(IX),CHORD(IX),VAR(8),THKN(IX),NMID(IY)
2402 C
2403 DATA VAR/ 300.0D0,0.0D0,5.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0/
2404 C
2405 12 WRITE(6,93)
2406 READ(IU,*)IDC
2407 C
2408 IF (IDC .LT. 1 .OR. IDC .GT. 4) THEN
2409 WRITE(6,103)
2410 PAUSE
2411 GOTO 12
2412 ENDIF
2413 C
2414 IF (IDC .EQ. 2) DNAME = 'CTE1.DAT'
2415 C
2416 IF (IDC .EQ. 3) DNAME = 'CTE2.DAT'
2417 C
2418 IF (IDC .EQ. 4) THEN
2419 WRITE(6,94)
2420 READ(IU,*)DNAME
2421 ENDIF
2422 C
2423 IF (IDC .GT. 1) THEN
2424 OPEN(UNIT = 10 , FILE = DNAME , STATUS = 'OLD' , IOSTAT = IERR)
2425 IF (IERR .GT. 0) THEN
2426 WRITE(6,95)DNAME
2427 PAUSE
2428 GO TO 12
2429 ENDIF

```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
2430      ENDIF
2431  C
2432      CALL CKFILE('CTEX1.DAT')
2433      OPEN(UNIT = 20 , FILE = 'CTEX1.DAT', STATUS = 'NEW')
2434  C
2435  C Step 2
2436  C
2437  2      WRITE(6,96) IY
2438      READ(IU,*) NAF
2439  C
2440      IF (NAF .LT. 2 .OR. NAF .GT. IY) THEN
2441          WRITE(6,97) IY
2442          PAUSE
2443          GOTO 2
2444      ENDIF
2445  C
2446      WRITE(6,98)
2447      READ(IU,*) SPAN
2448  C
2449      CALL SYSTEM('CLS')
2450      WRITE(6,99)
2451      READ(IU,*) SCALE
2452  C
2453      IF (IDC .GT. 1) THEN
2454          READ(10,*) NOW
2455  C
2456          DO 8 I = 1, NOW
2457              READ(10,*) ROR(I), COR(I), TWI(I), THK(I)
2458              WRITE(20,205) ROR(I), COR(I), TWI(I), THK(I) * 100.0D0
2459              ZE(I) = ROR(I) * SPAN
2460              CHORD(I) = COR(I) * SPAN
2461  8      CONTINUE
2462  C
2463          CLOSE(UNIT = 10)
2464      ENDIF
2465  C
2466      IF (IDC.EQ.1) NOW = NAF
2467      MAX = 0
2468      OPEN(UNIT = 8 , FILE = 'AI0A.DAT' , STATUS = 'UNKNOWN')
2469  C
2470  C Step 3
2471  C
2472      DO 100 K = 1, NAF
2473  C
2474  9      CALL SYSTEM('CLS')
2475          WRITE(6,302) K
2476  C
2477          READ(IU,*) NDF(K)
2478  C
2479          IF (NDF(K) .GT. 7) THEN
2480              WRITE(6,103)
2481              PAUSE
```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2482          GOTO 9
2483      ENDIF
2484  C
2485          IF (NDF(K) .EQ. 1) NAME(K) = 'S806A.DAT'
2486          IF (NDF(K) .EQ. 2) NAME(K) = 'S805A6.DAT'
2487          IF (NDF(K) .EQ. 3) NAME(K) = 'S805A.DAT'
2488          IF (NDF(K) .EQ. 4) NAME(K) = 'S805A7.DAT'
2489          IF (NDF(K) .EQ. 5) NAME(K) = 'S807.DAT'
2490          IF (NDF(K) .EQ. 6) NAME(K) = 'S808.DAT'
2491          IF (NDF(K) .EQ. 7) THEN
2492              WRITE(6,108)
2493              READ(IU,*)NAME(K)
2494      ENDIF
2495  C
2496          FNAME = NAME(K)
2497          LUN = 10
2498          OPEN(UNIT = LUN, FILE = FNAME , STATUS = 'OLD' , IOSTAT = IERR)
2499          IF (IERR .GT. 0) THEN
2500              WRITE(6,109)
2501              PAUSE
2502              GOTO 9
2503      ENDIF
2504  C
2505          WRITE(6,110) FNAME
2506  C
2507          READ(LUN,*)NU
2508          READ(LUN,*)NL
2509          N = NU + NL - 1
2510          NNN(K) = N
2511  C
2512          DO 50 I = 1,NU
2513              READ(LUN,*)XU(I,K),YU(I,K)
2514      50 CONTINUE
2515  C
2516          DO 60 I = 1,NL
2517              READ(LUN,*)XL(I,K),YL(I,K)
2518      60 CONTINUE
2519  C
2520          CLOSE(UNIT = LUN)
2521  C
2522          WRITE (6,305) FNAME , K
2523          PAUSE
2524          CALL SYSTEM('CLS')
2525  C
2526          IF (IDC .GT. 1) THEN
2527              SP1 = SPAN * ROR(1) - 1.0D-10
2528              SP2 = SPAN * ROR(NOW) + 1.0D-10
2529      ENDIF
2530  C
2531          IF (IDC .EQ. 1) THEN
2532              SP1 = 0.D0
2533              SP2 = SPAN

```

SUBROUTINE TINPUT Compiling Options: /N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2534      ENDIF
2535  C
2536      IF (K .EQ. 1) THEN
2537  61      WRITE(6,303)SP1,SP2
2538          WRITE(6,304)K
2539          READ(IU,*)ZZ(K)
2540  C
2541          IF (ZZ(K) .LT. SP1 .OR. ZZ(K) .GT. SP2) GOTO 61
2542  C
2543          IF (ZZ(K) .EQ. SP2 .AND. K .NE. NAF) THEN
2544              WRITE(6,111)
2545              PAUSE
2546              GOTO 61
2547          ENDIF
2548  C
2549      ELSE
2550  C
2551          KM1 = K-1
2552  62      WRITE(6,303)ZZ(KM1),SP2
2553          WRITE(6,304)K
2554          READ(IU,*)ZZ(K)
2555  C
2556          IF (ZZ(K) .LE. ZZ(KM1) .OR. ZZ(K) .GT. SP2) GOTO 62
2557  C
2558          IF (ZZ(K) .EQ. SP2 .AND. K .NE. NAF) THEN
2559              WRITE(6,111)
2560              PAUSE
2561              GOTO 62
2562          ENDIF
2563  C
2564      ENDIF
2565  C
2566      IF (IDC .GT. 1) THEN
2567          ZIT = ZZ(K)
2568          SIGMACH = -1.D0
2569          SIGMATW = -1.D0
2570          SIGMATH = -1.D0
2571          CALL SPT11D(NOW , ZE , CHORD , SIGMACH , ZIT , CHL )
2572          CALL SPT11D(NOW , ZE , TWI , SIGMATW , ZIT , TWA )
2573          CALL SPT11D(NOW , ZE , THK , SIGMATH , ZIT , THICK)
2574          CL(K) = CHL
2575          TH1(K) = TWA
2576          THKN(K) = THICK
2577  C
2578      ELSE
2579  C
2580          WRITE(6,112)
2581          READ(IU,*)TH1(K)
2582  C
2583  63      WRITE(6,113)K
2584          READ(IU,*)CL(K)
2585  C
2586          IF (CL(K) .LE. 0.D0) THEN

