NASA TM-84666

NASA Technical Memorandum 84666

NASA-TM-84666 19850022698

COMPUTER PROGRAMS FOR SMOOTHING AND SCALING AIRFOIL COORDINATES

Harry L. Morgan, Jr.

July 1983

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

This report contains detailed descriptions of the theoretical methods and associated computer codes of a program to smooth and a program to scale arbitrary airfoil coordinates. The smoothing program utilizes both least-squares polynomial and least-squares cubic spline techniques to smooth iteratively the second derivatives of the y-axis airfoil coordinates with respect to a transformed x-axis system which unwraps the airfoil and stretches the nose and trailing-edge regions. The corresponding smooth airfoil coordinates are then determined by solving a tridiagonal matrix of simultaneous cubic spline equations relating the y-axis coordinates and their corresponding second derivatives. A technique for computing the camber and thickness distribution of the smoothed airfoil is also discussed.

The scaling program can then be used to scale the thickness distribution generated by the smoothing program to a specified maximum thickness which is then combined with the camber distribution to obtain the final scaled airfoil contour. Computer listings of the smoothing and scaling programs are included as appendices. A user-guide and sample input and output cases for both programs are also included as appendices. Both computer programs are available from COSMIC with identifications LAR-13132 for the airfoil smoothing program "AFSMO" and LAR-13133 for the airfoil scaling program "AFSCL".

INTRODUCTION

Since its early beginning, the NACA and the NASA have been actively involved in the design and testing of airfoil sections for a wide variety of applications. During the 1930's, 40's, and 50's, the airfoils developed by the NACA consisted of the well-known 4-digit-, 5-digit-, 1-, 6-, and 7-series airfoils. These airfoils were generated by combining thickness and camber distributions that were defined analytically by polynomial equations of various order, and, therefore, the surface coordinates of these airfoils are very smooth. A summary of many of the NACA airfoils and a detailed description of the equations used to generate their coordinates are presented in reference 1.

During the mid-1960's, the introduction of the supercritical airfoil concept by Dr. Richard Whitcomb of the Langley Research Center created a renewed interest in the development of an improved series of airfoils for applications at high subsonic and transonic flow conditions. Initial attempts to generate a series of supercritical airfoils from analytical expressions were unsuccessful because no theoretical methods were available to guide in the selection of adequate analytical expressions relating airfoil shape and the desired high-speed flow characteristics. During the early 1970's, Dr. Paul R. Garabedian of New York University developed a series of computer codes for the design and analysis of supercritical airfoils with no or very weak shocks. These codes, as described in reference 2, relied on a system of equations based on the method of complex characteristics in the hodograph plane and are solved numerically using conformal mapping and fast Fourier transform techniques.

During the mid- and latter-1970's, the NASA was also actively involved in the development of an improved series of subsonic airfoils for application to general aviation, glider, and commuter aircraft. Several computer codes were developed, such as the NASA/Lockheed-Georgia Multi-Component Airfoil Code (ref. 3) and the Eppler Low-Speed Airfoil Code (ref. 4) to aid in the design and analysis of these new airfoils. These codes utilize a variety of conformal mapping and distributed source- and vortex-singularity methods to obtain the potential flow characteristics of the airfoil and a variety of finite-difference and integral boundary-layer methods to obtain the viscous characteristics.

Both the subsonic and transonic airfoil codes have undergone extensive refinement and improvement in the past decade and are widely utilized by both the domestic and foreign scientific communities. The agreement between the theoretical and experimental characteristics of the airfoils designed using these codes has been generally excellent for airfoils with fully attached flow. The rapid development of the high-speed digital computer since the 1970's has greatly reduced the computer costs to design and analyze a new airfoil; therefore, it is no longer necessary to test a large number of airfoils to obtain one with the desired performance characteristics. The theoretical methods used in these computer codes are generally sensitive to the numerical techniques used and, as a result, often generate airfoils with wavy or unsmooth surface coordinates. The transonic airfoils have been shown to be particularly sensitive to coordinate smoothness both experimentally and theoretically.

The purpose of this report is to describe in detail the features of a computer code developed to smooth and scale airfoil coordinates. The smoothing code utilizes a variety of least-squares poynomial and cubic spline techniques to smooth the airfoil coordinates in the second derivative. The computer code has an internal Langley designation of "AFSMO" and consists primarily of a main controlling program and an input, a smoothing, a punch output, and plotting subroutines. Additional subroutines have been included to compute the camber and thickness distributions of the smoothed airfoil and to interpolate additional coordinates. The airfoil scaling program has an internal Langley designation of "AFSCL" and uses the camber and thickness distribution data generated by the AFSMO code to generate additional airfoil shapes with the same camber distribution and a scaled thickness distribution. The AFSCL code consists of a main controlling program, a subroutine to scale the coordinates, and a subroutine to fit a cubic spline through a set of input points. A detailed description of the smoothing and scaling methods used in these codes is presented in addition to a discussion of the possible applications of the codes. Appendices are included that describe the user input requirements, a sample input case, a sample output listing, sample plots, and tabulated listings for both programs.

SYMBOLS

a_i, b_i, c_i, d_i	polynomial coefficients
c	chord of airfoil
g	generalized cubic spline function
h	cubic spline interval
k	curvature
K	value of x/c where $\theta = \pi$
N	total number of upper and lower surface coordinates
t	thickness
S	least-squares cubic spline smoothing parameter
w	weighting factor
x,y	coordinates of airfoil
$\bar{x},\bar{y}$	nondimensionalized x/c and y/c coordinates
$\hat{x},\hat{y}$	coordinates in local camberline axis system
x_c,y_c	x/c and y/c coordinates of camberline
$\bar{y}'$	$d(y/c)/d\theta$
$\bar{y}''$	$d^2(y/c)/d\theta^2$
γ	local surface slope in $\hat{x}$- and $\hat{y}$-axis system
ϕ	local slope of camberline
θ	x-axis transformation function

Subscripts:

c	camber
i	iteration or element number
ℓ	lower surface
u	upper surface

DESCRIPTION OF SMOOTHING METHOD

Smoothing Criteria

The smoothness criteria used in the development of the smoothing method presented in this report is that the curvature distribution of the airfoil surface be continuous and smooth. The curvature, which is the reciprocal of the radius-of-curvature, is defined as

$$k = \frac{\left| d^2y/dx^2 \right|}{\left[1 + \left(\frac{dy}{dx} \right)^2 \right]^{3/2}} \qquad (1)$$

The curvature distribution will be continuous, provided the airfoil contour is continuous with single-valued upper and lower surface coordinates. This can easily be determined by visual inspection of the initial input airfoil shape. The application of cubic spline functions to relate the smoothed y-axis airfoil coordinates to their smoothed second derivatives with respect to the x-axis will insure that the first derivatives are smooth and, consequently, that the curvature distribution is also smooth. Therefore, the smoothing method established is first to compute the second derivatives of the input airfoil coordinates, to smooth the second derivatives, and then to employ cubic spline functions to determine the new smoothed airfoil coordinates.

The second derivatives of the input y-coordinates are determined by fitting a least-squares polynomial to each coordinate and a specified number of points adjacent to the coordinate and then by analytical differentiation, computing the second derivative of the coordinate and its new y-value. This procedure is repeated for each y-coordinate until a new set of y-values are obtained which are then substituted for the previous set of y-values. The entire procedure is repeated and each time the sum of the squares of the differences between the current and prior second derivatives is computed. This iterative procedure continues until a specified number of iterations have been reached, or the sum of the squares quantity falls below a specified value or begins to oscillate.

X-Axis Transformation Function

Initial attempts to employ this least-squares polynomial technique to an input set of x- and y- coordinates resulted in large oscillations in the computed second derivatives and the new y-values from one iteration to the next. The oscillation was caused by the very rapid change in the curvature in the nose or leading-edge region which is characteristic of most airfoils. This problem was eliminated by utilizing an x-axis transformation function that stretches the axis in the nose region. One such transformation function used in the multi-component airfoil analysis code developed by Lockheed-Georgia (ref. 3) is

$$\bar{x} = \frac{1}{2}\left[1 - \cos(\theta)\right], \qquad (2)$$

where $0 \leq \theta \leq \pi$. However, this transformation function stretches the x-axis in both the leading- and trailing-edge regions. For application in the multi-component analysis code this stretching at both ends of the airfoil is necessary to ensure adequate definition of the maximum suction peak in the leading-edge region and to properly satisfy the Kutta flow condition in the trailing-edge region. To smooth an airfoil does not require as much stretching of the x-axis in the trailing-edge region as in the leading-edge region because the curvature is generally considerably less near the trailing edge of the airfoil. The hyperbolic functions behave in a manner similar to that for trigometric functions and, after considerable trial-and-error, the following transformation equation was found that reduced the amount of trailing-edge stretching and that could be mated with the trigonometric equation (2) for the leading edge:

$$\bar{x} = K\left\{\tan^{-1}\left[\sinh\left(\theta - \pi/2\right)\right] + 1\right\}, \tag{3}$$

where $\pi/2 \leq \theta \leq \pi$. The constant K was determined by specifying that at θ equals π, the value of $\bar{x}$ is unity; therefore,

$$K = \frac{1}{\tan^{-1}\left[\sinh\left(\pi/2\right)\right] + 1} = 0.46278 \tag{4}$$

By substituting the constant of 1/2 in equation (2) with the constant K from equation (3), the transformation equation for the leading-edge region becomes

$$\bar{x} = K\left[1 - \cos(\theta)\right] \tag{5}$$

where $0 \leq \theta \leq \pi/2$.

The first and second derivatives of equation (3) are

$$\frac{d\bar{x}}{d\theta} = \frac{K}{\cosh\left(\theta - \pi/2\right)} \quad \text{and} \tag{6}$$

$$\frac{d^2\bar{x}}{d\theta^2} = -\frac{K\sinh\left(\theta - \pi/2\right)}{\cosh^2\left(\theta - \pi/2\right)}, \tag{7}$$

respectively, and of equation (5) are

$$d\bar{x}/d\theta = K\sin(\theta) \quad \text{and} \tag{8}$$

$$d^2\bar{x}/d\theta^2 = K\cos(\theta), \tag{9}$$

respectively. At $\theta = \pi/2$, the value of equations (3), (5), (6), and (8) is equal to K and the value of equations (7) and (9) is zero which verifies that the leading- and trailing-edge transformation equations are continuous at the matching point. A plot of the resultant transformation function and its first and second derivatives are presented in figure 1 and tabulated in table I.

The inverse of equation (3) is

$$\theta = \pi/2 + \sinh^{-1}\left[\tan\left(\frac{\bar{x}}{K} - 1\right)\right] , \tag{10}$$

where $\sinh^{-1}(z) = \ln\left(z + \sqrt{z^2 + 1}\right)$ and the inverse of equation (5) is

$$\theta = \cos^{-1}\left(1 - \frac{\bar{x}}{K}\right). \tag{11}$$

The first and second derivatives of the $\bar{y}$-coordinate with respect to $\bar{x}$ can be obtained from the derivatives with respect to the θ value using the following relationships:

$$d\bar{y}/d\bar{x} = \bar{y}' \frac{1}{d\bar{x}/d\theta} \tag{12}$$

$$d^2\bar{y}/d\bar{x}^2 = \frac{\bar{y}''(d\bar{x}/d\theta) - \bar{y}'\,(d^2\bar{x}/d\theta^2)}{(d\bar{x}/d\theta)^3} \tag{13}$$

Piecewise Least-Squares Polynomial Smoothing to Determine Second Derivative

The piecewise least-squares polynomial smoothing procedure requires that the independent variable increase monotonically to prevent simultaneous smoothing of upper and lower surface coordinates. This meant simply that the airfoil had to be unwrapped around the nose, which was easily accomplished by letting the lower surface transformation function run from 0 to $-\pi$ and the upper surface function run from 0 to $+\pi$. The remaining problem associated with computing the second derivatives using the least-squares polynomial procedure was to determine the number of points to include adjacent to the coordinate point and the degree of the polynomial. To determine these two quantities, the coordinates of the well-known NACA 0012 airfoil were input and various values were tried for each quantity until a combination was found that produced the best agreement between the calculated and theoretical values of the second derivatives. The number of points adjacent to the coordinate point was found to be 3 before and 3 after for a total of 7 points, and the degree of the polynomial was found to be 4. The computer code for the piecewise least-squares polynomial smoothing procedure is contained in subroutine LSQSMO.

Least-Squares Cubic Spline Smoothing of Second Derivative

After completion of the least-squares polynomial smoothing procedure, the resultant values of $\bar{y}''$ are input to subroutine CSDS which was formulated based on a method that fits a smooth cubic spline through a set of input data in a least-squares manner. The method defines a continuous cubic spline function in the form

$$g(\theta)_i = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i, \qquad (14)$$

where $h_i = (\theta - \theta_i)$ and $i = 1, 2, 3, \ldots, N-1$.

The coefficients a_i, b_i, c_i, and d_i are computed such that

$$\sum_{i=1}^{N} \left[\frac{g(\theta)_i - f_i}{\delta f_i}\right]^2 \leq S \tag{15}$$

and $$\int_{\theta_1}^{\theta_N} \left[d^2g/d\theta^2\right]^2 d\theta \text{ is a minimum} \tag{16}$$

where the smoothing parameter S is in the interval $(N - \sqrt{2N}) \leq S \leq (N + \sqrt{2N})$, N is the number of points, $f_i = \bar{y}_i''$, and δf_i is the allowable standard error deviation of f_i. A detailed description of the least-squares cubic spline method is presented in reference 5. After extensive application of the smoothing program to a wide range of airfoil shapes, the value of 10^{-4} was selected for standard error deviation and a conservative value of N was chosen for the smoothing parameter S.

Cubic-Spline to Compute New $\bar{y}$-Coordinate.

After obtaining the new smoothed second derivatives, the next step is to determine the corresponding smoothed $\bar{y}$-coordinate values that are also smooth and continuous in the interval between input points. The natural choice was a cubic spline which consists of defining the $\bar{y}$ coordinates between the interval end points with a third-order polynomial similar to equation (14) and solving for the coefficients so that the $\bar{y}$ coordinates and the first- and second-derivatives at the intersection with the adjacent interval are equal at each end. This ensures that the $\bar{y}$ coordinates, the slope, and the curvature are continuous and smooth. The cubic spline polynomial and its first- and second-derivatives are:

$$\bar{y}_i = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \quad , \tag{17}$$

$$\bar{y}'_i = 3a_i h_i^2 + 2b_i h_i + c_i \quad , \tag{18}$$

and $$\bar{y}''_i = 6a_i h_i + 2b_i \tag{19}$$

where $h_i = (\theta - \theta_i)$.

At the two end points of the ith interval, the $\bar{y}$ coordinates are

$$\bar{y}_i = d_i \tag{20}$$

at $\theta = \theta_i$ and

$$\bar{y}_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \tag{21}$$

at $\theta = \theta_{i+1}$, and the second derivatives are

$$\bar{y}_i'' = 2b_i \text{ or } b_i = \bar{y}_i''/2 \tag{22}$$

at $\theta = \theta_i$ and

$$\bar{y}''_{i+1} = 6a_i h_i + 2b_i \text{ or } a_i = \frac{\bar{y}''_{i+1} - \bar{y}''_i}{6h_i} \tag{23}$$

at $\theta = \theta_{i+1}$.

Combining equations (20) through (23) and simplifying,

$$c_i = \left(\frac{\bar{y}_{i+1} - \bar{y}_i}{h_i}\right) - \left(\frac{\bar{y}''_{i+1} + 2\bar{y}''_i}{6}\right) h_i \tag{24}$$

At $\theta = \theta_i$, $\bar{y}''_i$ equals c_i and from the previous interval

$$\bar{y}'_i = 3a_{i-1}h^2_{i-1} + 2b_{i-1}h_{i-1} + c_{i-1} \quad (25)$$

where from a similar analysis,

$$a_{i-1} = \left(\frac{\bar{y}''_i - \bar{y}''_{i-1}}{6h_{i-1}}\right) , \quad (26)$$

$$b_{i-1} = \frac{\bar{y}''_{i-1}}{2} , \quad (27)$$

and

$$c_{i-1} = \left(\frac{\bar{y}_i - \bar{y}_{i-1}}{h_{i-1}}\right) - \left(\frac{\bar{y}''_i + 2\bar{y}''_{i-1}}{6}\right) h_{i-1}. \quad (28)$$

By substituting equations (26), (27), and (28) into equation (25) and setting equation (24) equal to (25), the following simplified form of the cubic-spline equation is derived:

$$\left(\frac{1}{h_{i-1}}\right) \bar{y}_{i-1} - \left(\frac{1}{h_i} + \frac{1}{h_{i-1}}\right) \bar{y}_i + \left(\frac{1}{h_i}\right) \bar{y}_{i+1} = \left(\frac{h_{i-1}}{6}\right) \bar{y}''_{i-1} +$$

$$\left(\frac{h_{i-1} + h_i}{3}\right) \bar{y}''_i + \left(\frac{h_i}{6}\right) \bar{y}''_{i+1} \quad (29)$$

which represents a set of tridiagonal equations with i = 2, 3, 4, . . ., N - 1. By specifying the desired $\bar{y}$ coordinates at the end points, the resultant N- by N-matrix equation can be solved with a simplified matrix inversion technique. The equations that define the $\bar{y}$ coordinates and the first- and second-derivatives in each interval are

$$\bar{y}(\theta) = \bar{y}_i'' \left[\frac{(\theta_{i+1} - \theta)^3}{6h_i} - \frac{(\theta_{i+1} - \theta)\, h_i}{6}\right] +$$

$$\bar{y}_{i+1}'' \left[\frac{(\theta - \theta_i)^3}{6h_i} - \frac{(\theta - \theta_i)}{6}\, h_i\right] + \left[\frac{\bar{y}_i(\theta_{i+1}-\theta) + \bar{y}_{i+1}(\theta - \theta_i)}{h_i}\right], \quad (30)$$

$$\bar{y}'(\theta) = \bar{y}_i'' \left[\frac{h_i}{6} - \frac{(\theta_{i+1} - \theta)^2}{2h_i}\right] + \bar{y}_{i+1}'' \left[\frac{(\theta - \theta_i)^2}{2h_i} - \frac{h_i}{6}\right] +$$

$$\left[\frac{\bar{y}_{i+1} - \bar{y}_i}{h_i}\right], \quad (31)$$

and

$$\bar{y}''(\theta) = \bar{y}_i'' \left(\frac{\theta_{i+1} - \theta}{h_i}\right) + \bar{y}_{i+1}'' \left(\frac{\theta - \theta_i}{h_i}\right) \quad (32)$$

where $h_i = (\theta_{i+1} - \theta_i)$. The computer code for this cubic spline method is contained in subroutine INVY in the airfoil smoothing program.

The initial application of the cubic spline method with the lower and upper trailing-edge $\bar{y}$ coordinates input for i=1 and N, produced airfoil shapes that did not generally pass through the nose

$\bar{y}$ coordinate computed during the previous least-squares smoothing step. This problem was partially overcome by first applying the cubic-spline method from the lower surface trailing edge to the nose and then from the nose to the upper surface trailing-edge coordinates. Although this procedure generated an airfoil shape that had the same $\bar{y}$ coordinate and second derivative at the nose when approaching from both the upper and lower surface, the first derivatives were not necessarily equal; therefore, the curvature was discontinuous at the nose. This additional problem was overcome by adding a small constant increment to the input second derivatives which would generate first derivatives at the nose that were more closely matched. The increment produced the same effect as a constant of integration, resulting in a very small global stretching or shrinking of the $\bar{y}$ coordinates. The value of the increment is determined iteratively using a simple Newton-Raphson technique which is very stable and generally converges in less than four iterations. The computer code for this iteration procedure is contained in subroutine YNEW in the airfoil smoothing program.

Camber and Thickness Distribution

By defining the smoothed airfoil shape with a cubic-spline function, the $\bar{y}$ coordinate and its derivatives can be computed at any desired θ-value with equations (30) through (32). Because of this capability, it was therefore possible to develop a method to compute a camberline and a thickness distribution for the smoothed airfoil. The equations for combining the camber and thickness distributions to obtain the upper surface coordinates of an airfoil are

$$x_u = x_c - t/2 \sin(\phi) \tag{33}$$

and $$y_u = y_c + t/2 \cos(\phi) , \tag{34}$$

and for the lower surfaces are

$$x_\ell = x_c + t/2 \sin(\phi) \tag{35}$$

$$y_\ell = y_c - t/2 \cos(\phi) \tag{36}$$

where x_c and y_c are the coordinates of the camberline, t is the local thickness, and ϕ is the local slope of the camberline. The airfoil generated with these equations will not be unique because a large number of other thickness and camber combining equations could be used to generate the same airfoil shape. However, given the shape of an airfoil, a unique camberline can be obtained which satisfies equations (33) through (36) by simply specifying that the absolute value of the slope at upper and lower points are equal in magnitude. The local slope is determined with respect to an axis system whose y-axis passes through the upper and lower surface points and whose x-axis passes through the mid-point of the line connecting the two points as illustrated in figure 2.