```


SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
2587             WRITE(6,114)
2588             GOTO 63
2589         ENDIF
2590     C
2591     64         WRITE(6,115)
2592             READ(IU,*)THKN(K)
2593     C
2594             IF (THKN(K) .GT. 1.0D0) THEN
2595                 WRITE(6,116)
2596                 GOTO 64
2597             ENDIF
2598     C
2599             THICK = THKN(K)
2600     C
2601             ROR(K) = ZZ(K)/SPAN
2602             COR(K) = CL(K)/SPAN
2603             TWI(K) = TH1(K)
2604             THK(K) = THKN(K)
2605     C
2606             WRITE(20,205)ROR(K) , COR(K) , TH1(K) , THKN(K)*100.0D0
2607     C
2608         ENDIF
2609     C
2610         CALL SYSTEM('CLS')
2611     C
2612     13         WRITE(6,117)
2613             READ(IU,*)TWCTR
2614     C
2615         CALL SYSTEM('CLS')
2616     C
2617             IF (TWCTR .EQ. 1) THEN
2618                 WRITE(6,118)
2619                 READ(IU,*)OSX(K)
2620                 WRITE(6,119)
2621                 READ(IU,*)OSY(K)
2622             ENDIF
2623     C
2624             IF (TWCTR .EQ. 2) THEN
2625                 TWCTRX = 0.3D0
2626                 SIG    = -1.0D0
2627                 CALL SPT11D(NU,XU,YU,SIG,TWCTRX,TWCTYU)
2628                 CALL SPT11D(NL,XL,YL,SIG,TWCTRX,TWCTYL)
2629                 OSX(K) = TWCTRX
2630                 OSY(K) = 0.5D0 * (TWCTYU-TWCTYL)
2631             ENDIF
2632     C
2633         CALL SYSTEM('CLS')
2634     C
2635     C SET UP FOR SCALING
2636     C
2637             IF (SCALE .EQ. 'Y' .OR. SCALE .EQ. 'y') THEN
2638                 CALL CKFILE('AFXYIN')
```

SUBROUTINE TINPUT
Source file Listing

Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX

```

2639      OPEN (15 , FILE = 'AFXIYN' , STATUS = 'NEW')
2640      WU = 1.0D0
2641      NU1 = NU-1
2642      VAR(1) = 0.D0
2643      WRITE(15,400)CHARNB(NAME(K))
2644      WRITE(15,401)(VAR(I),I = 1,8)
2645      WRITE(15,402)NU
2646      YU(NU,K) = (YU(NU,K) + YU(NU1,K)) / 2.0D0
2647 C
2648      DO 66 I = 1,NU
2649          WRITE(15,403)XU(I,K),YU(I,K),WU
2650 66      CONTINUE
2651 C
2652      WRITE(15,402)NL
2653 C
2654      DO 67 I = 1,NL
2655          WRITE(15,403)XL(I,K),YL(I,K),WU
2656 67      CONTINUE
2657 C
2658      CLOSE(UNIT = 15)
2659 C
2660      CALL CKFILE('SCXYIN')
2661      CALL SYSTEM('CLS')
2662 C
2663      WRITE(6,191)
2664 C
2665      CALL SYSTEM('SMOOTH2')
2666 C
2667      OPEN (UNIT = 16 , FILE = 'SCALEINP.DAT' , ACCESS = 'APPEND'
2668 /      , STATUS = 'OLD')
2669 C
2670      XNT = 1.0D0
2671      WRITE(16,402)XNT
2672      WRITE(16,402)THICK
2673      CLOSE(UNIT = 16)
2674 C
2675      CALL SYSTEM('RENAME SCALEINP.DAT SCXYIN')
2676 C
2677      CALL SYSTEM('SCALE1')
2678 C
2679 C CONCLUSION OF SCALING
2680 C
2681 C
2682 C -- RETRIEVE SCALED DATA FROM DATA FILES --
2683 C
2684      OPEN (UNIT = 17 , FILE = 'XUXLSS.DAT' , STATUS = 'OLD')
2685      READ(17,*)NU
2686      READ(17,*)(XU(I,K),YU(I,K),I = 1,NU)
2687      READ(17,*)NL
2688      READ(17,*)(XL(I,K),YL(I,K),I = 1,NL)
2689      CLOSE(UNIT = 17)
2690 C
2691      ENDIF

```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
2692 C
2693 C CONTINUE AND FORM THE AIRFOIL
2694 C
2695 C      N = NU + NL - 1
2696 C      NNN(K) = N
2697 C      NMID(K) = NL
2698 C      YU(NU,K) = 0.0D0
2699 C      YL(NL,K) = 0.0D0
2700 C
2701 C      DO 70 I = 1,N
2702 C          J = NL - I + 1
2703 C          L = I - NL + 1
2704 C
2705 C          IF (I .LT. NL) THEN
2706 C
2707 C              XX(I,K) = XL(J,K)
2708 C              YY(I,K) = YL(J,K)
2709 C
2710 C          ELSE
2711 C
2712 C              XX(I,K) = XU(L,K)
2713 C              YY(I,K) = YU(L,K)
2714 C
2715 C          ENDIF
2716 C
2717 C          WRITE(8,*)XX(I,K),YY(I,K),ZZ(K)/SPAN
2718 C
2719 C      70 CONTINUE
2720 C
2721 C      IF (NNN(K) .GT. MAX) MAX = NNN(K)
2722 C
2723 C      100 CONTINUE
2724 C
2725 C      CLOSE (UNIT = 8)
2726 C
2727 C      Step 4
2728 C
2729 C      CALL SYSTEM('CLS')
2730 C      SIGMA = -1.D0
2731 C      101 WRITE(6,122)IX
2732 C      READ(IU,*)N1
2733 C
2734 C      Check to see if the number entered falls within dimensional bounds.
2735 C
2736 C      IF (N1 .LT. 50 .OR. N1 .GE. IX) GOTO 101
2737 C
2738 C      Check to see if the number entered is an odd number.
2739 C
2740 C      IF (MOD(N1,2) .EQ. 0) THEN
2741 C          WRITE(6,123)
2742 C          GOTO 101
2743 C      ENDIF
```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2744 C
2745 102 CALL SYSTEM('CLS')
2746 C
2747     WRITE(6,124)
2748     READ(IU,*)ALPHA
2749 C
2750     WRITE(6,125)
2751     READ(IU,*)BETA
2752 C
2753     IF (ALPHA .GT. 10000.D0 .OR. ALPHA .LT. 0.D0 .OR.
2754 /      BETA .GT. 10000.D0 .OR. BETA.  LT. 0.D0) THEN
2755         WRITE(6,126)
2756         PAUSE
2757         GOTO 102
2758     ENDIF
2759 C
2760     WRITE(6,104)
2761 C
2762     OPEN(UNIT = 100 , FILE = 'INPUT.DAT' , ACCESS = 'APPEND'
2763 /      , FORM = 'FORMATTED',CARRIAGE CONTROL = 'FORTRAN'
2764 /      , STATUS = 'UNKNOWN')
2765 C
2766     WRITE(100,105)NAF,ALPHA,BETA
2767 C
2768     DO 200 K = 1,NAF
2769 C
2770         WRITE(100,106)K,NAME(K)
2771         WRITE(100,203)
2772         WRITE(100,220)CL(K),TH1(K),OSX(K),OSY(K),ZZ(K),THKN(K)
2773         WRITE(100,107)
2774         IA = (NNN(K) + 1) / 2
2775 C
2776         IF (NMID(K) .GE. IA) THEN
2777             NREF = NMID(K)
2778         ELSE
2779             NREF = NNN(K) - NMID(K) + 1
2780         ENDIF
2781 C
2782         DO 120 I = 1,NREF
2783             WRITE(100,210)XU(I,K),YU(I,K),XL(I,K),YL(I,K)
2784 120     CONTINUE
2785 C
2786 200 CONTINUE
2787 C
2788     N = MAX
2789 C
2790     CLOSE(UNIT = 20)
2791     CLOSE(UNIT = 100)
2792 C
2793     RETURN
2794 93  FORMAT(1X,/,/,
2795 1      ' Since this is a free nation , you are given a choice of'
2796 2      ',/, ' how you would like to generate your blade.'
```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2797      3  ,/, ' Following are your options:'
2798      4  ,/, ' 1.....GENERATE   YOUR OWN TWISTED AND TAPERED BLADE.'
2799      5  ,/, '           (i.e., The user will input the chord/twist/thickness'
2800      6  ,/, '                distribution at the terminal.)'
2801      7  ,/, ' 2.....USE Example 1 chord/twist/thickness distribution.'
2802      8  ,/, ' 3.....USE Example 2 chord/twist/thickness distribution.'
2803      9  ,/, ' 4.....USE A chord/twist/thickness distribution that is'
2804      1  ,/, '           not listed here.'
2805      2  ,/, ' NOTE: Option 4 can only be used if you have already'
2806      3  ,/, ' created a chord/twist/thickness distribution data file.'
2807      4  ,/, ' (Please refer to the user's manual on how to generate"
2808      5  ,/, ' this data file.)'
2809      6  ,/, ' Please enter your choice number here ---> ')
2810  94  FORMAT(1X,/,
2811      1  ' PLEASE ENTER THE NAME OF DATA FILE CONTAINING THE'
2812      2  ,/, ' chord/twist/thickness distribution required.'
2813      3  ,/, ' File name extension must be .DAT '
2814      4  ,/, ' e.g. File name ---> JOSEPH.DAT '
2815      5  ,/, ' File name ---> ')
2816  95  FORMAT(1X,/, ' FILE NAME ',A12,' DOES NOT EXIST. TRY AGAIN !')
2817  96  FORMAT(1X,/,
2818      1  ' How many airfoils do you want to input to form the blade?'
2819      2  ,/, ' *** Enter a number between 2 and',I3,' ***'
2820      3  ,/, ' ---> ')
2821  97  FORMAT(1X,/, ' *** ERROR! Input must be between 2 and',I3,'.')
2822  98  FORMAT(1X,/,
2823      1  ' Enter the span of the desired blade (i.e., the radius)'
2824      2  ,/, ' ---> ')
2825  99  FORMAT(1X,/,
2826      1  ' Do you want to scale the input airfoils ?'
2827      2  ,/, ' Enter :           N for NO'
2828      3  ,/, '           Y for YES'
2829      4  ,/, ' Enter your choice here ---> ')
2830 103  FORMAT(1X,/,
2831      1  ' You must choose an existing option, please try AGAIN!')
2832 104  FORMAT(1X,/,
2833      1  ' PLEASE WAIT.....'
2834      2  ,/, ' CALCULATIONS ARE TAKING PLACE .....')
2835 105  FORMAT(/,10X,' Number of input airfoils = ',I2,/,10X,
2836      /  ' Tension parameters: Alpha = ',F7.2,', and Beta = ',F7.2)
2837 106  FORMAT(1H1,/,10X,' Input data file name for airfoil',I2,' is '
2838      /  ',A12,' and the input data are:',/)
2839 107  FORMAT(/,17X,'XU',12X,'YU',13X,'XL',11X,'YL',/)
2840 108  FORMAT(1X,/,
2841      1  ' Enter the name of file containing the normalized airfoil data'
2842      2  ,/, ' File name extension must be .DAT'
2843      3  ,/, ' e.g. File name ---> JOSEPH.DAT '
2844      4  ,/, ' Please enter the file name here ---> ')
2845 109  FORMAT(1X,/, ' File name does not exist !! Please try AGAIN!')
2846 110  FORMAT(1X,/, ' Reading data from data file ',A)
2847 111  FORMAT(1X,/, ' YOU HAVE TO LEAVE ROOM FOR MORE SECTIONS.'
2848      1  ' REDUCE Z !')

```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2849 112 FORMAT(1X,/, ' Enter the twist angle (in degrees) ---> ')
2850 113 FORMAT(1X,/, ' Enter the chord length of airfoil ',I3,' ---> ')
2851 114 FORMAT(1X,/, ' The chord length cannot be zero or less.'
2852 1      '      Try again !')
2853 115 FORMAT(1X,/,
2854 1      ' Enter the maximum thickness/chord ratio.'
2855 2      ,/, ' The value should be a decimal less than 1.0. ---> ')
2856 116 FORMAT(1X,/,
2857 1      ' The maximum thickness/chord ratio cannot be'
2858 2      ,/, ' greater than 1.0. Try again !')
2859 117 FORMAT(1X,/,
2860 1      ' Enter twist center X- and Y-coordinates.'
2861 3      ,/, ' ENTER 1 if you want to input by yourself'
2862 4      ,/, ' ENTER 2 if you want to determine by the program'
2863 5      ,/, ' NOTE: OPTION 2 only applies to the case where the'
2864 6      ,/, ' twist center is located at the intersection'
2865 7      ,/, ' of X = 1/3 and the meanline.'
2866 8      ,/, ' PLEASE ENTER YOUR CHOICE NUMBER HERE ---> ')
2867 118 FORMAT(1X,/, ' Enter X-coordinate here ---> ')
2868 119 FORMAT(1X,/, ' Enter Y-coordinate here ---> ')
2869 121 FORMAT(1X,/, ' Please wait, program is running!')
2870 122 FORMAT(1X,/,
2871 1      ' ENTER the number of points you want per airfoil:'
2872 2      ,/, ' An ODD integer number between 50 and ',I4,', ---> ')
2873 123 FORMAT(1X,/,
2874 1      ' .....YOU ENTERED AN EVEN NUMBER.....'
2875 2      ,/, ' **** YOU DID NOT ENTER AN ODD NUMBER. TRY AGAIN ! ****')
2876 124 FORMAT(1X,/,
2877 1      ' WE NOW NEED INPUT FOR BLADE SURFACE INTERPOLATION.'
2878 2      ,/, ' Enter the tension factors in X and Z directions,'
2879 3      ,/, ' respectively.'
2880 4      ,/, ' The numbers must be greater than 0 (zero)'
2881 4      ,/, ' and less than 1000'
2882 5      ,/, ' ENTER X TENSION FACTOR (standard value is 1) ---> ')
2883 125 FORMAT(1X,/,
2884 1      ' ENTER Z TENSION FACTOR (standard value is 1) ---> ')
2885 126 FORMAT(1X,/,
2886 1      ' X and Z tension factors must be greater than zero'
2887 2      ,/, ' and less than 1000. Please enter these two'
2888 3      ,/, ' parameters AGAIN !')
2889 191 FORMAT(' Please wait, scaling program is running!')
2890 201 FORMAT(1X,I2)
2891 202 FORMAT(1X,2F8.2)
2892 203 FORMAT(7X,'Chord-',7X,'Twist',10X,'Twist Center',9X,'Span',
2893 / 9X,'max',/,7X,'Length',7X,'Angle',10X,'X',10X,'Y',8X,'Coord.'
2894 / ,5X,'thickness')
2895 205 FORMAT(1X,4(F9.5,1X))
2896 210 FORMAT(10X,4(2X,E12.5))
2897 220 FORMAT(2X,6(2X,F10.4))
2898 301 FORMAT(A1)
2899 302 FORMAT(
2900 / ' Below are airfoil types to choose from. Data are',/,
2901 / ' contained in data files. They are already normalized',/,

```

SUBROUTINE TINPUT Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2902      /      ' with respect to chord.',/
2903      /      ' If you do not desire any of the stated airfoil types.',/,
2904      /      ' you can choose option 7.',/,
2905      /      ' ',/,
2906      /      ' 1.....S806A   Tangler Somers thin airfoil',/,
2907      /      ' 2.....S805A/6A   "      "      "      "      ',/,
2908      /      ' 3.....S805A   "      "      "      "      ',/,
2909      /      ' 4.....S805A/7A   "      "      "      "      ',/,
2910      /      ' 5.....S807     "      "      "      "      ',/,
2911      /      ' 6.....S808     "      "      "      "      ',/,
2912      /      ' ',/,
2913      /      ' 7.....None of the above. I will enter my own.',/,
2914      /      ' ',/,
2915      /      ' Enter desired airfoil number for position ',I2,' here ---> ' )
2916      303  FORMAT(' The Z coordinate must be between',F7.2,' and ',F7.2)
2917      304  FORMAT(' Enter Z coordinate for airfoil ',I2,' here ---> ')
2918      305  FORMAT(' Data from file ',A,' for airfoil ',I2,' have been read')
2919      400  FORMAT(A)
2920      401  FORMAT(8F10.5)
2921      402  FORMAT(F10.5)
2922      403  FORMAT(3E15.6)
2923      END