The equations for translating and rotating the input coordinates in the x- and y-axis system to the camberline $\hat{x}$- and $\hat{y}$-axis system are

$$\hat{x} = (\bar{x} - x_c) \cos(\phi) + (\bar{y} - y_c) \sin(\phi), \tag{37}$$

$$\hat{y} = (\bar{y} - y_c) \cos(\phi) - (\bar{x} - x_c) \sin(\phi). \tag{38}$$

The differentials with respect to $\bar{x}$ are

$$d\hat{x}/d\bar{x} = \cos(\phi) + \sin(\phi)\, d\bar{y}/d\bar{x} \tag{39}$$

$$d\hat{y}/d\bar{x} = \cos(\phi)\, d\bar{y}/d\bar{x} - \sin(\phi) \tag{40}$$

which combines to obtain the equation for the local slope

$$d\hat{y}/d\hat{x} = \frac{d\hat{y}/d\bar{x}}{d\hat{x}/d\bar{x}} = \frac{\cos(\phi)\, d\bar{y}/d\bar{x} - \sin(\phi)}{\sin(\phi)\, d\bar{y}/d\bar{x} + \cos(\phi)}, \tag{41}$$

where for a given set of upper and lower surface input points,

$$\phi = \tan^{-1}\left(\frac{\bar{y}_u - \bar{y}_\ell}{\bar{x}_u - \bar{x}_\ell}\right). \tag{42}$$

To determine the camberline simply requires that for either an upper or lower surface input point, an opposite surface point be located which satisfies the criteria that

$$\left|d\hat{y}/d\hat{x}\right|_u = \left|d\hat{y}/d\hat{x}\right|_\ell \tag{43}$$

The computer code for the camberline technique is contained in subroutine CAMTK. The execution procedure in this subroutine starts the search for the camberline at the upper surface trailing edge and proceeds in a counterclockwise direction toward the nose of the airfoil. A simply linear interpolation procedure is used to locate the corresponding lower surface point which satisfies the camberline criteria. The search for the lower surface point is performed with an interpolation interval of 1/2000th of the chord. After locating the lower surface point, execution continues to the next upper surface point and the search for the lower surface point begins at the previously located point. This cycle continues until all of the upper surface points have been used. The leading-edge point of the camberline (where thickness equals zero) is computed by fitting a second-order polynomial to the three previous camberline points in the nose region and then extrapolating to determine the intersection of the camberline with the input airfoil contour. The only noteworthy problem that has occurred with the use of this technique has been difficulty locating the first few camberline coordinates for airfoils with reflexed (upward-turned) camberlines near the trailing edge. This problem can generally be overcome by simply reversing the input order of the upper and lower surface coordinates to the smoothing program which means that the search for the camberline will be reversed proceeding clockwise along the lower surface from the trailing edge to the nose.

DESCRIPTION OF COMPUTER PROGRAM

The airfoil smoothing computer program AFSMO consists of a main program, fifteen subroutines, and two function subprograms and is listed in Appendix A. The airfoil scaling computer program AFSCL consists of a main program and two subroutines and is listed in Appendix B. A description of the input data requirements for the airfoil smoothing program is presented in Appendix C and a corresponding description of the output for a sample case presented in Appendix D. Likewise, a description of the input data requirements for the airfoil scaling program is presented in Appendix E and the description of the output in Appendix F. The primary input and output quantities and execution sequence of each main program and subroutine are described in this section.

Program AIRSMO

The primary function of the main program AIRSMO is to control the overall execution of the airfoil smoothing process. After specifying and computing several global program constants, calls are made to subroutines PSEUDO and LEROY to initialize the plot vector file SAVPLT for subsequent postprocess plotting on a variety of plotters at Langley. The subroutine INPUT is then called which reads and prepares the user-supplied input data. The subroutine SMOXY is then called which smooths the input airfoil coordinates. If punched output data are desired by the user, subroutine PCARD is then called. All punched data are written on output file TAPE1 which can be disposed of in any manner the user desires.

If plots of the coordinates, first and second derivatives, and curvature of the smoothed airfoil are desired, calls are then made to subroutine PLOTAF and PLOTCK. If the user also desires to compute the camber and thickness distribution of the smoothed airfoil, subroutine CAMTK is then called. Then, if the user desires to interpolate additional smoothed airfoil coordinates, subroutine INTP is called. This entire execution procedure is repeated until all input cases have been input and smoothed. A call is then made to subroutine CALPLT to finalize the plot vector file.

The following arrays must be dimensioned and constants defined or checked in this program:

TITLE	80-column title for input case
XINT	array containing $\bar{x}$ interpolation values
X,Y	arrays containing reordered $\bar{x}$ and $\bar{y}$ coordinates
W	array containing input weighing factors
YSMO	array containing smoothed $\bar{y}$ coordinates
YPS	array containing smoothed $\bar{y}'$ values
YPPS	array containing smoothed $\bar{y}''$ values
THETA	array containing θ-transformation values
PI	value of π
RAD	value of one radian π/180
CONS	value of constant K defined by equation (4)
JREAD	number of tape or file containing input data
JWRITE	number of tape or file containing output data
IPRINT	if equal to zero, the smoothing data generated during each iteration of the least-squares polynomial smoothing process in subroutine SMOXY and the interpolated data in PLOTAF and PLOTCK will be output
EPS	convergence criteria used during least-squares polynomial smoothing process in subroutine SMOXY
DF	standard deviation used during least-squares cubic spline smoothing process in subroutines SMOXY and CSDS

IERR if a nonzero value appears following a call to subroutine INPUT, it indicates that another case follows; and if it appears following a call to subroutine SMOXY, an error has occurred

Subroutine INTER

Subroutine INTER is a utility subprogram used to interpolate a y-value at a given x-value from an input table of x- and y-values. The interpolation can be performed using either a linear (straight line) or a weighted quadratic-equation fit of the y-values in the interpolation interval. The only restrictions are that the input table of x-values be single-valued and monotonically increasing or decreasing and that, for the weighted quadratic-equation fit, the input table of x-values contain at least four values. The initial execution step in this subroutine is a search to determine the x-interval containing the desired interpolation x-value ($x_{i-1} \leq x \leq x_i$). For the weighted quadratic-equation method, three y-values are interpolated:

(1) y_s by fitting a straight line between x_{i-1} and x_i,

(2) y_1 by fitting a quadratic equation between x_{i-2}, x_{i-1}, and x_i, and

(3) y_2 by fitting a quadratic equation between x_{i-1}, x_i, and x_{i+1}.

The deviations between the quadratic-equation and straight-line interpolated y-values are

$$\varepsilon_1 = \left| y_1 - y_s \right| \quad \text{and} \quad \varepsilon_2 = \left| y_2 - y_s \right| \tag{44}$$

The final interpolated y-value is obtained by linear weighting of the two deviations so that

$$y = w_1 y_2 + w_2 y_1 \quad , \tag{45}$$

where $$w_1 = \frac{\Delta_1}{\Delta_1 + \Delta_2} \quad \text{and} \quad w_2 = \frac{\Delta_2}{\Delta_1 + \Delta_2} \tag{46}$$

$$\Delta_1 = \varepsilon_1(x - x_{i-1}) \text{ and } \Delta_2 = \varepsilon_2(x_i - x). \tag{47}$$

For the linear interpolation method, the interpolated y-value is simply equal to y_s.

The following is a description of the parameters in the argument list for this subroutine:

XINT	input interpolation x-value
YINT	output interpolated y-value
N	number of values in input x and y arrays
X and Y	arrays containing input x- and y-values
JSTART	array index to begin search for interval containing XINT
JEND	array index of x-interval containing XINT
ICD	if equal to 0, the weighted quadratic-equation method is used, and, if equal to 1, the linear method is used

In the airfoil smoothing program, subroutine INTER is called by subroutine BADPT which checks for bad input airfoil coordinates and by subroutine SMOXY during the search for inflection points in the final smoothed airfoil contour.

Subroutine INPUT

The primary functions of subroutine INPUT are to read and print the input airfoil data and to prepare the input data in the proper format for input to the smoothing program. A detailed description of the required input airfoil data and the various options available for plotting and punching the output data is presented in the user-guide given in Appendix C. After reading the input data from the file JREAD and writing on output file JWRITE and if desired, the next execution step is to call subroutine BADPT to check the upper and lower surface coordinates for obvious bad points. If no errors occur during the check for bad points and again if desired, subroutine TRNSRT is called to translate and rotate the input airfoil to an axis system coincident with the longest chord of the airfoil.

The next execution step is to reorder the input coordinates, which are input from the leading edge to the trailing edge for each surface, from the lower surface trailing edge clockwise around the airfoil to the upper surface trailing edge. The reordered coordinates are also nondimensionalized by the chord length and, at the same time, the equivalent transformation θ-values computed using equations (10) and (11). If, instead of x and y coordinates, the $\bar{y}$ coordinates, $\bar{y}'$ values, or $\bar{y}''$ values as a function of θ are input, the equivalent $\bar{x}$ values are computed using equations (3) and (5).

The following input quantities are defined in this subroutine:

ITER	allowable number of smoothing iterations
IPLOT	plotting option
IPUNCH	punch output option
IOP	input airfoil coordinate option
ICAMTK	camber and thickness distribution option
INTR	interpolation option
IBAD	bad coordinate check option
ITRN	translation and rotation option
YLTE, YNOSE, YUTE	input desired $\bar{y}$ coordinates at the lower surface trailing edge, the nose, and the upper surface trailing edge, respectively

NINT	number of input interpolation $\bar{x}$ values
CNEW	desired chord of interpolated $\bar{y}$ coordinates (all $\bar{y}$ coordinates computed in subroutines INTP are multiplied by CNEW)
NP	number of elements in output arrays X, Y, W, THETA, YPS, and YPPS
NOSE	array index of nose point after reordering the coordinate
CHORD	computed longest chord length
IERR	if not equal to zero, the last input case has been read or an error occurred during the calls to subroutine IBAD
TITLE	input 80-column title
X	output array containing reordered $\bar{x}$ coordinates
Y	output array containing reordered $\bar{y}$ coordinates for IOP=0 or 1
W	output array containing reordered weighing factors
THETA	output array containing equivalent θ values
YPS	output array containing $\bar{y}'$ values for IOP=2
YPPS	output array containing $\bar{y}''$ values for IOP=3

The following arrays and constants are used internally in this subroutine:

XL	array containing input lower surface x coordinates if IOP=0 and θ-values if IOP=0.
YL	array containing input lower surface y coordinate if IOP=0, $\bar{y}$ coordinates if IOP=1, $\bar{y}'$ values if IOP=2, and $\bar{y}''$ if IOP=3
WL	array containing input lower surface weighting factors

XU,YU,WU	same as XL, YL, and WL except for upper surface
NL	number of elements in XL, YL, and WL arrays
NU	number of elements in XU, YU, and WU arrays
ITRMAX	maximum number of allowable smoothing iterations
TOLR	allowable deviation between input and interpolated $\bar{y}$ coordinate in subroutine BADPT
NMAX	maximum number of NU or NL values

Subroutine TRNSRT

The function of subroutine TRNSRT is to translate and rotate the input airfoil coordinates to an axis system coincident with the longest chord. The longest chord is defined as the distance from the trailing-edge bisector to the farthest input coordinate in the nose region of the airfoil. The translation and rotation equations are identical to equations (37) and (38) where x_c and y_c are the nose coordinates and ϕ is the angle between the longest chord and the input x-axis. After the input coordinates have been translated and rotated, the input coordinate and weighing factor arrays are reloaded with the newly defined transformed values. The following parameters are used internally in this subroutine:

ANGLE	computed angle of longest chord and input x-axis
XNOSE, YNOSE	computed nose coordinate of longest chord
XTE, YTE	computed coordinates of trailing-edge bisector of longest chord

Subroutine BADPT

The function of subroutine BADPT is to identify and possibly to correct input $\bar{y}$ coordinates whose corresponding interpolated values exceeds a specified tolerance. The user may execute a call to this subroutine by specifying a nonzero value for the parameter IBAD in subroutine INPUT; however, the call should be made only if the user has a concern about possible bad points or excessive waviness in the input coordinates. Following entry to this subroutine the θ equivalent of each input $\bar{x}$ coordinate is computed for use during the interpolation process. Then for each input $\bar{y}$ coordinate, a corresponding interpolated value is obtained using the weighted quadratic-equation method of subroutine INTER with input arrays loaded with the remaining $\bar{y}$ coordinates and θ values. (Note that the input $\bar{y}$ coordinate itself is not loaded.) If the deviation between the input and interpolated $\bar{y}$ coordinate exceeds a specified tolerance, the interpolated $\bar{y}$ coordinate is flagged as being out-of-tolerance, the interpolated value substituted, and then execution continues to the next point. If, however, during this interpolation process, two consecutive points are found to be out-of-tolerance, an error flag is set which will terminate the execution of the particular input case. The following additional parameters are used in this subroutine:

X,Y	input arrays containing either upper or lower surface $\bar{x}$ and $\bar{y}$ coordinates
ISURF	if equal to 1, indicates upper surface coordinates input, and, if equal to 2, lower surface
TI	work array containing all θ values except value at desired interpolation point
YI	work array containing all $\bar{y}$ coordinates except value at the desired interpolation point
YN	temporary array containing interpolated $\bar{y}$ coordinates
IERR	if output with a nonzero value, two adjacent points are out-of-tolerance

Subroutine SMOXY

The primary function of subroutine SMOXY is to perform the iterative smoothing process and is, therefore, the most important subroutine in the entire airfoil smoothing program. The basic inputs to this subroutine are the initial $\bar{x}$ and $\bar{y}$ coordinates, either $\bar{y}'$ or $\bar{y}''$, the transformed θ values, weighting factors for each input point, and the input option parameter IOP which specifies the type of input data. If either $\bar{y}'$ or $\bar{y}''$ are input instead of the $\bar{y}$ coordinates, the desired trailing edge and nose $\bar{y}$ coordinates must also be input.

After entry to the subroutine, the input option parameter is checked to determine the type of input data. If the first derivatives $\bar{y}'$ are input (IOP = 2), two sets of second derivatives $\bar{y}''$ are computed. One set is computed using the least-squares polynomial smoothing method (subroutine LSQSMO) and the second set, using the least-squares cubic-spline method (subroutine CSDS). Each set of second derivatives and the desired trailing-edge and nose $\bar{y}$ coordinates are then input to subroutine YNEW which computes a corresponding set of $\bar{y}$ coordinates. These $\bar{y}$ coordinates and their corresponding second derivatives are then used to compute a new set of first derivatives using the spline equation (31). The $\bar{y}$ coordinates and the sum-of-the-squares of the difference between the original input and computed first derivatives are then computed for each set and the set with the smallest sum is chosen for subsequent smoothing.

If the second derivatives $\bar{y}''$ and the desired trailing-edge and nose $\bar{y}$ coordinates are input (IOP=3), a corresponding set of $\bar{y}$ coordinates are computed with subroutine YNEW and a set of first derivatives computed with spline equation (31). Then, regardless of the input option, program execution proceeds to the iterative smoothing process. Prior to the start of this iteration cycle, a search is made of the upper and lower surface $\bar{y}$ coordinates to determine the maximum upper surface and minimum lower surface values. During each smoothing cycle, these two coordinates are heavily weighted in an attempt to insure that the maximum thickness of the final smoothed airfoil is reasonably close to that of the original input airfoil.

As discussed in the method section of this report, the initial step in the smoothing process is to determine the smoothed second derivatives of the input $\bar{y}$ coordinates using an iterative piecewise least-squares polynomial smoothing method. During this iteration process, each call to subroutine LSQSMO produces a new set of $\bar{y}$ coordinates and their corresponding first and second derivatives. The next step in the iteration process is to compute the sum-of-the-squares of the difference between the current and previous set of second derivatives and then to check the sum to insure that the current value is less than the previous value. This will determine whether or not the iteration process is converging. If the process is diverging, the iteration cycle is terminated, an appropriate error message printed, and execution proceeds to the next step. If the process is converging, the next iteration input $\bar{y}$ coordinates for subroutine LSQSMO are computed using the following weighting procedure:

$$\text{If} \quad \Delta_{i-1} = \left[\bar{y}_N - \bar{y}_I\right]_{i-1}$$

$$\text{and} \quad \Delta_i = \left[\bar{y}_N - \bar{y}_I\right]_i \tag{48}$$

and if the sign or magnitude of Δ_i equals Δ_{i-1}, then

$$(\bar{y}_N)_{i+1} = \frac{1}{2}(\bar{y}_I + \bar{y}_N)_i \tag{49}$$

and, if not, the Newton-Raphson formula

$$(\bar{y}_N)_{i+1} = (\bar{y}_I)_{i-1} - \left(\frac{\Delta_{i-1}}{\Delta_i - \Delta_{i-1}}\right) \cdot \left[(\bar{y}_I)_i - (\bar{y}_I)_{i-1}\right] \tag{50}$$

is used, where i is the iteration number, I indicates input value, and N indicates new value computed by LSQSMO. After computing the new weighted coordinates, the sum-of-the-squares difference of the second derivatives is checked to see if it is less than the specified convergence value EPS. However, if the value of the difference sum is greater than the convergence value, the iteration cycle is repeated. If the value has converged or the iteration cycle begins to diverge, program execution proceeds to the next step which is to smooth the second derivatives one additional time using the least-squares cubic-spline method of subroutine CSDS. The additionally smoothed second derivatives and the final trailing-edge and nose $\bar{y}$ coordinate from the piecewise least-square polynomial smoothing process are then input to subroutine YNEW which computes a corresponding final set of smoothed $\bar{y}$ coordinates.

The final smoothed coordinates are then checked for relative smoothness by another call to LSQSMO with all the coordinate weighting factors set equal to 1.0. The next execution step is to compute a corresponding set of final smoothed first derivatives using spline equation (31). Then the final smoothed first and second derivatives with respect to $\bar{x}$ and the curvature are computed and printed in addition to the original input and final smoothed coordinates and the final smoothed first and second derivatives $\bar{y}'$ and $\bar{y}''$.

Following the detailed printout step, a check is made for negative thickness or crossover between the upper and lower surface near the trailing edge of the airfoil. During the least-squares polynomial smoothing process, the input weighting for the trailing-edge coordinates are multiplied by a factor of 7 to help ensure that the final smoothed airfoil has the same trailing-edge thickness as the original input airfoil. In spite of this additional weighting, the final smoothed airfoil will often have negative trailing-edge thickness; especially if the input airfoil has zero or a very small trailing-edge thickness. If a crossover is discovered during this step, an error message is printed, an error flag set, and execution returned to the calling program.

If no crossover is discovered, the next and final step is to determine the location of all inflection points (i.e. $\bar{y}' = 0$) in the final smoothed airfoil. This step is accomplished by checking each θ-interval of the final airfoil for θ locations where the first derivative spline equation (31) is equal to zero. This equation can be written as the quadratic equation

$$a\theta^2 + b\theta + c = 0 \tag{51}$$

with

$$a = \left(\frac{\bar{y}_i'' - \bar{y}_{i+1}''}{2h_i}\right)$$

$$b = \left(\frac{\bar{y}_{i+1}''\theta_i - \bar{y}_i''\theta_{i+1}}{h_i}\right) \tag{52}$$

$$c = \left(\frac{\bar{y}_i''\theta_{i+1}^2 - \bar{y}_{i+1}''\theta_i^2}{2h_i}\right) + \frac{h_i}{6}(\bar{y}_{i+1}'' - \bar{y}_i'') - \left(\frac{\bar{y}_{i+1} - \bar{y}_i}{h_i}\right)$$

where $h_i = \theta_{i+1} - \theta_i$. The real solutions to this equation which lie within the θ-interval are the inflection points. All inflection point locations and the results of the final smoothness check are then printed and control returned to the calling program.

A description of the parameters in the argument list for this subroutine is presented in the description of program AIRSMO and the subroutine INPUT. The following parameters are used internally:

WT	multiplier for weighting of maximum thickness coordinates
YPP and YPPU	work arrays containing current values of $\bar{y}''$
YUSMO and YN	work arrays containing current values of $\bar{y}$
WK, A, and DUM	internal work arrays
SUMY	array containing sum-of-squares differences from least-squares polynomial smoothing process
JMAXL and JMAXU	array index values for the minimum lower surface $\bar{y}$ and for the maximum upper surface $\bar{y}$, respectively
GP and GPP	$d\bar{x}/d\theta$ and $d^2\bar{x}/d\theta^2$
DYDX and DY2DX	$d\bar{y}/d\bar{x}$ and $d^2\bar{y}/d\bar{x}^2$
CURV	curvature k
RLE	leading-edge radius (1/k at nose)

Subroutine YNEW

The primary function of subroutine YNEW is to control the iterative procedure that computes a set of new $\bar{y}$ coordinates from an input set of second derivatives and desired trailing edge and nose coordinates. The new set of coordinates can be computed using two different solution approaches. For the first approach (IPT = 0), the resultant simultaneous cubic-spline equations solved are generated using the combined upper and lower surface second derivatives and setting the end conditions equal to the leading- and trailing-edge coordinates. The value of the first derivative at the nose will, of course, be the same whether approached from either the upper or lower surface; however, the $\bar{y}$-coordinate at the nose may differ from the desired input value. The desired input nose coordinate can be obtained by adding a small constant incremental value to the input second derivatives. This small value acts the same as a constant of integration resulting in a small stretching or shrinking of the computed $\bar{y}$ coordinates. The incremental value is determined in this subroutine using the simple iterative Newton-Raphson equation

$$\Delta x_{i+1} = \Delta x_i - \frac{f(\Delta x_i)}{f'(\Delta x_i)} \tag{53}$$

where Δx represents the incremental value, $f(\Delta x)$ the difference between the desired and computed nose coordinates, $f'(\Delta x)$ the slope of the difference curve (determined using simple differencing), and i the iteration number.