```

SUBROUTINE TRANS Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX

Source file Listing

```

2924 C
2925 C-----
2926 C
2927       SUBROUTINE TRANS(N,NNN,NMID,NAF,SIGMA)
2928 C
2929 C*****
2930 C  This subroutine transforms the data from the physical coordinates *
2931 C  to the computational coordinates. *
2932 C*****
2933 C  The sequence of steps is as follows: *
2934 C *
2935 C      1.   The normalized Z coordinate ETA is computed. *
2936 C      2.   The polygonal arclength S is first computed. *
2937 C      3.   The polygonal arclenth is normalised and saved as ZI *
2938 C      4.   New values of Y and X are interpolated for by using the *
2939 C            normalised polygonal arclength, ZI, as the independent *
2940 C            variable. *
2941 C      5.   The new values of X are dumped into array U1 *
2942 C      6.   The new values of Y are dumped into array U2. *
2943 C *
2944 C*****
2945 C  There are going to be two computational domains. *
2946 C  Both computational domains consist of ZI and ETA as independent *
2947 C  variables. *
2948 C      In the first, X is the dependent variable. ie. X = X(zi,eta) *
2949 C      In the second, Y is the dependent variable. ie. Y = Y(zi,eta) *
2950 C  The maximum S value, SMAX(I), at each span station, is stored for *
2951 C  later use. *
2952 C*****
2953 C
2954       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2955       PARAMETER(IX = 120 , IY = 10)
2956 C
2957       COMMON /C3/X(IX,IY),Y(IX,IY)/C12/Z(IY)/C4/ZI(IX),ETA(IY)
2958       COMMON /C5/XMIN(IY),XMAX(IY)
2959       COMMON /C20/U1(IX,IY),U2(IX,IY)/C25/SLE(IY),TAU
2960 C
2961       DIMENSION NNN(IY),X1(IX),Y1(IX),SN(IX),S(IX),NMID(IY)
2962 C
2963       CALL SYSTEM('CLS')
2964       WRITE(6,100)
2965       READ(5,*)TAU
2966 C
2967       CALL SYSTEM('CLS')
2968       WRITE(6,101)
2969 C
2970       DO 40 J = 1,NAF
2971           NIX = NNN(J)
2972 C
2973 C  Compute the normalized Z coordinate
2974 C
2975       ETA(J) = (Z(J)-Z(1))/(Z(NAF)-Z(1))

```


SUBROUTINE TRANS Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```

2976 C
2977 C   Compute the polygonal arclenth. Initialize the first value as zero
2978 C
2979 C       S(1) = 0.0D0
2980 C
2981 C       DO 10 I = 2,NIX
2982 C           XSQ = X(I,J) - X(I-1,J)
2983 C           YSQ = Y(I,J) - Y(I-1,J)
2984 C           S(I) = DSQRT(XSQ*XSQ+YSQ*YSQ) + S(I-1)
2985 10   CONTINUE
2986 C
2987 C   The length of the spanstation polygonal arc is :
2988 C
2989 C       SMAX = S(NIX)
2990 C
2991 C   Compute the normalized polygonal arclengths.
2992 C
2993 C       DO 20 I = 1,NIX
2994 C           SN(I) = S(I)/SMAX
2995 C           X1(I) = X(I,J)
2996 C           Y1(I) = Y(I,J)
2997 20   CONTINUE
2998 C
2999 C       IA = NMID(J)
3000 C       SLE(J) = SN(IA)
3001 C       SAME = SLE(J) * (SLE(J)-1.D0)
3002 C       CONST1 = (SLE(J) * SLE(J) - 0.5D0) / SAME
3003 C       CONST2 = (0.5D0 - SLE(J)) / SAME
3004 C
3005 C       DO 50 I = 1,NIX
3006 C           SN(I) = SN(I) * (CONST1 + CONST2 * SN(I) )
3007 50   CONTINUE
3008 C
3009 C       SLE(J) = 0.5D0
3010 C       SMID = SLE(J)
3011 C
3012 C   Resample X and Y using the normalized polygonal arc length as the
3013 C   independent variable.
3014 C
3015 C       B1 = 1.0D0 + (DEXP(TAU) - 1.0D0) * SMID
3016 C       B2 = 1.0D0 + (DEXP(-TAU) - 1.0D0) * SMID
3017 C       B = DLOG(B1/B2) / (2.0D0*TAU)
3018 C       DZI = 1.0D0 / FLOAT(N-1)
3019 C
3020 C       DO 30 I = 1,N
3021 C           ZIT = DZI*FLOAT(I-1)
3022 C           ZZIT = SMID*(1.0D0+(DSINH(TAU*(ZIT-B) )/DSINH(TAU*B) ) )
3023 C           CALL SPT11D(NIX,SN,X1,SIGMA,ZZIT,X1IT)
3024 C           CALL SPT11D(NIX,SN,Y1,SIGMA,ZZIT,Y1IT)
3025 C           U1(I,J) = X1IT
3026 C           U2(I,J) = Y1IT
3027 C           ZI(I) = ZZIT

```

SUBROUTINE TRANS Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
3028 30      CONTINUE
3029 C
3030 40      CONTINUE
3031 C
3032      RETURN
3033 100     FORMAT(1X,/,/,
3034 1        ' ENTER THE STRETCHING PARAMETER FOR CONCENTRATING'
3035 2        ',/, ' THE AIRFOIL DATA POINTS ABOUT THE LEADING EDGE.'
3036 3        ',/, ' The STANDARD VALUE RANGES BETWEEN 4.0 AND 6.0'
3037 4        ',/, ' THE FEWER POINTS YOU USED TO REPRESENT THE'
3038 5        ',/, ' AIRFOIL CROSS SECTION, THE SMALLER'
3039 6        ',/, ' VALUE YOU SHOULD USE.'
3040 7        ',/, ' If no stretching desired, enter 0.0001'
3041 8        ',/, ' Enter your choice here ---> ')
3042 101     FORMAT(1X,
3043 1        ' PLEASE WAIT.....'
3044 2        ',/, ' CALCULATIONS IN PROGRESS.....')
3045      END
```

SUBROUTINE TRANSPO Compiling Options:/N0/N7/NA/NB/NF/H/NI/L/P/R/S/NT/W/NX
Source file Listing

```
3046 C
3047 C-----
3048 C
3049 C          SUBROUTINE TRANSPO(N,M,A,AT)
3050 C
3051 C*****
3052 C This subroutine finds the transpose of matrix A and stores  *
3053 C the result in matrix AT.                                     *
3054 C*****
3055 C
3056 C          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
3057 C
3058 C          DIMENSION A(N,M),AT(M,N)
3059 C
3060 C          DO 20 I = 1,N
3061 C
3062 C              DO 10 J = 1,M
3063 C                  AT(J,I) = A(I,J)
3064 10          CONTINUE
3065 C
3066 20          CONTINUE
3067 C
3068 C          RETURN
3069 C          END
```

APPENDIX B

DERIVATION OF EQUATIONS (2.2) AND (2.5)

For a given partition

$$a = x_1 < x_2 < \dots < x_i < \dots < x_{n-1} < x_n = b$$

of an interval $[a, b]$, and a set of tension parameters α_i ($1 \leq i \leq n$), $g(x)$ is a tension spline if it satisfies the following conditions [1]:

$$g \in C^3[a, b] \tag{B.1}$$

and

$$(D^4 - \alpha_i^2 D^2)g = 0 \tag{B.2}$$

in each subinterval i , where $x_i \leq x \leq x_{i+1}$. In equation (B.1) C^3 is a differentiable space up to third order and D in equation (B.2) is the differential operator defined as $D = d/dx$.

Accordingly, as in [2], the general solution of equation (B.2) is

$$g_i(x) = A_{i,1} + A_{i,2}(x - x_i) + A_{i,3}e^{\alpha_i(x - x_i)} + A_{i,4}e^{-\alpha_i(x - x_i)} \tag{B.3}$$

where coefficients $A_{i,1}$, $A_{i,2}$, $A_{i,3}$, and $A_{i,4}$ are unknown constants to be determined by the boundary conditions for each subinterval i .