For the second approach (IPT = 1), the resultant simultaneous cubic-spline equations solved are generated in a piecewise manner first using the lower surface second derivatives and setting the end conditions equal to the trailing-edge and nose coordinates, and then using the corresponding quantitites for the upper surface. This approach ensures, of course, that the resultant airfoil will have the desired nose coordinate; however, the slope at the nose may differ when approached from the upper and lower surfaces. Here again, like the first approach, a better match can be obtained by adding a small incremental value to the input second derivatives. This increment is determined using the same iterative Newton-Raphson equation as that used for the first approach except the Δx represents the difference between upper and lower surface first derivatives at the nose. Both approaches should theoretically produce the same incremental values; however, experience has shown that the convergence of the second approach is generally quicker and more stable.

The following additional parameters are used internally in this subroutine:

DUM and WK	internal work arrays
DELTA	incremental value added to second derivatives

Subroutine INVY

The function of this subroutine is to compute a set of $\bar{y}$ coordinates from an input set of second derivatives and desired $\bar{y}$ coordinates at the start and end of the set. The input second derivatives and transformation θ-values are used to compute a matrix of simultaneous equations using the cubic-spline equation (29). The resultant matrix is tridiagonal with two less equations than unknowns and relates the second derivatives $\bar{y}''$ and the corresponding $\bar{y}$ coordinates. The two remaining unknowns are specified as the desired $\bar{y}$ coordinates at the start and end of the set. The solution of the resultant matrix is greatly simplified because only the diagonal elements d_i and the two adjacent elements e_i and f_i differ from zero. Using the Crout reduction method described in reference 6, the solution becomes a simple back substitution

$$\bar{y}_N = \bar{c}_N \text{ for } i=N$$

and $\bar{y}_i = \bar{c}_i - \bar{f}_i \bar{y}_{i+1}$ for $i = N-1, N-2, \ldots 1$ (54)

where

$$\bar{d}_i = d_i - e_i \bar{f}_{i-1}$$

$$\bar{f}_i = f_i / \bar{d}_i \quad (55)$$

and $\bar{c}_i = \dfrac{c_i - e_i \bar{c}_{i-1}}{\bar{d}_i}$.

The tridiagonal terms from equation (29) are

$$e_i = 1/h_{i-1}$$

$$d_i = -1/h_{i-1} - 1/h_i$$

$$f_i = 1/h_i \quad (56)$$

and $c_i = \left(\dfrac{h_{i-1}}{6}\right) \bar{y}''_{i-1} + \left(\dfrac{h_{i-1} + h_i}{3}\right) \bar{y}''_i + \left(\dfrac{h_i}{6}\right) \bar{y}''_{i+1}$

At the ends the coefficient terms are

$$d_1 = 1, \; f_1 = 0, \; c_1 = \bar{y}_1 \quad (57)$$

and

$$e_N = 0, \; d_N = 1, \; c_N = \bar{y}_N \; .$$

The following is a description of the parameters in argument list for this subroutine:

X	input array containing θ values
YPP	input array containing $\bar{y}''$ values
NS	index of start element
NE	index of end element
Y	output array containing $\bar{y}$ coordinates
YSTART	desired $\bar{y}$ coordinate at start
YEND	desired $\bar{y}$ coordinate at end
A	internal work array

Subroutine LSQSMO

The function of this subroutine is to smooth and compute the second derivatives of an input set of $\bar{y}$ coordinates using the piecewise least-squares polynomial method described in the previous method section. The subroutine smooths each coordinate by fitting a least-squares polynomial of the 4th degree through the input coordinate and six adjacent coordinates. If possible, the six coordinates used are the three coordinates just prior to and the three just after the input coordinate; otherwise, six consecutive coordinates are used. Prior to the execution of the smoothing process, a check is made of the three corresponding upper and lower surface coordinates adjacent to the nose coordinate to determine whether or not the input airfoil is symmetric about the θ-axis in the nose region. If the airfoil is symmetric in the nose, the smoothing process is performed in the clockwise direction for the upper surface and counterclockwise for the lower surface; otherwise, it is performed clockwise for both surfaces.

During the smoothing process, each coordinate is given the specified input weighting factor and the six adjacent coordinates are given a weighting of 1.0. The maximum and minimum thickness coordinates are also given an additional weighting equal to the parameter WT times the input value. In a similar manner, the upper and lower surface trailing-edge coordinates are given an additional weighting of 7 times the input value. After computing the coefficients of the least-squares polynomial for each coordinate, a new $\bar{y}$-coordinate value, the first-, and the second-derivatives are computed using equation (17), (18), and (19), respectively.

The following is a description of the parameters in the argument list and the internally used arrays and constants:

X	input array of θ values
Y	input array of $\bar{y}$ values
W	input array of weighting factors
YN	output array of smoothed $\bar{y}$ coordinates
YP	output array of first derivatives $\bar{y}'$
YPP	output array of second derivatives $\bar{y}''$
N	number of input coordinates
IMAX and JMAX	array index of maximum and minimum thickness coordinates
NOSE	array index of nose coordinate
WT	additional weighting factor for maximum and minimum thickness coordinate

EPS	allowable deviation between corresponding upper and lower surface θ and $\bar{y}$ values in the nose region
ISYM	if equal to zero, input airfoil is symmetric in nose region
XI, YI, WI	arrays containing 7 consecutive values of θ, $\bar{y}$, and w
A	array containing elements of symmetric least-squares matrix
B	array containing coefficients of resultant 4th order least-squares polynomial

Subroutine CSDS

The function of subroutine CSDS is to fit a least-squares cubic spline through a set of input θ values and either the $\bar{y}$ coordinates or the second derivative $\bar{y}''$. A very detailed description of theory and computer coding associated with this subroutine is presented in reference 5 and, therefore, will not be presented in this report. This subroutine is also a part of the standard math-library subprogram package on the Langley CDC computer system and is identified by the same call name and parameter list. A complete description of the input and output parameters are presented at the beginning of the listing of the subroutine in Appendix A.

Subroutine PCARD

The function of subroutine PCARD is to write the final smoothed data on an output file (TAPE1) for postprocess disposal to a desired output device. The case title is written on the output file initially and is followed by a card image containing the value of the input option (IOP parameter) corresponding to the output option

(IPUNCH parameter). Then for the upper and lower surface, the number of coordinates is written on the output file followed by one of four types of smoothed output data as specified by the value of the output option parameter IPUNCH. The four types of output data are as follows:

IPUNCH = 1	x-coordinate, smoothed y-coordinate, and weighting
IPUNCH = 2	θ-value, smoothed $\bar{y}$ coordinates, and weighting
IPUNCH = 3	θ-value, smoothed $\bar{y}'$, and weighting
IPUNCH = 4	θ-value, smoothed $\bar{y}''$, and weighting

If IPUNCH equals 3 or 4, the $\bar{y}$ coordinates of the lower surface trailing edge, the nose, and the upper surface trailing edge are also written on the output file. All data are written on the output file in a format suitable for input to the airfoil smoothing program. Except for the IPUNCH parameter, all other parameters in the argument list are fully defined in the description of subroutine INPUT.

Subroutine PLOTAF

The function of subroutine PLOTAF is to plot the input and smoothed $\bar{y}$ coordinates, smoothed $\bar{y}'$, and smoothed $\bar{y}''$ versus the θ values (IPLOT=1) and to plot the input and smoothed $\bar{y}$ coordinates versus the input $\bar{x}$ coordinates (IPLOT=2). All plots are scaled for postprocess plotting on the Langley 33-inch CALCOMP drum plotters. The called subroutines CALPLT, NOTATE, AXES, PNTPLT, LINE, and NFRAME are all part of the Langley plotting subroutine package and are available by attaching the CALCOMP direct-access library file. Prior to plotting the smoothed $\bar{y}$ and $\bar{x}$ coordinates and the smoothed $\bar{y}'$ values, additional values are interpolated at each degree of θ from -180 to +180 degrees. The ordinate axes are automatically scaled to insure that all input values will be plotted. A sample of the two types of plots generated by this subroutine is presented in figure 3 for IPLOT=1 and in figure 4 for IPLOT=2. Except for the IPLOT parameter, all other input parameters are fully defined in the description of subroutine INPUT.

Subroutine PLOTCK

The function of subroutine PLOTCK is to plot the square root of the local smoothed curvature versus the θ-transformation value (IPLOT=3). Prior to plotting the curvature, additional values are interpolated at each one-half degree of θ from -180 to +180 degrees. By plotting the square root of the curvature rather than just the curvature, the very large curvature peaks in the nose region of the airfoil are reduced and the normally low curvatures in the trailing-edge regions are increased and, as a result, a more evenly proportioned plot is generated. A sample of the type of plot generated by this subroutine is presented in figure 5. All input argument parameters are fully defined in the description of subroutine INPUT.

Subroutine CAMTK

The function of subroutine CAMTK is to compute the camber and thickness distribution of the final smoothed airfoil. A detailed explanation of the method used to compute the camberline is presented in the method section of this report. The first execution step in the subroutine is to load the x and y coordinates and y values into separate arrays for the upper and lower surfaces from the nose to the trailing edge. The first derivatives dy/dx are then computed at each input point on the upper surface.

The next execution step is the search for the camberline. As previously stated in the method section, the search begins at the upper surface trailing-edge point and proceeds counterclockwise along the upper surface to the nose point. At each upper surface point, a simple linear interpolation procedure is used to locate the corresponding lower surface point that satisfies the camberline criteria of equal magnitudes of the local upper and lower surface slopes with respect to an axis system aligned with the local camberline. The search for the lower surface point is performed with an interpolation interval of 1/2000th of the chord. After locating the lower surface point, execution continues to the next upper surface point and the search begins on the lower surface at the previously located point and proceeds clockwise toward the nose point.

After completing the camberline search for each point on the upper surface, the next execution step is to locate the intersection of the camberline with the airfoil leading edge which is the location of zero thickness. This intersection is found by fitting a second-order polynomial to the previous three camberline coordinates and then extrapolating to find the intersection with the nose region which is defined with cubic-spline functions. The upper surface coordinates, corresponding lower surface coordinates, camberline coordinates, thickness, and slope of the camberline are printed at each step during the search for the camberline and the nose intersection points. An error term is also printed for each point and represents the absolute value of the difference between the local slopes of the upper and lower surface camberline search points with respect to the local camberline-axis system.

The next execution step is to write the camber and thickness distribution data on an output file (TAPE1) for possible input to the airfoil scaling program AFSCL. This execution step is activated only if the value of the IPUNCH input parameter equals 5. The final execution step, if the value of the input KPLOT parameter is nonzero (IPLOT = 4, 8, 9, or 10), is to plot the camber and thickness distribution data. A sample of the type of plot generated is presented in figure 6. The camberline coordinates are plotted at the bottom part of the figure, the half-thickness distribution at the center, and the upper and lower surface search points at the top part of the figure.

A description of the parameters in the argument list for this subroutine is presented in the description of program AIRSMO and subroutine INPUT. The following parameters are used internally:

TU and TL	temporary arrays containing input upper and lower surface θ-values from nose to trailing-edge points.
YU and YL	temporary arrays containing input upper and lower surface smoothed $\bar{y}$ coordinates
YPPU and YPPL	temporary arrays containing input upper and lower surface $\bar{y}''$ values
DYXU	array containing $d\bar{y}/d\bar{x}$ values for upper surface

XLS and YLS	arrays containing $\bar{x}$ and $\bar{y}$ coordinates of lower surface camberline search points
TH	array containing value of slope of camberline
XC and YC	arrays containing x_c and y_c coordinates of camberline
TK	array containing the half-thickness values
NM	number of interpolated points allowed on the lower surface
NT	number of camberline coordinates
DU and DL	slope of the upper and lower surface search points with respect to the local camberline axis system

Subroutine INTP

The function of subroutine INTP is to interpolate additional smoothed airfoil coordinates. This subroutine is called if the user specifies a value of 1 or 2 for the parameter INTR read by subroutine INPUT. If the value of INTR equals 1, the interpolation is is performed at a standard set of 57 $\bar{x}$ values loaded internally in the subroutine and defined as follows:

$\bar{x}$ = 0.0, 0.00025, 0.0005, 0.00075, 0.001, 0.0015, 0.002, 0.0025, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375, 0.4, 0.425, 0.45, 0.475, 0.5, 0.525, 0.55, 0.575, 0.6, 0.625, 0.65, 0.675, 0.7, 0.725, 0.75, 0.775, 0.8, 0.825, 0.85, 0.875, 0.9, 0.925, 0.95, 0.97, 0.98, 0.99, 1.0.

If the value of INTR equals 2, the desired $\bar{x}$ values are input by the user and may include up to 100 values as specified by the parameter NINT. The interpolation is performed for the upper and then the lower surfaces using the cubic-spline equations (30), (31), and (32). The derivatives $d\bar{y}/d\bar{x}$ and $d^2\bar{y}/d\bar{x}^2$ and the curvature are also computed and printed for each $\bar{x}$ value. The user must also input a value for the parameter CNEW which is the desired value of the chord. The $\bar{x}$ and $\bar{y}$ interpolated coordinates are multiplied by CNEW and printed as x and y coordinates. If the value of the parameter IPUNCH equals 6, the interpolated x and y coordinates are written on the output file (TAPE1) for postprocess disposal to a desired output device. A description of the parameters in the argument list for this subroutine is presented in the description of program AIRSMO and subroutine INPUT.

Subroutine COORD

The function of subroutine COORD is to interpolate a value for for $\bar{y}$, $d\bar{y}/d\bar{x}$, $d^2\bar{y}/d\bar{x}^2$, and the curvature at a specified value of θ using the cubic-spline equations (30), (31), and (32). The following subroutine constants are used internally:

TI	input θ value
YI	interpolated $\bar{y}$-coordinate
DYDX	interpolated first derivative $d\bar{y}/d\bar{x}$
DY2DX	interpolated second derivative $d^2\bar{y}/d\bar{x}^2$
CURV	interpolated curvature

Function Subprograms SINH and COSH

The function of these two function subprograms is to compute the hyperbolic sine and cosine in terms of the exponential function. The relationships are

$$\sinh(x) = \frac{e^{x} - e^{-x}}{2} \tag{58}$$

and

$$\cosh(x) = \frac{e^{x} + e^{-x}}{2} \quad , \tag{59}$$

respectively.

Program SCALE

The primary function of program SCALE is to read the input data and control the execution of the airfoil scaling process. The camber and thickness distribution data input to this program are generated by the subroutine CAMTK in the airfoil smoothing program AFSMO. After specifying and computing several global program constants, the first execution step is to read the input data. A detailed description of the input data and the required formats are discussed in the user-guide presented in Appendix E. After reading the input data, calls are made to subroutines PSEUDO and LEROY to initialize the plot vector file SAVPLT for subsequent postprocess plotting on a variety of plotters at Langley. The input x_c coordinates of the input camberline are then checked to insure monotonically increasing order. The equivalent θ value for each camberline x_c coordinate is then computed.

The next execution step is to compute the x_c location and the magnitude of the maximum value of the input half-thickness distribution. A cubic spline is fit through the input thickness data and then all locations and corresponding thickness values where the first derivative of the spline function equals zero are computed using equations (51) and (52). The location of the maximum value is then determined and printed on the output file. If the value of the input parameter IOP equals 1, the slope of the camberline coordinates are then computed using spline equation (31). The angular value of the slope is then obtained by computing the arctangent of the value of the first derivative.

The next step is to call the scaling subroutine SCTK to generate first the coordinates of the airfoil with the input maximum thickness-chord ratio and then the coordinates of the airfoil with each of the desired scaled maximum thickness-chord ratios. The final execution step is to call subroutine CALPLT to finalize the plot-vector file SAVPLT.

The following arrays must be dimensioned and constants defined in this program:

XC and YC	arrays containing input x_c and y_c coordinates of the camberline
TK	array containing input half-thickness distribution t/c/2
TH	array containing input camberline slopes ϕ
THETA	array containing computed θ values
YPP	array containing computed second derivatives $\frac{d^2y_c}{dx_c^2}$
TKNEW	array containing input values of desired maximum thickness-chord ratios
TITLE	80-column title for input case
VAR and WK	work arrays
JWRITE	number of tape or file containing output data

JREAD	number of tape or file containing input data
NTMAX	maximum number of allowable elements in TKNEW array
PI	value of π
RAD	value of one radian $180/\pi$
CONS	value of constant K defined by equation (4)
NT	number of elements in XC, YC, TK, and TH arrays
IOP	camberline slope option
IPLOT	plotting option
IPUNCH	punch output option
LT	number of desired input maximum thickness values
TKMAX	value of the maximum thickness-chord ratio of the input thickness distribution
DELTA	x_c location of TKMAX
IERR	if nonzero, error occurred during generation of scaled airfoil in subroutine SCTK

Subroutine SCTK

The function of subroutine SCTK is to scale the coordinates of an input airfoil from the input maximum thickness-chord ratio to a new desired maximum thickness-chord ratio. The first execution step is to generate the coordinates of the baseline airfoil by combining the input camber and the scaled thickness distributions using equation (33) and (34) for the upper surface and equations (35) and (36) for the lower surface. Each scaled thickness distribution is obtained by multiplying the input thickness distribution by the ratio of the desired-to-input maximum thickness-chord ratio. This procedure is simple; however, several problems may occur which require special handling.

If the value of the input camber distribution is nonzero in the trailing-edge region, the airfoil generated may not have either an upper or lower surface $\bar{y}$ coordinate at the trailing-edge location where $\bar{x}$ equals 1.0. To eliminate this problem, a second-order polynomial is fit to the last three computed coordinates near the trailing edge on each surface and a new $\bar{y}$ coordinate either extrapolated or interpolated at $\bar{x}$ equals 1.0. Also, if the camber distribution is nonzero in the nose region, the airfoil generated may have $\bar{x}$ coordinates that are less than 0.0. This problem is eliminated by translating and stretching or shrinking the coordinates of the airfoil so that the nose of the adjusted airfoil is at $\bar{x}$ equals 0.0 and the trailing edge at $\bar{x}$ equals 1.0. The only other problem that may occur is the possible generation of either upper or lower surface $\bar{x}$ coordinates that are not monotonically increasing from nose to trailing edge. This particular problem cannot be eliminated; therefore, a check is made to see if it occurred and, if so, an error message is printed, an error flag set, and execution returned to program SCALE.

The upper and lower surface $\bar{x}$ and $\bar{y}$ coordinates are multiplied by the value of the parameter CNEW and then loaded into separate arrays from the nose to the trailing edge. The coordinates, input camber distribution, and scaled thickness distributions are then printed. If the IPUNCH parameter is nonzero, the scaled airfoil coordinates are then written on the output file TAPE1 in a format suitable for input to the smoothing program. If the IPLOT parameter is nonzero, the next and final execution step is to plot the scaled airfoil and its corresponding camber and thickness distributions as illustrated in figure 7. A description of the parameters in the argument list for this subroutine is presented in the description of program SCALE.

Subroutine CUBSPL

The function of subroutine CUBSPL is to fit a cubic spline through an input set of x and y values. The input data are used to compute a matrix of simultaneous equations using the cubic spline equation (29) with the unknowns being the second derivatives at each input point. This tridiagonal matrix has two less equations than unknowns; therefore, the second derivative at end points of the data set must be specified. In this subroutine second derivatives at the end points are computed by fitting a second-order polynomial of the form

$$y = ax^2 + bx + c \tag{60}$$

to each end point and its two adjacent points and then differentiating to determine the second derivative which is

$$d^2y/dx^2 = 2a \tag{61}$$

The Crout reduction method, which is discussed in the description of subroutine INVY, is used to solve the matrix for the remaining second derivative. The tridiagonal matrix terms are

$$e_i = h_{i-1}/6$$

$$d_i = \frac{h_{i-1} + h_i}{3} \tag{62}$$

$$f_i = h_i/6$$

and

$$c_i = \left(\frac{y_{i+1} - y_i}{h_i}\right) - \left(\frac{y_i - y_{i-1}}{h_{i-1}}\right) .$$

The following parameters are used in this subroutine:

X and Y	array containing input x and y values
YPP	array containing computed second derivatives d^2y/dx^2
N	number of elements in X Y, and YPP arrays
A	work array dimensioned by 2 times N in the calling program

DISCUSSION OF PROGRAM APPLICATION AND RESULTS

The airfoil smoothing program was formulated to smooth the coordinates of airfoil-type contours which are characteristically round in the front and sharp or blunt in the rear. Several users in the past have attempted to use this program to smooth nonairfoil shapes such as internal contours of engine nacelles or wind tunnels. These attempts have been generally unsuccessful because of the effects of the θ-transformation function which was formulated to stretch the x-axis in the leading- and trailing-edge regions. The smoothing program can be used successfully to smooth nonairfoil contours by redefining the θ-transformation function as

$$\theta = \pm\pi\bar{x} \tag{63}$$

and making the appropriate changes in the computer code.