For convenience in applying the boundary conditions for a finite domain, equation (B.3) is rewritten [2] in terms of the hyperbolic functions as

$$g_i(x) = A_{i,1}(x - x_i) + A_{i,2}(x_{i+1} - x) + A_{i,3}\psi_i(x - x_i) + A_{i,4}\psi_i(x_{i+1} - x) \tag{B.4}$$

where

$$\psi_i(x) = \frac{\Delta x_i \sinh(\alpha_i x) - x \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \quad (B.5)$$

and

$$\Delta x_i = x_{i+1} - x_i \quad (B.6)$$

with the following conditions:

$$g(x_i) = y_i \quad (B.7)$$

$$g(x_{i+1}) = y_{i+1} \quad (B.8)$$

$$g'(x_i) = y'_i \quad (B.9)$$

$$g'(x_{i+1}) = y'_{i+1} \quad (B.10)$$

where superscript ' denotes the derivative of y with respect to x.

Equation (B.4) in summation form is actually equation (2.2).

There are four unknown coefficients ($A_{i,k}$, $k = 1, 2, 3$, and 4 for each i) in equation (B.4) and there are four conditions ([B.7] through [B.10]) to solve them. These coefficients are determined as follows.

By solving equation (B.4) at $x = x_i$, and $x = x_{i+1}$ and using (B.7) and (B.8) as the boundary conditions, one can get the following two relations:

$$A_{i,2} \Delta x_i = y_i \quad (B.11)$$

for $x = x_i$, and

$$A_{i,1} \Delta x_i = y_{i+1} \quad (B.12)$$

for $x = x_{i+1}$.

The other two relations are obtained by solving for the derivative of the interpolating function (B.4), at the boundaries of the subdomain, i , and then equating the results to the boundary conditions (B.9) and (B.10). From equation (B.4), the first derivative of g_i with respect to x is

$$g'_i(x) = A_{i,1} - A_{i,2} + A_{i,3} \left[\frac{\Delta x_i \alpha_i \cosh(\alpha_i x - x_i) - \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right] - A_{i,4} \left[\frac{\Delta x_i \alpha_i \cosh(\alpha_i x_{i+1} - x) - \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right] \quad (B.13)$$

By solving equation (B.13) at $x = x_i$, and $x = x_{i+1}$, and equating the result to the true values of slope at these points, we get the following two equations:

$$A_{i,1} - A_{i,2} - A_{i,3} - v_i A_{i,4} = y'_i \quad (B.14)$$

for $x = x_i$, and

$$A_{i,1} - A_{i,2} + v_i A_{i,3} + A_{i,4} = y'_{i+1} \quad (B.15)$$

for $x = x_{i+1}$, where

$$v_i = \left[\frac{\Delta x_i \alpha_i \cosh(\alpha_i \Delta x_i) - \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right]$$

Equations (B.11), (B.14), (B.12), and (B.15) can be put into the following matrix form:

$$\begin{bmatrix} 0 & \Delta x_i & 0 & 0 \\ 1 & -1 & -1 & -v_i \\ \Delta x_i & 0 & 0 & 0 \\ 1 & -1 & v_i & 1 \end{bmatrix} \begin{bmatrix} A_{i,1} \\ A_{i,2} \\ A_{i,3} \\ A_{i,4} \end{bmatrix} = \begin{bmatrix} y_i \\ y'_i \\ y_{i+1} \\ y'_{i+1} \end{bmatrix} \quad (B.16)$$

for $x_i \leq x \leq x_{i+1}$.

This matrix system can then be used to solve for the spline coefficients A_{ik} , $k = 1, 2, 3, 4$, for each subinterval i , by expressing it in the form

$$A_{i,k} = C(\Delta x_i, v_i) K_i \quad (B.17)$$

where

$$A_{i,k} = \begin{bmatrix} A_{i,1} \\ A_{i,2} \\ A_{i,3} \\ A_{i,4} \end{bmatrix} \quad (B.18)$$

$$C(\Delta x_i, v_i) = \begin{bmatrix} 0 & 0 & \frac{1}{\Delta x_i} & 0 \\ \frac{1}{\Delta x_i} & 0 & 0 & 0 \\ -\frac{1}{\Delta x_i} \frac{1}{1-v_i} & \frac{-1}{1-v_i^2} & \frac{1}{\Delta x_i} \frac{1}{1-v_i} & \frac{-v_i}{1-v_i^2} \\ \frac{1}{\Delta x_i} \frac{1}{1-v_i} & \frac{v_i}{1-v_i^2} & -\frac{1}{\Delta x_i} \frac{1}{1-v_i} & \frac{1}{1-v_i^2} \end{bmatrix} \quad (B.19)$$

and

$$K_i = \begin{bmatrix} y_i \\ y'_i \\ y_{i+1} \\ y'_{i+1} \end{bmatrix} \quad (B.20)$$

Equation (B.17) is actually equation (2.5).

REFERENCE

1. Pruess, S., "Properties of Splines in Tension," Journal of Approximating Theory, Vol. 17, pp. 86-96, 1976.
2. SpäTh, H., "Exponential Spline Interpolation," Computing, Vol. 4, pp. 225-233, 1969.