An airfoil contour may be input into the smoothing program in several forms. The most widely used form is, of course, as x and y coordinates (IOP = 0) which have been obtained from actual measurements of an existing airfoil or from theoretical computations.

Regardless of the source of the coordinates, the user should strive to input a proportionally larger number of coordinates in regions of higher curvature which is generally the nose region for most airfoils. The user may input as many as 100 coordinates for each airfoil surface; however, it is recommended that no more than 35 to 40 coordinates be input for each surface because, in general, the more dense the coordinate spacing the more restricted the smoothing process will become. If the user desires to limit the extent of smoothing in a particular region, it is suggested that a few highly weighted coordinates be input rather than a large number of closely spaced coordinates.

The question often arises as to the number of smoothing iterations (ITER parameter) the user should specify. It is recommended that zero iterations be specified for the initial run of a new airfoil case. The plots generated during the initial run can then be examined to establish the initial smoothness of the airfoil, the suitability of the input x-coordinate spacing, and the possible existence of bad input y coordinates. During all subsequent runs, it is recommended that the maximum of 300 iterations be specified. The convergence criteria for this smoothing program is rather stringent; however, the smoothing process should converge or be near convergence in less than 100 iterations for most airfoils. If the process has not conveyed in 300 iterations, the resultant coordinates can be written on the output file TAPE1 in the form of either x and y coordinates or θ and $\bar{y}$ values and then input again into the smoothing program for another 300 iterations. If, during the initial smoothing attempt, the process begins to oscillate, it is suggested that fewer coordinates be selected in the region where the oscillation occurs and the case be resubmitted. The oscillatory region can be located by setting the IPRINT parameter in program AIRSMO equal to 0 which will generate a summary print of the computed second derivatives for each iteration.

The airfoil contour may also be input in the form of $\bar{y}$ coordinates and the corresponding θ-values (IOP = 2). This form is often used to resubmit a set of coordinates that required adjustment due to either bad or poorly defined nose $\bar{y}$ coordinates that are often revealed during the initial run of a new airfoil. The stretching effect of the θ-transformation function will highlight any coordinate discrepancies in the nose region of the airfoil.

Two additional input forms are available to modify or smooth an airfoil contour and are less direct than the previous two forms discussed. The two additional input forms consists of inputting the first $\bar{y}'$ (IOP = 3) or second $\bar{y}''$ (IOP = 4) derivatives as a function of the θ-transformation value. The corresponding $\bar{y}$ coordinates are obtained by solving a tridiagonal matrix of simultaneous cubic spline equations; therefore, local changes in the input derivatives have a less localized and more global effect in the computed $\bar{y}$ coordinates. Great care should be exercised when using either of these two input forms; especially the second derivative, because seemly small changes in the derivatives will very often result in rather large changes in the $\bar{y}$ coordinates. In spite of its sensitivity, these two input forms provide a very easy and direct method to reduce or eliminate waviness in the curvature of the final smoothed airfoil.

The airfoil smoothing program has been used extensively at Langley for the past several years and has worked successfully for a wide range of airfoil shapes. A comparison between the unsmoothed and smoothed first and second derivatives for a typical airfoil is presented in figure 8. The corresponding changes in the $\bar{y}$ coordinate are very small and are not distinguishable on a page-size plot of the airfoil contours. As illustrated in figure 9, the improvement in the smoothness of the curvature distribution is excellent.

Only two problems have occurred persistently during the past several years of program utilization. The first problem occurs when attempting to smooth airfoils with very sharp or zero-thickness trailing edges. Although the trailing-edge coordinates are heavily weighted, the smoothing process will often result in a small shift in the upper and lower surface trailing-edge coordinates. Many times the shift will be in the opposite direction and a negative trailing-edge thickness will occur. As previously discussed, the program checks for negative thickness and, if detected, will print an error message and proceed to the next input case. The most practical solution to this problem is simply to terminate the input coordinates very near the trailing edge at a point with small finite thickness. The second problem, as noted in the method section of this report, is a difficulty in locating the first few camberline coordinates of an airfoil with a reflexed (upward-turned) camberline near the trailing edge. This problem can generally be overcome by simply reversing the input order of the coordinates so that lower surface coordinates are input first, followed by the upper surface coordinates. This will not affect the smoothing process, but will cause the camberline search procedure to reverse surfaces.

CONCLUDING REMARKS

The airfoil computer programs AFSMO and AFSCL described in this report have been used successfully at Langley for several years to smooth and scale a wide variety of airfoil shapes generated by various theoretical methods or measured from existing airfoil models and wing panels. The smoothing process is very stable and generally converges in less than a hundred iterations. The smoothing program user-supplied input requirements are very simple and consist of basically specifying the title, input/output options, and the upper and lower surface coordinates. The camber-line search procedure in the smoothing program generates the basic camber and thickness distribution data needed as input to the scaling program. The only additional user-supplied input for the scaling program are a title, input/output option, and the number of and the values for the desired maximum thickness-chord ratios.

The output plots generated by the smoothing program are very helpful during the analysis and possible modification of the smoothed airfoil. After several years of extensive use by Langley personnel, no appreciable execution errors have occurred or airfoil shape limitations been revealed. The use of the AFSMO program to smooth nonairfoil shapes should not be attempted without redefinition of the x-axis transformation function. Both programs were coded for use on the Langley CDC CYBER computers. No specialized system software is needed to execute either program and all required subroutines are listed in this report except for several basic CALCOMP plotting subroutines which are unique to the Langley computers. Both programs have been successfully converted for use on other computer systems; however, double-precision accuracy was necessary for the conversion of the smoothing program because of its very stringent convergence criteria.

APPENDIX A

COMPUTER LISTING OF AIRFOIL SMOOTHING PROGRAM AFSMO

This appendix contains a computer listing of the airfoil smoothing program AFSMO which consists of a main program, fifteen subroutines, and two function subprograms.

CARD NO.

```
            PROGRAM AIRSMO(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
      C
      C     THIS PROGRAM PRESENTS A TECHNIQUE FOR SMOOTHING AIRFOIL
      C     COORDINATES USING LEAST SQUARES POLYNOMIAL AND CUBIC SPLINE
      C     METHODS
      C
      C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
      C
            DIMENSION TITLE(8), XINT(100), X(200), Y(200), W(200), YSMO(200),
           1YPS(200), YPPS(202), THETA(202)
      C
            COMMON /HLM/ DUMMX(2000)
      C
            COMMON /SMY/ DUMMY(2130)
      C
            COMMON /BLK1/ PI,PI2,RAD,CONS
      C
            COMMON /INOUT/ JREAD,JWRITE,IPRINT
      C
      C     SINH(X)=0.5*(EXP(X)-EXP(-X))
      C
      C     INITIALIZE PROGRAM CONSTANTS
      C
            PI=ACOS(-1.)
            PI2=PI/2.
            RAD=180./PI
            CONS=1./(1.+ATAN(SINH(PI2)))
            JREAD=5
            JWRITE=6
            IPRINT=1
            EPS=1.E-6
            DF=1.E-4
            REWIND 1
      C
      C     INITIALIZE PLOTTING DEVICE
      C
            CALL PSEUDO
            CALL LEROY
      C
      C     READ INPUT DATA
```

CARD NO.

```
      C                                                                 AS  41
      1  CALL INPUT (TITLE,ITER,IPLOT,IPUNCH,IOP,ICAMTK,INTR,YLTE,YNOSE,YUT AS  42
        1E,NINT,XINT,CNEW,NP,X,Y,W,THETA,YPS,YPPS,NOSE,CHORD,IERR)      AS  43
         IF (IERR-1) 2,1,5                                              AS  44
      C                                                                 AS  45
      C  SMOOTH AIRFOIL COORDINATES                                     AS  46
      C                                                                 AS  47
      2  CALL SMOXY (THETA,X,Y,W,YSMO,YPS,YPPS,NP,NOSE,YLTE,YNOSE,YUTE,EPS, AS  48
        1DF,ITER,TITLE,IOP,IERR)                                        AS  49
         IF (IERR.NE.0) GO TO 1                                         AS  50
      C                                                                 AS  51
      C  PUNCH OUTPUT DATA                                              AS  52
      C                                                                 AS  53
         IF (IPUNCH.GE.1.AND.IPUNCH.LE.4) CALL PCARD (IPUNCH,X,Y,W,THETA,YS AS  54
        1MO,YPS,YPPS,NOSE,NP,CHORD,TITLE)                               AS  55
      C                                                                 AS  56
      C  PLOT SMOOTHED AND UNSMOOTHED Y/C, SMOOTHED YPS, AND SMOOTHED   AS  57
      C  YPPS VERSUS THETA. ALSO PLOT SMOOTHED AND UNSMOOTHED Y/C VERSUS AS  58
      C  X/C                                                            AS  59
      C                                                                 AS  60
         IF (IPLOT.EQ.0.OR.IPLOT.EQ.4) GO TO 4                          AS  61
         IF (IPLOT.EQ.3) GO TO 3                                        AS  62
      C                                                                 AS  63
         CALL PLOTAF (THETA,Y,YSMO,YPS,YPPS,NP,TITLE,IPLOT)             AS  64
      C                                                                 AS  65
         IF (IPLOT.EQ.5.OR.IPLOT.EQ.1) GO TO 4                          AS  66
         IF (IPLOT.EQ.6.OR.IPLOT.EQ.7) GO TO 3                          AS  67
         IF (IPLOT.EQ.10) GO TO 3                                       AS  68
         GO TO 4                                                        AS  69
      C                                                                 AS  70
      C  PLOT SMOOTHED CURVATURE VERSUS THETA                           AS  71
      C                                                                 AS  72
      3  CALL PLOTCK (THETA,YSMO,YPS,YPPS,NP,TITLE)                     AS  73
      C                                                                 AS  74
      4  KPLOT=0                                                        AS  75
         IF (IPLOT.EQ.4.OR.IPLOT.GE.8) KPLOT=1                          AS  76
      C                                                                 AS  77
      C  COMPUTE THICKNESS AND CAMBER DISTRIBUTION                      AS  78
      C                                                                 AS  79
         IF (ICAMTK.EQ.1) CALL CAMTK (THETA,YSMO,YPPS,NOSE,NP,EPS,KPLOT,IPU AS  80
```

CARD NO.

```
           1NCH,TITLE)
      C
      C     INTERPOLATE NEW COORDINATES
      C
            IF (INTR.GT.0) CALL INTP (THETA,X,YSMO,YPPS,NP,NOSE,CHORD,TITLE,NI
           1NT,XINT,CNEW,INTR,IPUNCH)
      C
      C     RETURN AND READ NEXT CASE
      C
            GO TO 1
      C
      C     FINALIZE PLOTTING DEVICE
      C
      5     CALL CALPLT (0.,0.,999)
            WRITE (JWRITE,6)
            END FILE 1
            REWIND 1
            STOP
      C
      6     FORMAT (1H1////48X,38H-- THE LAST CASE HAS BEEN PROCESSED --)
            END
```

CARD NO.

```
      SUBROUTINE INTER (XINT,YINT,N,X,Y,JSTART,JEND,ICD)
C
C     INTERPOLATION ROUTINE
C
C        ROUTINE SOURCE -- NORTH AMERICAN ROCKWELL L. A. DIVISION  1973
C
C        ICD=0  WEIGHTING METHOD USED
C        ICD=1  LINEAR INTERPOLATION
C
      DIMENSION X(N), Y(N)
C
C     CHECK TO SEE IF XINT IS OUTSIDE BOUNDS OF X-ARRAY
C
      JEND=JSTART
      IF (JSTART.EQ.N) GO TO 12
C        CHECK TO SEE IF X ARRAY IS INCREASING OR DECREASING
      SGN=1.
      IF (X(N).LT.X(JSTART)) SGN=-1.
      D1=SGN*(XINT-X(N))
      IF (D1.GE.0.0) GO TO 12
      D1=SGN*(XINT-X(JSTART))
      IF (D1.LE.0.0) GO TO 13
      IF (ICD.EQ.1) GO TO 14
C     WEIGHTING METHOD REQUIRES AT LEAST 4 VALUES IN X AND Y ARRAYS
      IF (N.LT.4) GO TO 14
C
C     WEIGHTING METHOD
C
C        DETERMINE X-ARRAY INDICES FOR TWO POINTS FORWARD (J,L) AND TWO
C        POINTS AFT (K,M) OF XINT
      DO 1 L=JSTART,N
      J=L
      D1=SGN*(X(J)-XINT)
      IF (D1) 1,2,3
    1 JEND=J
    2 YINT=Y(J)
      RETURN
    3 IF (J.LE.2) GO TO 5
      IF (J.EQ.N) GO TO 4
      JJ=3
```

CARD NO.

```
                GO TO 6
      4         JJ=2
                J=N-1
                GO TO 6
      5         JJ=1
                J=3
      6         K=J-1
                M=J-2
                L=J+1
      C            INTERPOLATE A YINT VALUE (YSL) BY FITTING A STRAIGHT LINE
      C            BETWEEN K AND J
                D1=XINT-X(M)
                D2=XINT-X(K)
                D3=XINT-X(J)
                D=(XINT-X(K))/(X(J)-X(K))
                YSL=D*Y(J)+(1.0-D)*Y(K)
      C            INTERPOLATE A YINT VALUE (YP1) BY FITTING A QUADRATIC BETWEEN
      C            M, K, AND J
                C1=D3*D2/((X(M)-X(K))*(X(M)-X(J)))
                C2=D1*D3/((X(K)-X(M))*(X(K)-X(J)))
                C3=D2*D1/((X(J)-X(M))*(X(J)-X(K)))
                YP1=C1*Y(M)+C2*Y(K)+C3*Y(J)
      C            INTERPOLATE A YINT VALUE (YP2) BY FITTING A QUADRATIC BETWEEN
      C            K, J, AND L
                D4=XINT-X(L)
                C1=D4*D3/((X(K)-X(J))*(X(K)-X(L)))
                C2=D2*D4/((X(J)-X(K))*(X(J)-X(L)))
                C3=D3*D2/((X(L)-X(K))*(X(L)-X(J)))
                YP2=C1*Y(K)+C2*Y(J)+C3*Y(L)
      C
                IF (JJ-2) 7,8,9
      7         YP2=YP1
                D=(XINT-X(1))/(X(2)-X(1))
                YSL=D*Y(2)+(1.0-D)*Y(1)
                GO TO 9
      8         YP1=YP2
                D=(XINT-X(N-1))/(X(N)-X(N-1))
                YSL=D*Y(N)+(1.0-D)*Y(N-1)
      C            COMPUTE DEVIATION BETWEEN LINEAR AND QUADRATIC YINT VALUES
      9         DEV1=ABS(YP1-YSL)
```

CARD NO.

```
          DEV2=ABS(YP2-YSL)
          IF (DEV1+DEV2) 10,10,11
   10     YINT=YSL
          RETURN
C            COMPUTE WEIGHTING FACTORS
   11     WT2=(DEV1*D)/(DEV1*D+(1.0-D)*DEV2)
          WT1=1.0-WT2
C            COMPUTE FINAL YINT
          YINT=WT2*YP2+WT1*YP1
          RETURN
   12     YINT=Y(N)
          JEND=N
          RETURN
   13     YINT=Y(JSTART)
          RETURN
C
C         LINEAR INTERPOLATION METHOD
C
   14     DO 15 L=JSTART,N
          J=L
          D1=SGN*(X(J)-XINT)
          IF (D1) 15,2,16
   15     JEND=J
   16     YINT=Y(J-1)+(Y(J)-Y(J-1))*(XINT-X(J-1))/(X(J)-X(J-1))
          RETURN
          END
```

CARD NO.

```
      SUBROUTINE INPUT (TITLE,ITER,IPLOT,IPUNCH,IOP,ICAMTK,INTR,YLTE,YNO IU   1
     1SE,YUTE,NINT,XINT,CNEW,NP,X,Y,W,THETA,YPS,YPPS,NOSE,CHORD,IERR)   IU   2
C                                                                       IU   3
C     ROUTINE TO READ INPUT DATA FOR AIRFOIL SMOOTHING PROGRAM          IU   4
C                                                                       IU   5
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB      1982          IU   6
C                                                                       IU   7
C********************************************************************** IU   8
C*                                                                    * IU   9
C*          DESCRIPTION OF INPUT CARDS FOR SMOOTHING PROGRAM          * IU  10
C*                                                                    * IU  11
C* CARD NUMBER                      DESCRIPTION                       * IU  12
C*                                                                    * IU  13
C*....................................................................* IU  14
C*   1    FORMAT(8A10)                                                * IU  15
C*             TITLE CARD                                             * IU  16
C*....................................................................* IU  17
C*   2    FORMAT(8F10.0)                                              * IU  18
C*             ITER - MAXIMUM NUMBER OF SMOOTHING ITERATIONS          * IU  19
C*             IPLOT - PLOTTING OPTION                                * IU  20
C*                     0 - NO PLOTS                                   * IU  21
C*                     1 - PLOT SMOOTHED AND UNSMOOTHED Y/C, SMOOTHED * IU  22
C*                         YPS, AND SMOOTHED YPPS VS THETA            * IU  23
C*                     2 - PLOT SMOOTHED AND UNSMOOTHED Y/C VS X/C    * IU  24
C*                     3 - PLOT SMOOTHED CURVATURE VS THETA           * IU  25
C*                     4 - PLOT CAMBER AND THICKNESS DISTRIBUTION     * IU  26
C*                     5 - PLOT OPTIONS 1 AND 2                       * IU  27
C*                     6 - PLOT OPTIONS 1 AND 3                       * IU  28
C*                     7 - PLOT OPTIONS 1, 2, AND 3                   * IU  29
C*                     8 - PLOT OPTIONS 1 AND 4                       * IU  30
C*                     9 - PLOT OPTIONS 1, 2, AND 4                   * IU  31
C*                    10 - PLOT OPTIONS 1, 2, 3, AND 4                * IU  32
C*             IPUNCH - PUNCH OUTPUT OPTION                           * IU  33
C*                      0 - NO PUNCHED OUTPUT                         * IU  34
C*                      1 - SMOOTHED (X,Y,W) PUNCHED                  * IU  35
C*                      2 - SMOOTHED (THETA,Y/C,W) PUNCHED            * IU  36
C*                      3 - SMOOTHED (THETA,YPS,W) PUNCHED (YLTE,     * IU  37
C*                          YNOSE, AND YUTE ALSO PUNCHED)             * IU  38
C*                      4 - SMOOTHED (THETA,YPPS,W) PUNCHED (YLTE,    * IU  39
C*                          YNOSE, AND YUTE ALSO PUNCHED)             * IU  40
```

CARD NO.

```
C*                     5 - THICKNESS AND CAMBER DISTRIBUTION (X/C,       * IU  41
C*                         Y/C, T/C/2, AND SLOPE) PUNCHED                * IU  42
C*                     6 - INTERPOLATED COORDINATES PUNCHED              * IU  43
C*             IOP - INPUT DATA OPTION                                   * IU  44
C*                   0 - (X,Y,W) INPUT                                   * IU  45
C*                   1 - (THETA,Y/C,W) INPUT                             * IU  46
C*                   2 - (THETA,YPS,W) INPUT                             * IU  47
C*                   3 - (THETA,YPPS,W) INPUT                            * IU  48
C*             ICAMTK - THICKNESS AND CAMBER DISTRIBUTION OPTION         * IU  49
C*                     0 - DO NOT COMPUTE THICKNESS AND CAMBER           * IU  50
C*                     1 - COMPUTE THICKNESS AND CAMBER                  * IU  51
C*             IBAD - BAD COORDINATE CHECK OPTION                        * IU  52
C*                    0 - DO NOT CHECK FOR BAD COORDINATES               * IU  53
C*                    1 - CHECK FOR BAD COORDINATES                      * IU  54
C*             ITRN - INPUT COORDINATE TRANSLATION AND ROTATION OPTION   * IU  55
C*                    0 - DO NOT TRANSLATE AND ROTATE                    * IU  56
C*                    1 - TRANSLATE AND ROTATE SO THAT X-AXIS            * IU  57
C*                        CORRESPONDS TO THE LONGEST CHORDLINE           * IU  58
C*             INTR - COORDINATE INTERPOLATION OPTION                    * IU  59
C*                    0 - NO INTERPOLATION DESIRED                       * IU  60
C*                    1 - INTERPOLATE NEW COORDINATES USING STANDARD 57  * IU  61
C*                        X/C COORDINATES DEFINED IN SUBROUTINE INTP     * IU  62
C*                    2 - INTERPOLATE NEW COORDINATES AT INPUT X/C       * IU  63
C*                        VALUES  (0.0 .GE. X/C .LE. 1.0)                * IU  64
C*.......................................................................* IU  65
C*   3   FORMAT(10.0)                                                    * IU  66
C*             NU - NUMBER OF UPPER SURFACE INPUT COORDINATES            * IU  67
C*.......................................................................* IU  68
C*   4   FORMAT(3F10.0)                                                  * IU  69
C*             XU,YU,WU - UPPER SURFACE INPUT COORDINATES AND WEIGHTING  * IU  70
C*                        (NU CARDS ARE INPUT)                           * IU  71
C*                     IF IOP=0, XU=X AND YU=Y COORDINATES               * IU  72
C*                     IF IOP=1, XU=THETA AND YU=Y/C                     * IU  73
C*                     IF IOP=2, XU=THETA AND YU=YPS                     * IU  74
C*                     IF IOP=3, XU=THETA AND YU=YPPS                    * IU  75
C*                     FOR ALL IOP, WU=WEIGHTING FACTOR                  * IU  76
C*.......................................................................* IU  77
C*   5   FORMAT(10.0)                                                    * IU  78
C*             NL - NUMBER OF LOWER SURFACE INPUT COORDINATES            * IU  79
C*.......................................................................* IU  80
```

CARD NO.

```
C*    6   FORMAT(3F10.0)                                                    *
C*              XL,YL,WL - LOWER SURFACE INPUT COORDINATES AND WEIGHTING  *
C*                         (NL CARDS ARE INPUT)                             *
C*                     IF IOP=0, XL=X AND YL=Y COORDINATES                  *
C*                     IF IOP=1, XL=THETA AND YL=Y/C                        *
C*                     IF IOP=2, XL=THETA AND YL=YPS                        *
C*                     IF IOP=3, XL=THETA AND YL=YPPS                       *
C*                     FOR ALL IOP, WL=WEIGHTING FACTOR                     *
C*.......................................................................*
C*    7   FORMAT(3F10.0)       SKIP IF IOP=0 OR 1                           *
C*              YLTE,YNOSE,YUTE - LOWER SURFACE TRAILING-EDGE, NOSE,        *
C*                               AND UPPER SURFACE TRAILING-EDGE            *
C*                               Y/C COORDINATES                            *
C*.......................................................................*
C*    8   FORMAT(F10.0)        SKIP IF INTR=0 OR 1                          *
C*              NINT - NUMBER OF INTERPOLATION X/C COORDINATES              *
C*.......................................................................*
C*    9   FORMAT(8F10.0)       SKIP IF INTR=0 OR 1                          *
C*              XINT - INTERPOLATION X/C COORDINATES (NINT VALUES INPUT)   *
C*.......................................................................*
C*   10   FORMAT(F10.0)        SKIP IF INTR=0                               *
C*              CNEW - DESIRED CHORD LENGTH OF INTERPOLATED COORDINATES    *
C*.......................................................................*
C*                                                                          *
C*        RESTRICTIONS:                                                     *
C*              ITER NOT GREATER THAN 300                                   *
C*              NU OR NL NOT GREATER THAN 100                               *
C*              NINT NOT GREATER THAN 100                                   *
C*                                                                          *
C**************************************************************************
C
      DIMENSION VAR(8), TITLE(8), XINT(1), X(1), Y(1), W(1), THETA(1), Y
     1PS(1), YPPS(1)
C
      COMMON /SMY/ XU(100),YU(100),WU(100),XL(100),YL(100),WL(100)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
```

CARD NO.

```
      C     SINH(X)=(EXP(X)-EXP(-X))/2.
      C
      C     INITIALIZE ROUTINE CONSTANTS
      C
            ITRMAX=300
            NMAX=100
            TOLP=1.E-2
            IERR=0
      C
      C     READ AND PRINT INPUT DATA
      C
      C        READ AND WRITE TITLE
            READ (JREAD,27) TITLE
            IF (EOF(JREAD)) 25,1
      1     WRITE (JWRITE,28) TITLE
      C        READ AND WRITE OPTIONS
            READ (JREAD,29) VAR
            ITER=IFIX(VAR(1))
            IPLOT=IFIX(VAR(2))
            IPUNCH=IFIX(VAR(3))
            IOP=IFIX(VAR(4))
            ICAMTK=IFIX(VAR(5))
            IBAD=IFIX(VAR(6))
            ITRN=IFIX(VAR(7))
            INTR=IFIX(VAR(8))
      C              CHECK LIMITS OF OPTIONS
            IF (ITER.GT.ITRMAX) ITER=ITRMAX
            IF (IPLOT.GT.10) IPLOT=0
            IF (IPUNCH.GT.6) IPUNCH=0
            IF (IOP.GT.3) GO TO 23
            IF (ICAMTK.NE.0) ICAMTK=1
            IF (IBAD.NE.0) IBAD=1
            IF (ITRN.NE.0) ITRN=1
            IF (INTP.GT.2) INTR=0
            WRITE (JWRITE,30) ITER,IPLOT,IPUNCH,IOP,ICAMTK,IBAD,ITRN,INTR
      C        READ AND WRITE NUMBER OF UPPER SURFACE INPUT POINTS
            READ (JREAD,29) VAR(1)
            NU=IFIX(VAR(1))
            IF (NU.GT.NMAX) GO TO 22
            WRITE (JWRITE,31) NU
```

```
      C        READ AND WRITE UPPER SURFACE INPUT POINTS AND WEIGHTING
               READ (JREAD,32) (XU(I),YU(I),WU(I),I=1,NU)
               DO 2 I=1,NU
               IF (WU(I).LT.1.0) WU(I)=1.0
      2        CONTINUE
               IF (IOP.EQ.0) WRITE (JWRITE,33) (XU(I),I=1,NU)
               IF (IOP.NE.0) WRITE (JWRITE,34) (XU(I),I=1,NU)
               IF (IOP.LT.2) WRITE (JWRITE,35) (YU(I),I=1,NU)
               IF (IOP.EQ.2) WRITE (JWRITE,36) (YU(I),I=1,NU)
               IF (IOP.EQ.3) WRITE (JWRITE,37) (YU(I),I=1,NU)
               WRITE (JWRITE,38) (WU(I),I=1,NU)
      C        READ AND WRITE NUMBER OF LOWER SURFACE INPUT POINTS
               READ (JREAD,29) VAR(1)
               NL=IFIX(VAR(1))
               IF (NL.GT.NMAX) GO TO 22
               WRITE (JWRITE,39) NL
      C        READ AND WRITE LOWER SURFACE INPUT POINTS AND WEIGHTING
               READ (JREAD,32) (XL(I),YL(I),WL(I),I=1,NL)
               DO 3 I=1,NL
               IF (WL(I).LT.1.0) WL(I)=1.0
      3        CONTINUE
               IF (IOP.EQ.0) WRITE (JWRITE,40) (XL(I),I=1,NL)
               IF (IOP.NE.0) WRITE (JWRITE,41) (XL(I),I=1,NL)
               IF (IOP.LT.2) WRITE (JWRITE,42) (YL(I),I=1,NL)
               IF (IOP.EQ.2) WRITE (JWRITE,43) (YL(I),I=1,NL)
               IF (IOP.EQ.3) WRITE (JWRITE,44) (YL(I),I=1,NL)
               WRITE (JWRITE,45) (WL(I),I=1,NL)
      C        READ AND WRITE TRAILING-EDGE COORDINATES
               IF (IOP.LE.1) GO TO 4
               READ (JREAD,29) YLTE,YNOSE,YUTE
               WRITE (JWRITE,46) YLTE,YNOSE,YUTE
      C        READ AND WRITE NUMBER OF INTERPOLATION COORDINATES
      4        IF (INTR.EQ.0) GO TO 6
               IF (INTR.NE.2) GO TO 5
               READ (JREAD,29) VAR(1)
               NINT=IFIX(VAR(1))
               IF (NINT.GT.NMAX) GO TO 24
               WRITE (JWRITE,47) NINT
      C        READ AND WRITE INTERPOLATION COORDINATES
               READ (JREAD,29) (XINT(I),I=1,NINT)
```

CARD NO.

```
      WRITE (JWRITE,48) (XINT(I),I=1,NINT)
C        READ AND WRITE NEW CHORD OF INTERPOLATED COORDINATES
5     READ (JREAD,29) CNEW
      WRITE (JWRITE,49) CNEW
C
C     CHECK UPPER SURFACE COORDINATES FOR BAD POINTS
C
6     IF (IOP.NE.0) GO TO 7
      IF (IBAD.EQ.1) CALL BADPT (XU,YU,NU,TOLR,1,IERR)
      IF (IERR.NE.0) GO TO 26
C
C     CHECK LOWER SURFACE COORDINATES FOR BAD POINTS
C
      IF (IBAD.EQ.1) CALL BADPT (XL,YL,NL,TOLR,2,IERR)
      IF (IERR.NE.0) GO TO 26
C
C     TRANSLATE AND ROTATE THE INPUT COORDINATES SO THAT THE X-AXIS
C     CORRESPONDS TO THE LONGEST CHORDLINE OF THE AIRFOIL
C
      IF (ITRN.EQ.1) CALL TRNSRT (XU,YU,WU,NU,XL,YL,WL,NL,TITLE)
C
C     LOAD X, Y, THETA, YPS, AND YPPS ARRAYS
C
7     IF (IOP) 8,8,15
C        IF IOP=0, COMPUTE THETA FROM INPUT X
C              COMPUTE THETA FOR LOWER SURFACE
8     CHORD=XL(NL)-XL(1)
      DELTA=XU(NU)-XU(1)
      IF (DELTA.GT.CHORD) CHORD=DELTA
      NP=0
      DO 11 I=1,NL
      NP=NP+1
      J=NL+1-I
      W(NP)=WL(J)
      DELTA=(XL(J)-XL(1))/CHORD
      IF (DELTA.LE.CONS) GO TO 9
      DELTA=TAN(DELTA/CONS-1.)
      THETA(NP)=-PI2-ALOG(DELTA+SQRT(DELTA*DELTA+1.))
      GO TO 10
9     THETA(NP)=-ACOS(1.-DELTA/CONS)
```

CARD NO.

```
 10     X(NP)=XL(J)/CHORD
 11     Y(NP)=YL(J)/CHORD
        NOSE=NP
C               COMPUTE THETA FOR UPPER SURFACE
        J=1
        IF (XL(1).EQ.XU(1).AND.YL(1).EQ.YU(1)) J=2
        DO 14 I=J,NU
        NP=NP+1
        W(NP)=WU(I)
        DELTA=(XU(I)-XU(1))/CHORD
        IF (DELTA.LE.CONS) GO TO 12
        DELTA=TAN(DELTA/CONS-1.)
        THETA(NP)=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))
        GO TO 13
 12     THETA(NP)=ACOS(1.-DELTA/CONS)
 13     X(NP)=XU(I)/CHORD
 14     Y(NP)=YU(I)/CHORD
        GO TO 20
C          IF IOP=1, 2, OR 3, COMPUTE X/C FROM INPUT THETA
C               COMPUTE X/C FOR LOWER SURFACE
 15     CHORD=1.0
        NP=0
        DO 17 I=1,NL
        NP=NP+1
        J=NL+1-I
        W(NP)=WL(J)
        IF (IOP.EQ.1) Y(NP)=YL(J)
        IF (IOP.EQ.2) YPS(NP)=YL(J)
        IF (IOP.EQ.3) YPPS(NP)=YL(J)
        THETA(NP)=XL(J)/RAD
        DELTA=ABS(THETA(NP))
        IF (DELTA.GT.PI2) GO TO 16
        XL(J)=CONS*(1.-COS(DELTA))
        GO TO 17
 16     XL(J)=CONS*(ATAN(SINH(DELTA-PI2))+1.)
 17     X(NP)=XL(J)
        NOSE=NP
C               COMPUTE X/C FOR UPPER SURFACE
        XU(1)=XL(1)
        DO 19 I=2,NU
```

```
      NP=NP+1
      W(NP)=WU(I)
      IF (IOP.EQ.1) Y(NP)=YU(I)
      IF (IOP.EQ.2) YPS(NP)=YU(I)
      IF (IOP.EQ.3) YPPS(NP)=YU(I)
      THETA(NP)=XU(I)/RAD
      DELTA=ABS(THETA(NP))
      IF (DELTA.GT.PI2) GO TO 18
      XU(I)=CONS*(1.-COS(DELTA))
      GO TO 19
   18 XU(I)=CONS*(ATAN(SINH(DELTA-PI2))+1.)
   19 X(NP)=XU(I)
C
C     PRINT SUMMARY OF INPUT DATA
C
   20 WRITE (JWRITE,50) TITLE
      DO 21 I=1,NP
      DELTA=THETA(I)*RAD
      IF (IOP.LE.1) WRITE (JWRITE,51) I,X(I),Y(I),DELTA,W(I)
      IF (IOP.EQ.2) WRITE (JWRITE,52) I,X(I),DELTA,YPS(I),W(I)
      IF (IOP.EQ.3) WRITE (JWRITE,53) I,X(I),DELTA,YPPS(I),W(I)
   21 CONTINUE
      WRITE (JWRITE,54) CHORD
      GO TO 26
C
C     PRINT ERROR MESSAGES
C
   22 NN=IFIX(VAR(1))
      WRITE (JWRITE,55) NN
      GO TO 25
   23 WRITE (JWRITE,56) IOP
      GO TO 25
   24 WRITE (JWRITE,57) NINT
C
C     NO ADDITIONAL INPUT DATA
C
   25 IERR=2
C
C     RETURN TO CALLING PROGRAM
C
```

CARD NO.

```
26    RETURN
C
27    FORMAT (8A10)
28    FORMAT (1H1,57X,14H--INPUT DATA--//5X,7HTITLE--,2X,8A10)
29    FORMAT (8F10.5)
30    FORMAT (/5X,6HITER =,I4,3X,7HIPLOT =,I3,3X,8HIPUNCH =,I3,3X,5HIOP
     1=,I3,3X,8HICAMTK =,I3,3X,6HIBAD =,I3,3X,6HITRN =,I3,3X,6HINTR =,I3
     2)
31    FORMAT (/5X,4HNU =,I4)
32    FORMAT (3F10.5)
33    FORMAT (/5X,3HXU=,8E15.6/(8X,8E15.6))
34    FORMAT (/5X,3HTU=,8E15.6/(8X,8E15.6))
35    FORMAT (/5X,3HYU=,8E15.6/(8X,8E15.6))
36    FORMAT (/4X,4HYPU=,8E15.6/(8X,8E15.6))
37    FORMAT (/3X,5HYPPU=,8E15.6/(8X,8E15.6))
38    FORMAT (/5X,3HWU=,8E15.6/(8X,8E15.6))
39    FORMAT (/5X,4HNL =,I4)
40    FORMAT (/5X,3HXL=,8E15.6/(8X,8E15.6))
41    FORMAT (/5X,3HTL=,8E15.6/(8X,8E15.6))
42    FORMAT (/5X,3HYL=,8E15.6/(8X,8E15.6))
43    FORMAT (/4X,4HYPL=,8E15.6/(8X,8E15.6))
44    FORMAT (/3X,5HYPPL=,8E15.6/(8X,8E15.6))
45    FORMAT (/5X,3HWL=,8E15.6/(8X,8E15.6))
46    FORMAT (/3X,6HYLTE =,E15.6,5X,7HYNOSE =,E15.6,5X,6HYUTE =,E15.6)
47    FORMAT (/3X,6HNINT =,I4)
48    FORMAT (/3X,5HXINT=,8E15.6/(8X,8E15.6))
49    FORMAT (/3X,6HCNEW =,F10.3)
50    FORMAT (1H1,29X,25H--SUMMARY OF INPUT DATA--//5X,9HTITLE--  ,8A10/
     1/9X,1HI,10X,3HX/C,12X,3HY/C,12X,5HTHETA,10X,3HYPS,12X,4HYPPS,14X,1
     2HW)
51    FORMAT (I10,2F15.6,F15.2,30X,F15.2)
52    FORMAT (I10,F15.6,15X,F15.2,F15.6,15X,F15.2)
53    FORMAT (I10,F15.6,15X,F15.2,15X,F15.6,F15.2)
54    FORMAT (/5X,7HCHORD =,F15.6)
55    FORMAT (//5X,28HINPUT CARD ERROR  NU OR NL =,I4)
56    FORMAT (//5X,23HINPUT CARD ERROR  IOP =,I4)
57    FORMAT (//5X,24HINPUT CARD ERROR  NINT =,I5)
      END
```

CARD NO.

```
      SUBROUTINE TRNSRT (XU,YU,WU,NU,XL,YL,WL,NL,TITLE)
C
C     ROUTINE TO TRANSLATE AND ROTATE THE INPUT AIRFOIL COORDINATES SO
C     THAT THE X-AXIS CORRESPONDS TO THE LONGEST CHORDLINE
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION XU(1), YU(1), WU(1), XL(1), YL(1), WL(1), TITLE(8)
C
      COMMON /HLM/ X(200),Y(200),W(200)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
C     PRINT INPUT COORDINATES
C
      WRITE (JWRITE,13) TITLE
      J=NU
      IF (NL.GT.NU) J=NL
      DO 1 I=1,J
      IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,14) I,XU(I),YU(I),XL(I),YL(
     1I)
      IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,14) I,XU(I),YU(I)
      IF (I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,15) I,XL(I),YL(I)
    1 CONTINUE
C
C     COMPUTE LONGEST CHORDLINE
C
C        LOAD LOWER SURFACE COORDINATES
      N=0
      DO 2 I=1,NL
      J=NL+1-I
      N=N+1
      W(N)=WL(J)
      X(N)=XL(J)
    2 Y(N)=YL(J)
      J=1
      IF (XL(1).EQ.XU(1).AND.YL(1).EQ.YU(1)) J=2
C        LOAD UPPER SURFACE COORDINATES
```

CARD NO.

```
            DO 3 I=J,NU
            N=N+1
            W(N)=WU(I)
            X(N)=XU(I)
      3     Y(N)=YU(I)
      C        COMPUTE MIDPOINT OF TRAILING-EDGE BASE
            XTE=0.5*(X(1)+X(N))
            YTE=0.5*(Y(1)+Y(N))
      C        FIND MOST FORWARD LEADING-EDGE POINT AND LONGEST CHORD
            CHORD=0.0
            DO 5 I=1,N
            DIST=SQRT((X(I)-XTE)**2+(Y(I)-YTE)**2)
            IF (DIST-CHORD) 5,5,4
      4     CHORD=DIST
            NOSE=I
            XNOSE=X(I)
            YNOSE=Y(I)
      5     CONTINUE
      C
      C     TRANSLATE AND ROTATE AIRFOIL
      C
            IF (CHORD.LE.0.0) GO TO 6
            COSA=(XTE-XNOSE)/CHORD
            SINA=(YTE-YNOSE)/CHORD
            ANGLE=ATAN(SINA/COSA)*RAD
            GO TO 7
      6     COSA=0.0
            SINA=0.0
            ANGLE=0.0
      7     DO 8 I=1,N
            DIST=X(I)
            X(I)=(DIST-XNOSE)*COSA+(Y(I)-YNOSE)*SINA
      8     Y(I)=(Y(I)-YNOSE)*COSA-(DIST-XNOSE)*SINA
      C
      C     REDEFINE LOWER AND UPPER SURFACE COORDINATES
      C
            DO 9 I=1,NOSE
            J=NOSE+1-I
            WL(I)=W(J)
            XL(I)=X(J)
```

CARD NO.

```
      9      YL(I)=Y(J)
             NL=NOSE
             DO 10 I=NOSE,N
             J=I+1-NOSE
             WU(J)=W(I)
             XU(J)=X(I)
      10     YU(J)=Y(I)
             NU=J
      C
      C      PRINT NEW AIRFOIL COORDINATES
      C
             WRITE (JWRITE,16) TITLE
             J=NU
             IF (NL.GT.NU) J=NL
             DO 11 I=1,J
             IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,14) I,XU(I),YU(I),XL(I),YL(
            1I)
             IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,14) I,XU(I),YU(I)
             IF (I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,15) I,XL(I),YL(I)
      11     CONTINUE
             WRITE (JWRITE,12) XNOSE,YNOSE,ANGLE
             RETURN
      C
      12     FORMAT (/5X,7HXNOSE =,F15.6,5X,7HYNOSE =,F15.6,5X,7HANGLE =,F8.3)
      13     FORMAT (1H1,32X,21H--INPUT COORDINATES--//5X,7HTITLE--,2X,8A10//9X
            1,1HI,11X,2HXU,13X,2HYU,13X,2HXL,13X,2HYL)
      14     FORMAT (5X,I5,4F15.6)
      15     FORMAT (5X,I5,30X,2F15.6)
      16     FORMAT (1H1,21X,38H--TRANSLATED AND ROTATED COORDINATES--//5X,7HTI
            1TLE--,2X,8A10//9X,1HI,11X,2HXU,13X,2HYU,13X,2HXL,13X,2HYL)
             END
```