APPENDIX C

DERIVATION OF EQUATION (2.11)

To derive equation (2.11), the curvature continuity condition will be used at each node i , $2 \leq i \leq n-1$, such that the unknown quantities y'_i can be determined. The second derivative of g_i for subdomain i is obtained by differentiating equation (B.4) twice with respect to x :

$$g_i''(x) = A_{i,3} \left[\frac{\Delta x_i \alpha_i^2 \sinh(\alpha_i(x - x_i))}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right] + A_{i,4} \left[\frac{\Delta x_i \alpha_i^2 \sinh(\alpha_i(x_{i+1} - x))}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right] \quad (C.1)$$

For subdomain $i-1$, g_{i-1}'' is

$$g_{i-1}''(x) = A_{i-1,3} \left[\frac{\Delta x_{i-1} \alpha_{i-1}^2 \sinh(\alpha_{i-1}(x - x_{i-1}))}{\sinh(\alpha_{i-1} \Delta x_{i-1}) - \alpha_{i-1} \Delta x_{i-1}} \right] + A_{i-1,4} \left[\frac{\Delta x_{i-1} \alpha_{i-1}^2 \sinh(\alpha_{i-1}(x_i - x))}{\sinh(\alpha_{i-1} \Delta x_{i-1}) - \alpha_{i-1} \Delta x_{i-1}} \right] \quad (C.2)$$

Solving these second derivatives at $x = x_i$ we get

$$g_{i-1}''(x_i) = A_{i-1,4} \left[\frac{\Delta x_{i-1} \alpha_{i-1}^2 \sinh(\alpha_{i-1} \Delta x_{i-1})}{\sinh(\alpha_{i-1} \Delta x_{i-1}) - \alpha_{i-1} \Delta x_{i-1}} \right] \quad (C.3)$$

and

$$g_i''(x_i) = A_{i,3} \left[\frac{\Delta x_i \alpha_i^2 \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right] \quad (C.4)$$

Equating (C.3) and (C.4) yields

$$A_{i,4} \left[\frac{\Delta x_i \alpha_i^2 \sinh(\alpha_i \Delta x_i)}{\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i} \right] = A_{i-1,3} \left[\frac{\Delta x_{i-1} \alpha_{i-1}^2 \sinh(\alpha_{i-1} \Delta x_{i-1})}{\sinh(\alpha_{i-1} \Delta x_{i-1}) - \alpha_{i-1} \Delta x_{i-1}} \right] \quad (C.5)$$

where $A_{i,4}$ and $A_{i-1,3}$ are obtained from solving system (B.17), and are represented as

$$A_{i,4} = \frac{-v_i y_i' - y_{i+1}' + (1 + v_i) \frac{\Delta y_i}{\Delta x_i}}{(v_i^2 - 1)} \quad (C.6)$$

and

$$A_{i-1,3} = \frac{y_{i-1}' + v_{i-1} y_i' - (1 + v_{i-1}) \frac{\Delta y_{i-1}}{\Delta x_{i-1}}}{(v_{i-1}^2 - 1)} \quad (C.7)$$

After substituting (C.6) and (C.7) into (C.5) and rearranging, we get (2.11), which takes the form

$$t_{i-1} y_{i-1}' + (t_{i-1} v_{i-1} + t_i v_i) y_i' + t_i y_{i+1}' = t_{i-1} (1 + v_{i-1}) \frac{\Delta y_{i-1}}{\Delta x_{i-1}} + t_i (1 + v_i) \frac{\Delta y_i}{\Delta x_i} \quad (C.8)$$

where

$$t_i = \left[\frac{\Delta x_i \alpha_i^2 \sinh(\alpha_i \Delta x_i)}{(v_i^2 - 1) (\sinh(\alpha_i \Delta x_i) - \alpha_i \Delta x_i)} \right] \quad (C.9)$$

Document Control Page	1. SERI Report No. NREL/TP-257-4099	2. NTIS Accession No. DE92010574	3. Recipient's Accession No.
4. Title and Subtitle A Surface Definition Code for Turbine Blade Surfaces			5. Publication Date May 1992
			6.
7. Author(s) S. L. Yang, D. Oryang, M. J. Ho			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Mechanical Engineering Department Michigan Technological University Houghton, Michigan 49931			10. Project/Task/Work Unit No. WE21.8104
			11. Contract (C) or Grant (G) No. (C) XX-7-06092-2 (G)
12. Sponsoring Organization Name and Address National Renewable Energy Laboratory 1617 Cole Blvd. Golden, Colorado 80401-3393			13. Type of Report & Period Covered Technical Report
			14.
15. Supplementary Notes NREL Technical Monitor: Peter K.C. Tu, (303) 231-7000			
16. Abstract (Limit: 200 words) A numerical interpolation scheme has been developed for generating the three-dimensional geometry of wind turbine blades. The numerical scheme consists of (1) creating the frame of the blade through the input of two or more airfoils at some specific spanwise stations and then scaling and twisting them according to the prescribed distributions of chord, thickness, and twist along the span of the blade; (2) transforming the physical coordinates of the blade frame into a computational domain that complies with the interpolation requirements; and finally (3) applying the bi-tension spline interpolation method, in the computational domain, to determine the coordinates of any point on the blade surface. Detailed descriptions of the overall approach to and philosophy of the code development are given along with the operation of the code. To show the usefulness of the bi-tension spline interpolation code developed, two examples are given, namely CARTER and MICON blade surface generation. Numerical results are presented in both graphic and tabular data forms. The solutions obtained in this work show that the computer code developed can be a powerful tool for generating the surface coordinates for any three-dimensional blade.			
17. Document Analysis a. Descriptors wind turbines, blade surfaces, surface definition code b. Identifiers/Open-Ended Terms c. UC Categories 261			
18. Availability Statement National Technical Information Service U.S. Department of Commerce 5285 Port Royal Road Springfield, VA 22161			19. No. of Pages 207
			20. Price A10

END

**DATE
FILMED**

7 / 8 / 92