CARD NO.

```
      SUBROUTINE BADPT (X,Y,NP,TOLR,ISURF,IERR)
C
C     ROUTINE TO EDIT BAD POINTS FROM X AND Y INPUT COORDINATES
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB        1982
C
      DIMENSION X(1), Y(1), SURF(2)
C
      COMMON /HLM/ TI(100),YI(100),YN(100),THETA(100)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
      DATA SURF(1)/5HUPPER/,SURF(2)/5HLOWER/
C
C     IF TOLERANCE IS ZERO OR NEGATIVE RETURN
C
      IERR=0
      IF (TOLR.LE.0.0) RETURN
C
C     COMPUTE LOCAL CHORD
C
      CHORD=X(NP)-X(1)
C
C     INITIALIZE ITERATION PARAMETERS
C
      ICD=0
      IPTP=0
      N1=NP-1
      NMAX=0
      TOLC=TOLR*CHORD
C
C     COMPUTE THETA EQUIVALENT OF X
C
      DO 2 I=1,NP
      DELTA=(X(I)-X(1))/CHORD
      IF (DELTA.LE.CONS) GO TO 1
      DELTA=TAN(DELTA/CONS-1.)
      THETA(I)=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))
```

CARD NO.

```
            GO TO 2
      1     THETA(I)=ACOS(1.-DELTA/CONS)
      2     CONTINUE
      C
      C     LOOP TO SEARCH FOR BAD POINTS
      C
      3     NMAX=NMAX+1
            JSTART=1
      C        COMPUTE NEW Y VALUE BY INTERPOLATION
            DO 5 I=2,N1
            K=0
      C              LOAD TI AND YI ARRAY - OMIT THE I(TH) INPUT DATA POINT
            DO 4 J=1,NP
            IF (I.EQ.J) GO TO 4
            K=K+1
            TI(K)=THETA(J)
            YI(K)=Y(J)
      4     CONTINUE
      C              INTERPOLATE I(TH) DATA POINT
            CALL INTER (THETA(I),YN(I),K,TI,YI,JSTART,JEND,ICD)
            JSTART=JEND
      5     CONTINUE
      C        CHECK TOLERANCE OF INTERPOLATED POINTS
            IPT=0
            ERRMAX=0.
            DO 7 I=2,N1
            ERRMIN=0.
            ERR=ABS(YN(I)-Y(I))
            IF (ERR.GE.TOLC) ERRMIN=ERR
            IF (ERRMIN-ERRMAX) 7,7,6
      6     IPT=I
            ERRMAX=ERRMIN
      7     CONTINUE
            IF (IPT.EQ.0) RETURN
      C        PRINT COORDINATES OF BAD POINTS
            IF (NMAX.EQ.1) WRITE (JWRITE,9) SURF(ISURF),TOLC
            WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
      C        REPLACE BAD POINT WITH INTERPOLATED VALUE
            Y(IPT)=YN(IPT)
      C        CHECK TO SEE IF THIS BAD POINT IS ADJACENT TO THE PREVIOUS BAD
```

CARD NO.

```
      C          POINT -- IF IT IS, PRINT A WARNING MESSAGE AND TERMINATE
      C          PROGRAM EXECUTION
            IF ((IPTP.EQ.IPT-1).OR.(IPTP.EQ.IPT+1)) GO TO 8
            IF (IPTP.EQ.IPT) GO TO 8
            IPTP=IPT
            IF (NMAX.GE.NP) RETURN
      C
      C     RETURN TO START OF LOOP AND SEARCH FOR NEXT BAD POINT
      C
            GO TO 3
      C
      C     WARNING MESSAGE PRINT STATEMENT
      C
      8     WRITE (JWRITE,11)
            IERR=1
            RETURN
      C
      9     FORMAT (1H1//1X,44HWARNING -- BAD POINTS HAVE BEEN FOUND ON THE,1X
           1,A5,1X,37HSURFACE BASED ON AN EDIT TOLERANCE OF,F10.6/)
      10    FORMAT (1X,15HBAD POINT AT I=,I4,5X,4HX = ,F10.6,5X,4HY = ,F10.6,5
           1X,18HREPLACED WITH Y = ,F10.6/)
      11    FORMAT (1X,93HADJACENT BAD POINTS HAVE BEEN FOUND -- PLEASE CORREC
           1T YOUR INPUT DATA AND RESUBMIT THIS CASE.)
            END
```

CARD NO.

```
      SUBROUTINE SMOXY (THETA,X,Y,W,YSMO,YPS,YPPS,NP,NOSE,YLTE,YNOSE,YUT
     1E,EPS,DF,ITER,TITLE,IOP,IERR)
C
C     THIS SUBROUTINE PRESENTS A TECHNIQUE FOR SMOOTHING Y INPUT
C     COORDINATES USING LEAST SQUARES POLYNOMIAL AND CUBIC SPLINE
C     METHODS
C
C     IF IOP=0 OR 1, COMPUTE YPPU (UNSMOOTHED SECOND DERIVATIVES) FROM
C     LEAST SQUARES POLYNOMIAL FITTING OF Y VS THETA. THEN COMPUTE
C     YPPS (SMOOTHED SECOND DERIVATIVES) FROM LEAST SQUARES CUBIC
C     SPLINE FITTING OF YPPU VS THETA. FINALLY COMPUTE YSMO (SMOOTHED Y
C     COORDINATES) USING INVERSE CUBIC SPLINE METHOD.
C
C     IF IOP=2, COMPUTE SECOND DERIVATIVES FROM INPUT FIRST DERIVATIVES.
C     THEN COMPUTE UNSMOOTHED Y COORDINATES FROM SECOND DERIVATIVES AND
C     FOLLOW SAME PROCEDURES AS OUTLINED ABOVE FOR IOP 0 OR 1.
C
C     IF IOP=3, COMPUTE UNSMOOTHED Y COORDINATES FROM INPUT SECOND
C     DERIVATIVES. THEN FOLLOW SAME PROCEDURES AS OUTLINED ABOVE FOR
C     IOP 0 OR 1.
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
C        DIMENSION THETA, X, Y, W, YSMO, YPS, AND YPPS BY NP IN CALLING
C        PROGRAM
      DIMENSION TITLE(8), THETA(1), X(1), Y(1), W(1), YSMO(1), YPS(1), Y
     1PPS(1)
C
      COMMON /HLM/ WK(200,10)
C
      COMMON /SMY/ YPP(200),YUSMO(200),DUM(200),A(200,4),YN(200),YPPU(20
     10),SUMY(300),LTER(30)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
      DATA LMX/200/,WT/100./
C
C     SINH(X)=(EXP(X)-EXP(-X))/2.
```

```
      C     COSH(X)=(EXP(X)+EXP(-X))/2.
      C
            IERR=0
            IF (IOP.EQ.0.OR.IOP.EQ.1) GO TO 13
            IF (IOP.EQ.2) GO TO 1
            IF (IOP.EQ.3) GO TO 11
      C
      C        IF IOP=2, COMPUTE SECOND DERIVATIVES FROM INPUT FIRST
      C        DERIVATIVES. THEN COMPUTE INITIAL Y/C COORDINATES FROM SECOND
      C        DERIVATIVES.
      C
      C              COMPUTE SECOND DERIVATIVES USING CSDS
      1     DO 2 I=1,NP
      2     DUM(I)=1.0
            T1=0.0
            CALL CSDS (LMX,NP,THETA,YPS,DUM,T1,-1,A,WK,IERR)
            IF (IERR.NE.0) GO TO 71
            DO 4 I=1,NP
            IF (I.EQ.NP) GO TO 3
            YPPS(I)=A(I,2)
            GO TO 4
      3     DELTA=THETA(I)-THETA(I-1)
            YPPS(I)=(3.*A(I-1,4)*DELTA+2.*A(I-1,3))*DELTA+A(I-1,2)
      4     CONTINUE
      C              COMPUTE SECOND DERIVATIVES USING LSQSMO
            DELTA=1.
            CALL LSQSMO (THETA,YPS,W,DUM,YPP,YUSMO,NP,1,NP,NOSE,DELTA,EPS,IERR
           1)
            IF (IERR.NE.0) RETURN
      C              COMPUTE Y/C COORDINATES
            CALL YNEW (THETA,YPPS,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRITE,
           10)
            CALL YNEW (THETA,YPP,YUSMO,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRI
           1TE,0)
      C              COMPUTE NEW FIRST DERIVATIVES AND COMPARE WITH INPUT
      C              FIRST DERIVATIVES
            WRITE (JWRITE,73) TITLE
            SUM1=0.0
            SUM2=0.0
            DO 7 I=1,NP
```

```
      IF (I.EQ.1) GO TO 5
      DELTA=THETA(I)-THETA(I-1)
      YN(I)=YPPS(I-1)*DELTA/6.+YPPS(I)*DELTA/3.+(Y(I)-Y(I-1))/DELTA
      DUM(I)=YPP(I-1)*DELTA/6.+YPP(I)*DELTA/3.+(YUSMO(I)-YUSMO(I-1))/DEL
     1TA
      GO TO 6
    5 DELTA=THETA(2)-THETA(1)
      YN(1)=-YPPS(1)*DELTA/3.-YPPS(2)*DELTA/6.+(Y(2)-Y(1))/DELTA
      DUM(1)=-YPP(1)*DELTA/3.-YPP(2)*DELTA/6.+(YUSMO(2)-YUSMO(1))/DELTA
    6 T1=YPS(I)-YN(I)
      T2=YPS(I)-DUM(I)
      SUM1=SUM1+T1*T1
      SUM2=SUM2+T2*T2
    7 WRITE (JWRITE,74) I,YPS(I),YN(I),T1,DUM(I),T2
      WRITE (JWRITE,75) SUM1,SUM2
C            SELECT OUTPUT FROM EITHER CSDS OR LSQSMO
      DO 10 I=1,NP
      IF (SUM2.LT.SUM1) GO TO 8
      YPP(I)=YPPS(I)
      GO TO 9
    8 Y(I)=YUSMO(I)
      YN(I)=DUM(I)
    9 YSMO(I)=Y(I)
   10 YUSMO(I)=Y(I)
      IF (SUM2.GE.SUM1) WRITE (JWRITE,76)
      IF (SUM2.LT.SUM1) WRITE (JWRITE,77)
      IF (ITER.EQ.0) GO TO 48
      GO TO 13
C
C        IF IOP=3, COMPUTE INITIAL Y/C FROM INPUT SECOND DERIVATIVES
C        AND Y/C COORDINATES AT THE UPPER AND LOWER SURFACE TRAILING
C        EDGE AND NOSE
C
   11 CALL YNEW (THETA,YPPS,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRITE,
     10)
C            COMPUTE FIRST DERIVATIVES
      DO 12 I=1,NP
      YSMO(I)=Y(I)
      YUSMO(I)=Y(I)
      IF (I.EQ.1) GO TO 12
```

```
           DELTA=THETA(I)-THETA(I-1)
           YN(I)=YPPS(I-1)*DELTA/6.+YPPS(I)*DELTA/3.+(Y(I)-Y(I-1))/DELTA
      12   YPP(I)=YPPS(I)
           DELTA=THETA(2)-THETA(1)
           YN(1)=-YPPS(1)*DELTA/3.-YPPS(2)*DELTA/6.+(Y(2)-Y(1))/DELTA
           IF (ITER.EQ.0) GO TO 48
      C
      C       INITIALIZE ARRAYS
      C
      13   DO 14 I=1,NP
           YUSMO(I)=Y(I)
           IF (IOP.LT.2) YPP(I)=0.0
           YSMO(I)=THETA(I)*RAD
      14   DUM(I)=1.
           IF (ITER.GT.0) GO TO 17
      C
      C       IF IOP=0 OR 1 AND NO SMOOTHING DESIRED (I.E. ITER=0) , COMPUTE
      C       SECOND DERIVATIVE FROM CUBIC SPLINE SUBROUTINE
      C
           CALL CSDS (LMX,NP,THETA,Y,DUM,0.0,-1,A,WK,IERR)
           IF (IERR.NE.0) GO TO 71
      C       COMPUTE Y AND SECOND DERIVATIVE
           DO 16 I=1,NP
           IF (I.EQ.NP) GO TO 15
           YSMO(I)=A(I,1)
           YN(I)=A(I,2)
           YPP(I)=2.*A(I,3)
           GO TO 16
      15   DELTA=THETA(I)-THETA(I-1)
           YSMO(I)=((A(I-1,4)*DELTA+A(I-1,3))*DELTA+A(I-1,2))*DELTA+A(I-1,1)
           YN(I)=(3.*A(I-1,4)*DELTA+2.*A(I-1,3))*DELTA+A(I-1,2)
           YPP(I)=6.*A(I-1,4)*DELTA+2.*A(I-1,3)
      16   CONTINUE
           GO TO 48
      C
      C       FIND MAXIMUM INPUT Y VALUE AND ITS LOCATION FOR UPPER AND
      C       LOWER SURFACES
      C             LOWER SURFACE
      17   YMAX=0.0
           JMAXL=1
```

CARD NO.

```
              DO 19 I=1,NOSE                                                SO 161
              J=NOSE+1-I                                                    SO 162
              IF (ABS(Y(J)).GT.YMAX) GO TO 18                               SO 163
              GO TO 19                                                      SO 164
        18    YMAX=ABS(Y(J))                                                SO 165
              JMAXL=J                                                       SO 166
        19    CONTINUE                                                      SO 167
        C               UPPER SURFACE                                       SO 168
              YMAX=0.0                                                      SO 169
              JMAXU=1                                                       SO 170
              DO 21 I=NOSE,NP                                               SO 171
              IF (ABS(Y(I)).GT.YMAX) GO TO 20                               SO 172
              GO TO 21                                                      SO 173
        20    YMAX=ABS(Y(I))                                                SO 174
              JMAXU=I                                                       SO 175
        21    CONTINUE                                                      SO 176
        C                                                                   SO 177
        C        COMPUTE UNSMOOTHED SECOND DERIVATIVE USING LEAST           SO 178
        C        SQUARES POLYNOMIAL METHOD                                  SO 179
        C                                                                   SO 180
              J1=0                                                          SO 181
              ICON=0                                                        SO 182
              MTER=0                                                        SO 183
              J=ITER                                                        SO 184
              KTI=0                                                         SO 185
              IF (IPRINT.NE.0) WRITE (JWRITE,78) TITLE                      SO 186
              DO 23 I=1,30                                                  SO 187
              KTI=KTI+1                                                     SO 188
              LTER(I)=10                                                    SO 189
              J=J-10                                                        SO 190
              IF (J) 22,24,23                                               SO 191
        22    LTER(I)=10+J                                                  SO 192
              GO TO 24                                                      SO 193
        23    CONTINUE                                                      SO 194
        24    DO 39 LL=1,KTI                                                SO 195
              N1=LTER(LL)                                                   SO 196
              DO 34 I=1,N1                                                  SO 197
        C               CALL LEAST SQUARES POLYNOMIAL SMOOTHING ROUTINE     SO 198
              CALL LSQSMO (THETA,YUSMO,W,YN,DUM,YPPU,NP,JMAXL,JMAXU,NOSE,WT,EPS,  SO 199
             1IERR)                                                         SO 200
```

CARD NO.

```
      IF (IERR.NE.0) RETURN                                            SO 201
C           COMPUTE ERROR TERM                                         SO 202
      SUMY(I)=0.0                                                      SO 203
      DO 25 J=1,NP                                                     SO 204
   25 SUMY(I)=SUMY(I)+(YPPU(J)-YPP(J))**2                              SO 205
      J1=J1+1                                                          SO 206
      IF ((I.LE.3).AND.(LL.EQ.1)) GO TO 26                             SO 207
      IF (I.EQ.1) GO TO 26                                             SO 208
C           CHECK FOR OSCILLATIONS IN CONVERGENCE OF ERROR TERM        SO 209
      IF (SUMY(I)-SUMY(I-1)) 26,26,32                                  SO 210
C           LOAD ARRAYS FOR NEXT ITERATION                             SO 211
   26 DO 31 J=1,NP                                                     SO 212
      WK(J,I)=YPPU(J)                                                  SO 213
      IF (LL.EQ.1.AND.I.EQ.1) YPPS(J)=YPPU(J)                          SO 214
      YPP(J)=YPPU(J)                                                   SO 215
      CC=YUSMO(J)                                                      SO 216
      IF (J1-2) 29,28,27                                               SO 217
   27 AA=YN(J)-YUSMO(J)                                                SO 218
      BB=A(J,1)-A(J,2)                                                 SO 219
      T1=SIGN(1.,AA)                                                   SO 220
      T2=SIGN(1.,BB)                                                   SO 221
      IF (T1.EQ.T2.OR.AA.EQ.BB) GO TO 28                               SO 222
      YUSMO(J)=A(J,2)-BB*(YUSMO(J)-A(J,2))/(AA-BB)                     SO 223
      GO TO 30                                                         SO 224
   28 YUSMO(J)=0.5*(YUSMO(J)+YN(J))                                    SO 225
      GO TO 30                                                         SO 226
   29 YUSMO(J)=YN(J)                                                   SO 227
   30 A(J,1)=YN(J)                                                     SO 228
   31 A(J,2)=CC                                                        SO 229
      GO TO 33                                                         SO 230
   32 NTER=I-1                                                         SO 231
      ICON=2                                                           SO 232
      GO TO 36                                                         SO 233
   33 NTER=I                                                           SO 234
C           CHECK FOR CONVERGENCE BASED ON INPUT EPS                   SO 235
      IF (SUMY(I).LE.EPS) GO TO 35                                     SO 236
   34 CONTINUE                                                         SO 237
      GO TO 36                                                         SO 238
   35 ICON=1                                                           SO 239
C                                                                      SO 240
```

CARD NO.

```
C         PRINT SECOND DERIVATIVES GENERATED DURING SMOOTHING PROCESS
C
36    IF (IPRINT.NE.0) GO TO 38
      WRITE (JWRITE,80) TITLE
      DO 37 J=1,NP
37    WRITE (JWRITE,81) J,YSMO(J),(WK(J,I),I=1,MTER)
      WRITE (JWRITE,82) (SUMY(I),I=1,NTER)
38    IF (IPRINT.NE.0) WRITE (JWRITE,79) LL,(SUMY(I),I=1,NTER)
      MTER=MTER+NTER
      IF (ICON.NE.0) GO TO 40
39    CONTINUE
40    IF (ICON.EQ.0) WRITE (JWRITE,83) MTER
      IF (ICON.EQ.1) WRITE (JWRITE,84) MTER
      IF (ICON.EQ.2) WRITE (JWRITE,85) MTER
C
C         COMPUTE SMOOTHED SECOND DERIVATIVE USING LEAST SQUARES
C         CUBIC SPLINE
C
      DO 41 I=1,NP
41    DUM(I)=DF
C               CALL LEAST SQUARES CUBIC SPLINE ROUTINE
      CALL CSDS (LMX,NP,THETA,YPPU,DUM,FLOAT(NP),-1,A,WK,IERR)
      IF (IERR.NE.0) GO TO 71
C               COMPUTE SECOND DERIVATIVE
      SUM=0.0
      DO 44 I=1,NP
      IF (I.EQ.NP) GO TO 42
      YPP(I)=A(I,1)
      GO TO 43
42    DELTA=THETA(I)-THETA(I-1)
      YPP(I)=((A(I-1,4)*DELTA+A(I-1,3))*DELTA+A(I-1,2))*DELTA+A(I-1,1)
43    SUM=SUM+(YPPU(I)-YPP(I))**2
44    YPPU(I)=YPPS(I)
      WRITE (JWRITE,88) SUM
C
C         COMPUTE NEW Y COORDINATES FROM SMOOTHED SECOND DERIVATIVES
C
      CALL YNEW (THETA,YPP,YSMO,NOSE,NP,YUSMO(1),YUSMO(NOSE),YUSMO(NP),E
     1PS,DUM,WK,JWRITE,1)
C
```

CARD NO.

```
      C        CHECK NEW Y COORDINATES FOR SMOOTHNESS
      C
      C             CALL LEAST SQUARES POLYNOMIAL ROUTINE
            DO 45 I=1,NP
      45    A(I,1)=1.0
            CALL LSQSMO (THETA,YSMO,A,YN,DUM,YPPS,NP,1,NP,NOSE,WT,EPS,IERR)
            IF (IERR.NE.0) RETURN
      C             COMPUTE ERROR TERMS
            SUM1=0.0
            SUM2=0.0
            DO 46 I=1,NP
            A(I,1)=YSMO(I)-YN(I)
            A(I,2)=YPP(I)-YPPS(I)
            SUM1=SUM1+A(I,1)**2
      46    SUM2=SUM2+A(I,2)**2
      C
      C        COMPUTE FIRST DERIVATIVE FROM SMOOTHED SECOND DERIVATIVE
      C
            N1=NP-1
            DO 47 I=1,N1
            DELTA=THETA(I+1)-THETA(I)
      47    YN(I)=-YPP(I)*DELTA/3.-YPP(I+1)*DELTA/6.+(YSMO(I+1)-YSMO(I))/DELTA
            DELTA=THETA(NP)-THETA(N1)
            YN(NP)=YPP(N1)*DELTA/6.+YPP(NP)*DELTA/3.+(YSMO(NP)-YSMO(N1))/DELTA
      C
      C        PRINT SUMMARY OF SMOOTHED AND UNSMOOTHED DATA
      C
      48    WRITE (JWRITE,86) TITLE
            DO 53 I=1,NP
            YPS(I)=YN(I)
            IF (THETA(I).LE.0.) YN(I)=-YN(I)
            T1=ABS(THETA(I))
            IF (T1.GT.PI2) GO TO 49
            GP=CONS*SIN(T1)
            GPP=CONS*COS(T1)
            GO TO 50
      49    DIF=COSH(T1-PI2)
            DELTA=SINH(T1-PI2)
            GP=CONS/DIF
            GPP=-CONS*DELTA/(DIF*DIF)
```

CARD NO.

```
50    IF (I.EQ.NOSE) GO TO 51
      DYDX=YN(I)/GP
      DY2DX=(YPP(I)*GP-YN(I)*GPP)/(GP**3)
      CURV=ABS(DY2DX)/(SQRT(1.+DYDX**2)**3)
      GO TO 52
51    DYDX=0.1E99
      DY2DX=0.1E99
      CURV=CONS/(YN(I)**2)
      RLE=1./CURV
52    DELTA=THETA(I)*RAD
      DIF=Y(I)-YSMO(I)
      YPPS(I)=YPP(I)
53    WRITE (JWRITE,87) I,DELTA,X(I),Y(I),YUSMO(I),YSMO(I),DIF,YPS(I),YP
     1P(I),DYDX,DY2DX,CURV
      WRITE (JWRITE,89) RLE
C
C     CHECK FOR INTERSECTION OF UPPER AND LOWER SURFACES
C
C        DEFINE ITERATION INTERVAL
      KRT=1001
      N1=2*KRT
      TE=THETA(NP)
      TN=-THETA(1)
      IF (TN.LT.TE) TE=TN
      DIF=TE/FLOAT(KRT-1)
      BB=0.5*DIF
      AA=0.85*TE
      YL1=YU1=YSMO(NOSE)
      TP=TN=0.0
      J1=NOSE
      J2=2
C        DO-LOOP TO SEARCH FOR INTERSECTION
      DO 59 I=2,N1
      IF (TP.LE.AA) TN=TN+DIF
      IF (TP.GT.AA) TN=TN+BB
      IF (TN.GT.TE) GO TO 61
      TI=TN
C        FIND UPPER SURFACE Y-COORDINATE AT THETA = TN
      DO 54 K=J1,NP
      J=K-1
```

CARD NO.

```
          IF (TI.GE.THETA(J).AND.TI.LE.THETA(J+1)) GO TO 55             SO 361
   54     CONTINUE                                                      SO 362
   55     DELTA=THETA(J+1)-THETA(J)                                     SO 363
          T2=THETA(J+1)-TI                                              SO 364
          T1=TI-THETA(J)                                                SO 365
          YU2=YPPS(J)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPS(J+1)*(T1**3/(6.*DE SO 366
         1LTA)-T1*DELTA/6.)+(YSMO(J)*T2+YSMO(J+1)*T1)/DELTA             SO 367
          J1=J                                                          SO 368
          IF (J1.LT.NOSE) J1=NOSE                                       SO 369
C            FIND LOWER SURFACE Y-COORDINATE AT THETA = TN              SO 370
          TI=-TN                                                        SO 371
          DO 56 K=J2,NOSE                                               SO 372
          J=NOSE+1-K                                                    SO 373
          IF (TI.GE.THETA(J).AND.TI.LE.THETA(J+1)) GO TO 57             SO 374
   56     CONTINUE                                                      SO 375
   57     DELTA=THETA(J+1)-THETA(J)                                     SO 376
          T2=THETA(J+1)-TI                                              SO 377
          T1=TI-THETA(J)                                                SO 378
          YL2=YPPS(J)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPS(J+1)*(T1**3/(6.*DE SO 379
         1LTA)-T1*DELTA/6.)+(YSMO(J)*T2+YSMO(J+1)*T1)/DELTA             SO 380
          J2=NOSE+1-J                                                   SO 381
          IF (J2.LT.2) J2=2                                             SO 382
C            COMPUTE THETA FOR INTERSECTION OF STRAIGHT LINE SEGMENTS THRU SO 383
C            LAST TWO POINTS ON EACH SURFACE                            SO 384
          CC=(YU2-YU1-YL2+YL1)/(TN-TP)                                  SO 385
          IF (ABS(CC).LT.1.E-10) GO TO 58                               SO 386
          T1=(YL1-YU1)/CC+TP                                            SO 387
          IF (I.EQ.2) GO TO 58                                          SO 388
C            CHECK TO SEE IF INTERSECTION THETA IS BETWEEN THIS TN-VALUE SO 389
C            AND THE PREVIOUS TN-VALUE                                  SO 390
          IF (T1.GE.TP.AND.T1.LE.TN) GO TO 60                           SO 391
C            CONTINUE TO NEXT TN-VALUE                                  SO 392
   58     YU1=YU2                                                       SO 393
          YL1=YL2                                                       SO 394
          TP=TN                                                         SO 395
   59     CONTINUE                                                      SO 396
          GO TO 61                                                      SO 397
   60     IF (T1.GE.TE) GO TO 61                                        SO 398
C            IF INTERSECTION OCCURS WRITE ERROR MESSAGE AND RETURN TO   SO 399
C            CALLING PROGRAM                                            SO 400
```

CARD NO.

```
           T1=T1*RAD
           WRITE (JWRITE,72) T1
           IERR=1
           RETURN
C
C              FIND LOCATIONS WHERE DY/DX=0.
C
61         KRT=0
           N1=NP-1
           DO 66 I=1,N1
           DELTA=THETA(I+1)-THETA(I)
           AA=(YPP(I)-YPP(I+1))/(2.*DELTA)
           BB=(YPP(I+1)*THETA(I)-YPP(I)*THETA(I+1))/DELTA
           CC=(YPP(I)*THETA(I+1)**2-YPP(I+1)*THETA(I)**2)/(2.*DELTA)+(YPP(I+1
          1)-YPP(I))*DELTA/6.-(YSMO(I+1)-YSMO(I))/DELTA
           GP=BB*BB-4.*AA*CC
           IF (GP) 66,62,62
62         GP=SQRT(GP)
           T1=(-BB+GP)/(2.*AA)
           T2=(-BB-GP)/(2.*AA)
           IF (T1.GE.THETA(I).AND.T1.LE.THETA(I+1)) GO TO 63
           GO TO 64
63         KRT=KRT+1
           WK(KRT,1)=T1
64         IF (T2.GE.THETA(I).AND.T2.LE.THETA(I+1)) GO TO 65
           GO TO 66
65         KRT=KRT+1
           WK(KRT,1)=T2
66         CONTINUE
           IF (KRT.EQ.0) GO TO 70
C                   FIND X/C AND Y/C WHERE DY/DX=0.
           DO 69 I=1,KRT
           CALL INTER (WK(I,1),WK(I,2),NP,THETA,X,1,KTI,0)
           DO 67 J=1,N1
           J1=J
           J2=J+1
           IF (WK(I,1).GE.THETA(J).AND.WK(I,1).LE.THETA(J+1)) GO TO 68
67         CONTINUE
68         AA=THETA(J2)-WK(I,1)
           BB=WK(I,1)-THETA(J1)
```

CARD NO.

```
          WK(I,1)=WK(I,1)*RAD
          DELTA=THETA(J2)-THETA(J1)
   69     WK(I,3)=YPP(J1)*(AA**3/(6.*DELTA)-AA*DELTA/6.)+YPP(J2)*(BB**3/(6.*
         1DELTA)-BB*DELTA/6.)+(YSMO(J1)*AA+YSMO(J2)*BB)/DELTA
   70     CONTINUE
          IF (KRT.GT.0) WRITE (JWRITE,90) (WK(I,2),WK(I,3),WK(I,1),I=1,KRT)
C
C            PRINT RESULTS OF SMOOTHNESS CHECK
C
          IF (ITER.EQ.0) RETURN
          WRITE (JWRITE,91) TITLE,DF
          WRITE (JWRITE,92) (I,A(I,1),A(I,2),I=1,NP)
          WRITE (JWRITE,93) SUM1,SUM2
          RETURN
C
C            PRINT WARNING MESSAGE IF ERROR OCCURRED IN CALL TO CSDS
C
   71     WRITE (JWRITE,94) IERR
          RETURN
C
   72     FORMAT (/5X,108HERROR MESSAGE  ---  SMOOTHING PROCESS RESULTED IN
         1AN INTERSECTION OF THE UPPER AND LOWER SURFACES AT THETA =,F10.3)
   73     FORMAT (1H1,1X,7HTITLE--,2X,8A10//12X,62H--CHECK OF FIRST DERIVATI
         1VES GENERATED FROM IOP=2 INPUT DATA--///9X,1HI,5X,12HDY/DT(INPUT),
         24X,11HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMO),8X,3HDIF/)
   74     FORMAT (5X,I5,5(5X,F10.6))
   75     FORMAT (/25X,16HSUM OF SQUARES =,4X,F10.6,20X,F10.6)
   76     FORMAT (/10X,25HOUTPUT FROM CSDS SELECTED)
   77     FORMAT (/10X,27HOUTPUT FROM LSQSMO SELECTED)
   78     FORMAT (1H1,1X,7HTITLE--,2X,8A10//30X,53H--SUM OF SQUARES GENERATE
         1D DURING SMOOTHING PROCESS--)
   79     FORMAT (/1X,I5,10F12.7)
   80     FORMAT (1H1,1X,7HTITLE--,2X,8A10//30X,67H--SECOND DERIVATIVES W/R
         1THETA GENERATED DURING SMOOTHING PROCESS--/4X,1HI,5X,5HTHETA,10(5X
         2,6HDY2/DT)/)
   81     FORMAT (I5,F10.2,10F11.6)
   82     FORMAT (/1X,14HSUM OF SQUARES,(10F11.6))
   83     FORMAT (/3X,41HSMOOTHING PROCESS HAS NOT CONVERGED AFTER,I4,1X,10H
         1ITERATIONS)
   84     FORMAT (/3X,33HSMOOTHING PROCESS CONVERGED AFTER,I4,1X,10HITERATIO
```

CARD NO.

```
           1NS)                                                         SO 481
    85      FORMAT (/3X,41HSMOOTHING PROCESS BEGAN OSCILLATING AFTER,I4,1X,10H SO 482
           1ITERATIONS)                                                 SO 483
    86      FORMAT (1H1,1X,7HTITLE--,2X,8A10//48X,28H--SMOOTHING OUTPUT SUMMAR SO 484
           1Y--//4X,1HI,5X,5HTHETA,5X,3HX/C,7X,3HY/C,7X,4HYT/C,5X,6HYSMO/C,4X, SO 485
           25HDELTA,7X,3HYPS,6X,4HYPPS,8X,5HDY/DX,7X,11HD(DY/DX)/DX,6X,    SO 486
           19HCURVATURE/)                                               SO 487
    87      FORMAT (I5,F10.2,7F10.6,3E15.6)                             SO 488
    88      FORMAT (/3X,58HSUM OF SQUARES FROM LEAST SQUARES CUBIC SPLINE SMOO SO 489
           1THING =,E12.4)                                              SO 490
    89      FORMAT (/3X,22HLEADING-EDGE RADIUS/C=,F10.6)                SO 491
    90      FORMAT (/3X,16HDY/DX=0. AT X/C=,F10.6,5X,4HY/C=,F10.6,5X,6HTHETA=, SO 492
           1F10.3)                                                      SO 493
    91      FORMAT (1H1,1X,7HTITLE--,2X,8A10//12X,29HCHECK OF SMOOTHED COORDIN SO 494
           1ATES,3X,3HDF=,F10.6//9X,1HI,5X,20H(YSMO/C-CHECK VALUE),7X,  SO 495
           218H(YPPS-CHECK VALUE)/)                                     SO 496
    92      FORMAT (5X,I5,10X,F10.6,15X,F10.6)                          SO 497
    93      FORMAT (/5X,15HSUM OF SQUARES=,F10.6,15X,F10.6)             SO 498
    94      FORMAT (/3X,21HINPUT ERROR -- POINT ,I3,18H IS NOT INCREASING/)  SO 499
            END                                                         SO 500-
```

CARD NO.

```
      SUBROUTINE YNEW (THETA,YPP,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JW
     1RITE,IPT)
C
C     ROUTINE TO COMPUTE NEW Y/C COORDINATES USING AN ITERATION
C     PROCEDURE THAT INSURES A DESIRED Y/C COORDINATE AT THE NOSE
C     (IPT=0) OR THAT INSURES CONTINUITY OF THE FIRST DERIVATIVE W/R TO
C     THETA AT THE NOSE (IPT=1)
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
C        DIMENSION THETA, YPP, Y, AND DUM BY NP AND WK BY 2*NP IN
C        CALLING PROGRAM
      DIMENSION THETA(1), YPP(1), Y(1), DUM(1), WK(1)
C
C     INITIALIZE ITERATION PARAMETERS
C
      NMAX=20
      N1=-1
      DELTA=0.
      T1=THETA(NOSE)-THETA(NOSE-1)
      T2=THETA(NOSE+1)-THETA(NOSE)
      DO 1 I=1,NP
    1 DUM(I)=YPP(I)
C
C     ITERATION LOOP TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND
C     DERIVATIVE TO INSURE THAT THE DESIRED CONVERGENCE OPTION AT
C     THE NOSE IS OBTAINED
C
    2 N1=N1+1
      IF (N1.GT.NMAX) GO TO 11
      IF (IPT.EQ.1) GO TO 3
C        IF IPT=0, COMPUTE UPPER AND LOWER SURFACE Y/C COORDINATES
C        CONCURRENTLY
      CALL INVY (THETA,DUM,1,NP,Y,YLTE,YUTE,WK)
C        COMPUTE DIFFERENCE BETWEEN OUTPUT AND DESIRED Y/C COORDINATE
C        AT THE NOSE
      DIF=Y(NOSE)-YNOSE
      GO TO 4
C        IF IPT=1, COMPUTE UPPER AND LOWER SURFACE Y/C COORDINATES
C        CONSECUTIVELY
```

CARD NO.

```
      3     CALL INVY (THETA,DUM,NOSE,NP,Y,YNOSE,YUTE,WK)
            CALL INVY (THETA,DUM,1,NOSE,Y,YLTE,YNOSE,WK)
      C        COMPUTE DIFFERENCE BETWEEN UPPER AND LOWER SURFACE FIRST
      C        DERIVATIVES AT THE NOSE
            AA=-DUM(NOSE)*T2/3.-DUM(NOSE+1)*T2/6.+(Y(NOSE+1)-Y(NOSE))/T2
            BB=DUM(NOSE-1)*T1/6.+DUM(NOSE)*T1/3.+(Y(NOSE)-Y(NOSE-1))/T1
            DIF=AA-BB
      C        CHECK FOR CONVERGENCE
      4     IF (ABS(DIF).LE.EPS) GO TO 9
      C        COMPUTE ADJUSTMENT VALUE TO SECOND DERIVATIVE
            IF (N1.EQ.0) GO TO 6
            IF (DIF.EQ.DIFP) GO TO 5
            SP=(DELTA-DELTAP)/(DIF-DIFP)
            DELTAP=DELTA
            DIFP=DIF
            DELTA=DELTA-DIF*SP
            GO TO 7
      5     DELTA=0.5*(DELTA+DELTAP)
            GO TO 7
      6     DELTAP=DELTA
            DIFP=DIF
            DELTA=DELTA+DIF
      C        ADD ADJUSTMENT VALUE TO SECOND DERIVATIVE
      7     DO 8 I=1,NP
      8     DUM(I)=YPP(I)+DELTA
      C        CONTINUE TO ITERATE
            GO TO 2
      C
      C     PRINT CONVERGENCE MESSAGE
      C
      9     WRITE (JWRITE,14) N1,DELTA
      C        REDEFINE THE SECOND DERIVATIVE
            DO 10 I=1,NP
      10    YPP(I)=DUM(I)
            IF (IPT.EQ.1) GO TO 12
            GO TO 13
      C
      C     PRINT NON-CONVERGENCE MESSAGE
      C
      11    N1=N1-1
```

CARD NO.

```
            WRITE (JWRITE,15) N1
      C
      C     COMPUTE NEW UPPER AND LOWER SURFACE Y/C COORDINATES CONCURRENTLY
      C
      12    CALL INVY (THETA,YPP,1,NP,Y,YLTE,YUTE,WK)
      C
      C     RETURN TO CALLING PROGRAM
      C
      13    RETURN
      C
      14    FORMAT (/3X,88HITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTME
           1NT TO SECOND DERIVATIVE CONVERGED IN ,I3,23H ITERATIONS AND DELTA
           2=,E12.4)
      15    FORMAT (///10X,40HWARNING THE FOLLOWING ERROR HAS OCCURRED//3X,95H
           1ITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND DE
           2RIVATIVE DID NOT CONVERGE IN ,I3,11H ITERATIONS)
            END
```

```
      SUBROUTINE INVY (X,YPP,NS,NE,Y,YSTART,YEND,A)
C
C     THIS ROUTINE COMPUTES Y VALUES FROM KNOWN SECOND DERIVATIVES AND
C     END CONDITIONS
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
C        IN CALLING PROGRAM DIMENSION X, YPP, AND Y BY NE AND A BY 2*NE
C
      DIMENSION X(1), YPP(1), Y(1), A(NE,2)
C
C     SET END CONDITIONS
C
      Y(NS)=YSTART
      Y(NE)=YEND
C
C     PERFORM FORWARD ELIMINATION
C
      A(1,1)=YSTART
      A(1,2)=0.0
      N=NE-NS+1
      N1=N-1
      DO 1 I=2,N1
      J=NS+I-1
      H1=X(J)-X(J-1)
      H2=X(J+1)-X(J)
      C=(H1*YPP(J-1)/6.+(H1+H2)*YPP(J)/3.+H2*YPP(J+1)/6.)*H1*H2
      D=-H2*(A(I-1,2)+1.)-H1
      A(I,2)=H1/D
    1 A(I,1)=(C-H2*A(I-1,1))/D
C
C     PERFORM BACK SUBSTITUTION
C
      J=NE
      DO 2 I=2,N1
      J=J-1
      N=N-1
    2 Y(J)=A(N,1)-A(N,2)*Y(J+1)
C
C     RETURN TO CALLING PROGRAM
```

```
CARD NO.

  41          C                                                       IV  41
                    RETURN                                            IV  42
                    END                                               IV  43-
```

CARD NO.

```
      SUBROUTINE LSQSMO (X,Y,W,YN,YP,YPP,N,IMAX,JMAX,NOSE,WT,EPS,IERR)   LM   1
C                                                                        LM   2
C     THIS SUBROUTINE IS USED TO SMOOTH X AND Y BY CONSECUTIVELY FITTING LM   3
C     A LEAST SQUARES POLYNOMIAL OF DEGREE 4 THRU 7 POINTS AT A TIME     LM   4
C                                                                        LM   5
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          LM   6
C                                                                        LM   7
      DIMENSION X(1), Y(1), W(1), YN(1), YP(1), YPP(1)                   LM   8
C                                                                        LM   9
      DIMENSION XI(7), YI(7), WW(7), A(5,6), B(5)                        LM  10
C                                                                        LM  11
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                 LM  12
C                                                                        LM  13
C     CHECK NOSE REGION FOR SYMMETRY                                     LM  14
C                                                                        LM  15
      ISYM=1                                                             LM  16
      DO 1 I=1,3                                                         LM  17
      IF (ABS(X(NOSE-I)+X(NOSE+I)).GT.EPS) ISYM=0                        LM  18
      IF (ABS(Y(NOSE-I)+Y(NOSE+I)).GT.EPS) ISYM=0                        LM  19
    1 CONTINUE                                                           LM  20
      IERR=0                                                             LM  21
C                                                                        LM  22
C     FIT A LEAST SQUARES POLYNOMIAL OF DEGREE 4 THRU 7 POINTS           LM  23
C                                                                        LM  24
      DO 14 I=1,N                                                        LM  25
C        LOAD 7 POINTS FOR LEAST SQUARES POLYNOMIAL FIT                  LM  26
      IF (I.LT.4) GO TO 2                                                LM  27
      IF (I.GT.N-3) GO TO 3                                              LM  28
      J1=I-3                                                             LM  29
      J2=I+3                                                             LM  30
      GO TO 4                                                            LM  31
    2 J1=1                                                               LM  32
      J2=7                                                               LM  33
      GO TO 4                                                            LM  34
    3 J1=N-6                                                             LM  35
      J2=N                                                               LM  36
    4 KK=0                                                               LM  37
      IF (ISYM.EQ.0) GO TO 7                                             LM  38
      IF (I.GT.NOSE-3.AND.I.LE.NOSE) GO TO 5                             LM  39
      IF (I.LT.NOSE+3.AND.I.GT.NOSE) GO TO 6                             LM  40
```

CARD NO.

```
            GO TO 7
      5     J1=NOSE-6
            J2=NOSE
            GO TO 7
      6     J1=NOSE
            J2=NOSE+6
      7     DO 8 L=J1,J2
            J=L
            IF (I.LE.NOSE) J=J1+J2-L
            KK=KK+1
            WW(KK)=1.0
            IF (I.EQ.J) WW(KK)=W(I)
            IF (J.EQ.IMAX.OR.J.EQ.JMAX) WW(KK)=WT*W(J)
            XI(KK)=X(J)
      8     YI(KK)=Y(J)
            IF (I.LE.4) WW(7)=7.*W(1)
            IF (I.GE.N-3) WW(7)=7.*W(N)
      C        COMPUTE LEAST SQUARES MATRIX
            DO 9 L=1,5
            DO 9 J=1,6
      9     A(L,J)=0.
            DO 11 K=1,7
            T1=1.
            DO 11 J=1,5
            T2=T1
            DO 10 L=1,5
            A(J,L)=A(J,L)+T2*WW(K)
      10    T2=T2*XI(K)
            A(J,6)=A(J,6)-YI(K)*T1*WW(K)
      11    T1=T1*XI(K)
      C        SOLVE FOR COEFFICIENTS OF LEAST SQUARES POLYNOMIAL
            DO 12 K=1,4
            DO 12 J=K,4
            T1=A(J+1,K)/A(K,K)
            DO 12 L=K,6
      12    A(J+1,L)=A(J+1,L)-A(K,L)*T1
            B(5)=-A(5,6)/A(5,5)
            DO 13 L=2,5
            K=6-L
            B(K)=-A(K,6)/A(K,K)
```

CARD NO.

```
      K1=K+1
      DO 13 J=K1,5
13    B(K)=B(K)-B(J)*A(K,J)/A(K,K)
C        COMPUTE NEW Y , FIRST , AND SECOND DERIVATIVE
      YN(I)=(((B(5)*X(I)+B(4))*X(I)+B(3))*X(I)+B(2))*X(I)+B(1)
      YP(I)=((4.*B(5)*X(I)+3.*B(4))*X(I)+2.*B(3))*X(I)+B(2)
      YPP(I)=(12.*B(5)*X(I)+6.*B(4))*X(I)+2.*B(3)
14    CONTINUE
      IF (ISYM.EQ.0) RETURN
      YN(NOSE)=0.0
      YPP(NOSE)=0.0
      YP(NOSE)=1.0
      RETURN
      END
```

CARD NO.

```
      SUBROUTINE CSDS (MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)               CS   1
C*********************************************************************** CS   2
C*                                                                     * CS   3
C*    PURPOSE:                                                         * CS   4
C*                  SUBROUTINE CSDS FITS A SMOOTH CUBIC SPLINE TO A    * CS   5
C*                  UNIVARIATE FUNCTION.  DATA MAY BE UNEQUALLY SPACED.* CS   6
C*                                                                     * CS   7
C*        USE:                                                         * CS   8
C*                  CALL CSDS(MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)        * CS   9
C*                                                                     * CS  10
C*        MAX       INPUT INTEGER SPECIFYING THE MAXIMUM NUMBER OF DATA * CS  11
C*                  POINTS FOR THE INDEPENDENT VARIABLE.               * CS  12
C*                                                                     * CS  13
C*         IX       INPUT INTEGER SPECIFYING THE ACTUAL NUMBER OF DATA * CS  14
C*                  POINTS FOR THE INDEPENDENT VARIABLE.  IX≤MAX.      * CS  15
C*                                                                     * CS  16
C*          X       ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST   * CS  17
C*                  IX IN THE CALLING PROGRAM.  UPON ENTRY TO CSDS,    * CS  18
C*                  X(I) MUST CONTAIN THE VALUE OF THE INDEPENDENT     * CS  19
C*                  VARIABLE AT POINT I.                               * CS  20
C*                                                                     * CS  21
C*          F       ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST   * CS  22
C*                  IX IN THE CALLING PROGRAM.  UPON ENTRY TO CSDS,    * CS  23
C*                  F(I) MUST CONTAIN THE VALUE OF THE FUNCTION AT     * CS  24
C*                  POINT X(I).                                        * CS  25
C*                                                                     * CS  26
C*         DF       ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST   * CS  27
C*                  IX IN THE CALLING PROGRAM.  UPON ENTRY TO CSDS,    * CS  28
C*                  DF(I) MUST CONTAIN AN ESTIMATE OF THE STANDARD     * CS  29
C*                  DEVIATION OF F(I).                                 * CS  30
C*                                                                     * CS  31
C*          S       A NON-NEGATIVE INPUT PARAMETER WHICH CONTROLS THE  * CS  32
C*                  EXTENT OF SMOOTHING.  S SHOULD BE IN THE RANGE     * CS  33
C*                  (IX-(2*IX)**.5)<S<(IX+(2*IX)**.5).                 * CS  34
C*                                                                     * CS  35
C*        IPT       INPUT INITIALIZATION PARAMETER.  THE USER MUST     * CS  36
C*                  SPECIFY IPT=-1 WHENEVER A NEW X ARRAY IS           * CS  37
C*                  INPUT.  THE ROUTINE WILL ALSO CHECK TO INSURE THAT * CS  38
C*                  THE X ARRAY IS IN STRICTLY INCREASING ORDER.       * CS  39
C*                                                                     * CS  40
```

CARD NO.

```
C*        COEF     A TWO-DIMENSIONAL OUTPUT ARRAY DIMENSIONED (MAX,4)  * CS  41
C*                 IN THE CALLING PROGRAM.  UPON RETURN, COEF(I,J)     * CS  42
C*                 CONTAINS THE J-TH COEFFICIENT OF THE SPLINE FOR     * CS  43
C*                 THE INTERVAL BEGINNING AT POINT X(I).  THE          * CS  44
C*                 FUNCTIONAL VALUE OF THE SPLINE AT ABSCISSA X1,      * CS  45
C*                 WHERE X(I)<X1<X(I+1), IS GIVEN BY:                  * CS  46
C*                    F(X1)=((COEF(I,4)*H+COEF(I,3))*H+COEF(I,2))*H    * CS  47
C*                    +COEF(I,1)                                       * CS  48
C*                 WHERE H=X1-X(I)                                     * CS  49
C*                                                                     * CS  50
C*          WK     A ONE-DIMENSIONAL WORK AREA ARRAY DIMENSIONED AT    * CS  51
C*                 LEAST (7*IX+9) IN THE CALLING PROGRAM.              * CS  52
C*                                                                     * CS  53
C*        IERR     OUTPUT ERROR PARAMETER:                             * CS  54
C*                 =0   NORMAL RETURN.  NO ERROR DETECTED.             * CS  55
C*                 =J   THE J-TH ELEMENT OF THE X ARRAY IS NOT IN      * CS  56
C*                      STRICTLY INCREASING ORDER.                     * CS  57
C*                 =-1  THERE ARE LESS THAN FOUR VALUES IN THE X ARRAY.* CS  58
C*                                                                     * CS  59
C*                 UPON RETURN FROM CSDS, THIS PARAMETER SHOULD BE     * CS  60
C*                 TESTED IN THE CALLING PROGRAM.                      * CS  61
C*                                                                     * CS  62
C*                                                                     * CS  63
C*                                                                     * CS  64
C*    REQUIRED ROUTINES                  -NONE                         * CS  65
C*                                                                     * CS  66
C*    SOURCE                             IMSL ROUTINE ICSSMU MODIFIED BY * CS  67
C*                                       COMPUTER SCIENCES CORPORATION * CS  68
C*                                                                     * CS  69
C*    LANGUAGE                           -FORTRAN                      * CS  70
C*                                                                     * CS  71
C*    DATE RELEASED                      SEPTEMBER 5, 1973             * CS  72
C*                                                                     * CS  73
C*    LATEST REVISION                    MARCH 1975                    * CS  74
C********************************************************************** CS  75
C                                                                        CS  76
C                                                                        CS  77
      DIMENSION X(1), F(1), DF(1), COEF(MAX,4), WK(1)                    CS  78
C                                                                        CS  79
C                                  SET UP WORKING AREAS                  CS  80
```

CARD NO.

```
C
      IERR=0
      IF (IPT.NE.-1) GO TO 4
      IPT=0
      IF (IX.LT.4) GO TO 1
      GO TO 2
    1 IERR=-1
      RETURN
    2 IX1=IX-1
      DO 3 I=1,IX1
      IF (X(I+1)-X(I).GT.0) GO TO 3
      IERR=I+1
      RETURN
    3 CONTINUE
      NP1=IX+1
      IB1=NP1
      IB2=IB1+NP1
      IB3=IB2+NP1+1
      IB4=IB3+NP1
      IB5=IB4+NP1
      IB6=IB5+NP1+1
      WK(1)=0.
      WK(2)=0.
      WK(IB2)=0.
      WK(IB3)=0.
      IJK2=IB2+NP1
      WK(IJK2)=0.
      IJK5=IB5+1
      WK(IJK5)=0.
      IJK5=IB5+2
      WK(IJK5)=0.
      WK(IB6)=0.
      IJK5=IB5+NP1
      WK(IJK5)=0.
    4 CONTINUE
      P=0.
      H=X(2)-X(1)
      F2=-S
      FF=(F(2)-F(1))/H
      IF (IX.LT.3) GO TO 10
```

CARD NO.

```
              DO 5 I=3,IX                                                 CS 121
              G=H                                                         CS 122
              H=X(I)-X(I-1)                                               CS 123
              E=FF                                                        CS 124
              FF=(F(I)-F(I-1))/H                                          CS 125
              COEF(I-1,1)=FF-E                                            CS 126
              IJK3=IB3+I                                                  CS 127
              WK(IJK3)=(G+H)*.66666666666667                              CS 128
              IJK4=IB4+I                                                  CS 129
              WK(IJK4)=H/3.                                               CS 130
              IJK2=IB2+I                                                  CS 131
              WK(IJK2)=DF(I-2)/G                                          CS 132
              WK(I)=DF(I)/H                                               CS 133
              IJK1=IB1+I                                                  CS 134
              WK(IJK1)=-DF(I-1)/G-DF(I-1)/H                               CS 135
    5         CONTINUE                                                    CS 136
              DO 6 I=3,IX                                                 CS 137
              IJK1=IB1+I                                                  CS 138
              IJK2=IB2+I                                                  CS 139
              COEF(I-1,2)=WK(I)*WK(I)+WK(IJK1)*WK(IJK1)+WK(IJK2)*WK(IJK2) CS 140
              COEF(I-1,3)=WK(I)*WK(IJK1+1)+WK(IJK1)*WK(IJK2+1)            CS 141
              COEF(I-1,4)=WK(I)*WK(IJK2+2)                                CS 142
    6         CONTINUE                                                    CS 143
    C                                                                     CS 144
    C                             NEXT ITERATION                          CS 145
    C                                                                     CS 146
    7         IF (IX.LT.3) GO TO 10                                       CS 147
              DO 8 I=3,IX                                                 CS 148
              IJK1=IB1+I-1                                                CS 149
              IJK0=I-1                                                    CS 150
              WK(IJK1)=FF*WK(IJK0)                                        CS 151
              IJK2=IB2+I-2                                                CS 152
              IJK0=I-2                                                    CS 153
              WK(IJK2)=G*WK(IJK0)                                         CS 154
              IJK0=I                                                      CS 155
              IJK3=IB3+I                                                  CS 156
              WK(IJK0)=1./(P*COEF(I-1,2)+WK(IJK3)-FF*WK(IJK1)-G*WK(IJK2)) CS 157
              IJK5=IB5+I                                                  CS 158
              IJKN=IJK5-1                                                 CS 159
              IJK0=IJKN-1                                                 CS 160
```

CARD NO.

```
      WK(IJK5)=COEF(I-1,1)-WK(IJK1)*WK(IJKN)-WK(IJK2)*WK(IJKO)      CS 161
      IJK4=IB4+I                                                    CS 162
      FF=P*COEF(I-1,3)+WK(IJK4)-H*WK(IJK1)                          CS 163
      G=H                                                           CS 164
      H=COEF(I-1,4)*P                                               CS 165
    8 CONTINUE                                                      CS 166
      DO 9 I=3,IX                                                   CS 167
      J=IX-I+3                                                      CS 168
      IJK5=IB5+J                                                    CS 169
      IJK6=IJK5+1                                                   CS 170
      IJK7=IJK6+1                                                   CS 171
      IJK1=IB1+J                                                    CS 172
      IJK2=IB2+J                                                    CS 173
      WK(IJK5)=WK(J)*WK(IJK5)-WK(IJK1)*WK(IJK6)-WK(IJK2)*WK(IJK7)   CS 174
    9 CONTINUE                                                      CS 175
   10 E=0                                                           CS 176
      H=0                                                           CS 177
C                                                                   CS 178
C                               COMPUTE U AND ACCUMULATE E          CS 179
C                                                                   CS 180
      DO 11 I=2,IX                                                  CS 181
      G=H                                                           CS 182
      IJK5=IB5+I                                                    CS 183
      H=(WK(IJK5+1)-WK(IJK5))/(X(I)-X(I-1))                         CS 184
      IJK6=IB6+I                                                    CS 185
      WK(IJK6)=(H-G)*DF(I-1)*DF(I-1)                                CS 186
      E=E+WK(IJK6)*(H-G)                                            CS 187
   11 CONTINUE                                                      CS 188
      G=-H*DF(IX)*DF(IX)                                            CS 189
      IJK6=IB6+NP1                                                  CS 190
      WK(IJK6)=G                                                    CS 191
      E=E-G*H                                                       CS 192
      G=F2                                                          CS 193
      F2=E*P*P                                                      CS 194
      IF (F2.GE.S.OR.F2.LE.G) GO TO 14                              CS 195
      FF=0.                                                         CS 196
      IJK6=IB6+2                                                    CS 197
      H=(WK(IJK6+1)-WK(IJK6))/(X(2)-X(1))                           CS 198
      IF (IX.LT.3) GO TO 13                                         CS 199
      DO 12 I=3,IX                                                  CS 200
```

CARD NO.

```
      G=H
      IJK6=IB6+I
      H=(WK(IJK6+1)-WK(IJK6))/(X(I)-X(I-1))
      IJK1=IB1+I-1
      IJK2=IB2+I-2
      G=H-G-WK(IJK1)*WK(I-1)-WK(IJK2)*WK(I-2)
      FF=FF+G*WK(I)*G
      WK(I)=G
   12 CONTINUE
   13 H=E-P*FF
      IF (H.LE.0) GO TO 14
C
C                                 UPDATE THE LAGRANGE MULTIPLIER P
C                                    FOR THE NEXT ITERATION
C
      P=P+(S-F2)/((SQRT(S/E)+P)*H)
      GO TO 7
C
C                                 IF E LESS THAN OR EQUAL TO S,
C                                 COMPUTE THE COEFFICIENTS AND RETURN.
C
   14 DO 15 I=2,NP1
      IJK6=IB6+I
      COEF(I-1,1)=F(I-1)-P*WK(IJK6)
      IJK5=IB5+I
      COEF(I-1,3)=WK(IJK5)
   15 CONTINUE
      DO 16 I=2,IX
      H=X(I)-X(I-1)
      COEF(I-1,4)=(COEF(I,3)-COEF(I-1,3))/(3.*H)
      COEF(I-1,2)=(COEF(I,1)-COEF(I-1,1))/H-(H*COEF(I-1,4)+COEF(I-1,3))*
     1H
   16 CONTINUE
      RETURN
      END
```

CARD NO.

```
      SUBROUTINE PCARD (IPUNCH,X,Y,W,THETA,YSMO,YPS,YPPS,NOSE,NP,CHORD,T
     1ITLE)
C
C     ROUTINE TO PUNCH OUTPUT DATA    (TAPE 1 IS PUNCH FILE)
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION TITLE(8), X(1), Y(1), W(1), THETA(1), YSMO(1), YPS(1), Y
     1PPS(1)
C
      COMMON /HLM/ DX(200),DY(200),DW(200)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
      IF (IPUNCH.LE.0.OR.IPUNCH.GE.5) RETURN
C
C     PUNCH TITLE CARD
C
      WRITE (JWRITE,10) IPUNCH,TITLE
      WRITE (1,11) TITLE
C
C     DETERMINE OUTPUT PUNCH OPTION
C
      IOP=0
      IF (IPUNCH.EQ.2) IOP=1
      IF (IPUNCH.EQ.3) IOP=2
      IF (IPUNCH.EQ.4) IOP=3
      WRITE (JWRITE,12) IOP
      XI=FLOAT(IOP)
      WRITE (1,13) XI
C
C     PUNCH UPPER SURFACE QUANTITIES
C
      J=KP=0
      DO 1 I=NOSE,NP
      J=J+1
      DW(J)=W(I)
      IF (W(I).GT.1.0) KP=1
```

CARD NO.

```
      IF (IOP.EQ.0) DX(J)=X(I)*CHORD
      IF (IOP.NE.0) DX(J)=THETA(I)*RAD
      IF (IOP.EQ.0) DY(J)=YSMO(I)*CHORD
      IF (IOP.EQ.1) DY(J)=YSMO(I)
      IF (IOP.EQ.2) DY(J)=YPS(I)
      IF (IOP.EQ.3) DY(J)=YPPS(I)
    1 CONTINUE
      WRITE (JWRITE,14) J
      XI=FLOAT(J)
      WRITE (1,15) XI
      IF (IOP.EQ.0) WRITE (JWRITE,16) (DX(I),I=1,J)
      IF (IOP.NE.0) WRITE (JWRITE,7) (DX(I),I=1,J)
      WRITE (JWRITE,17) (DY(I),I=1,J)
      IF (KP.EQ.1) WRITE (JWRITE,21) (DW(I),I=1,J)
      DO 3 I=1,J
      IF (IOP.NE.0) GO TO 2
      IF (DW(I).GT.1.0) WRITE (1,22) DX(I),DY(I),DW(I)
      IF (DW(I).LE.1.0) WRITE (1,18) DX(I),DY(I)
      GO TO 3
    2 IF (DW(I).GT.1.0) WRITE (1,8) DX(I),DY(I),DW(I)
      IF (DW(I).LE.1.0) WRITE (1,9) DX(I),DY(I)
    3 CONTINUE
C
C     PUNCH LOWER SURFACE QUANTITIES
C
      J=KP=0
      DO 4 I=1,NOSE
      J=J+1
      K=NOSE+1-I
      DW(J)=W(K)
      IF (W(K).GT.1.0) KP=1
      IF (IOP.EQ.0) DX(J)=X(K)*CHORD
      IF (IOP.NE.0) DX(J)=THETA(K)*RAD
      IF (IOP.EQ.0) DY(J)=YSMO(K)*CHORD
      IF (IOP.EQ.1) DY(J)=YSMO(K)
      IF (IOP.EQ.2) DY(J)=YPS(K)
      IF (IOP.EQ.3) DY(J)=YPPS(K)
    4 CONTINUE
      WRITE (JWRITE,19) J
      XI=FLOAT(J)
```

CARD NO.

```
              WRITE (1,15) XI
              IF (IOP.EQ.0) WRITE (JWRITE,16) (DX(I),I=1,J)
              IF (IOP.NE.0) WRITE (JWRITE,7) (DX(I),I=1,J)
              WRITE (JWRITE,17) (DY(I),I=1,J)
              IF (KP.EQ.1) WRITE (JWRITE,21) (DW(I),I=1,J)
              DO 6 I=1,J
              IF (IOP.NE.0) GO TO 5
              IF (DW(I).GT.1.0) WRITE (1,22) DX(I),DY(I),DW(I)
              IF (DW(I).LE.1.0) WRITE (1,18) DX(I),DY(I)
              GO TO 6
      5       IF (DW(I).GT.1.0) WRITE (1,8) DX(I),DY(I),DW(I)
              IF (DW(I).LE.1.0) WRITE (1,9) DX(I),DY(I)
      6       CONTINUE
      C
      C       PUNCH YLTE AND YUTE
      C
              IF (IOP.LE.1) RETURN
              YLTE=YSMO(1)
              YNOSE=YSMO(NOSE)
              YUTE=YSMO(NP)
              WRITE (JWRITE,20) YLTE,YNOSE,YUTE
              WRITE (1,18) YLTE,YNOSE,YUTE
      C
      C       RETURN TO CALLING PROGRAM
      C
              RETURN
      C
      7       FORMAT (/3X,4HTH =,8F10.5/(7X,8F10.5))
      8       FORMAT (F10.5,F10.6,F10.2)
      9       FORMAT (F10.5,F10.6)
      10      FORMAT (1H1,10X,36HTHE FOLLOWING DATA HAVE BEEN PUNCHED,5X,7HIPUNC
             1H=,I4//3X,8A10)
      11      FORMAT (8A10)
      12      FORMAT (/3X,5HIOP =,I4)
      13      FORMAT (30X,F10.2)
      14      FORMAT (/3X,4HNU =,I4)
      15      FORMAT (F10.2)
      16      FORMAT (/3X,4HDX =,8F10.6/(7X,8F10.6))
      17      FORMAT (/3X,4HDY =,8F10.6/(7X,8F10.6))
      18      FORMAT (3F10.6)
```

CARD NO.

```
      19     FORMAT (/3X,4HNL =,I4)                                                  PH 121
      20     FORMAT (/3X,6HYLTE =,F10.6,5X,7HYNOSE =,F10.6,5X,6HYUTE =,F10.6)        PH 122
      21     FORMAT (/3X,4HDW =,8F10.2/(7X,8F10.2))                                  PH 123
      22     FORMAT (2F10.6,F10.2)                                                   PH 124
             END                                                                     PH 125-
```

CARD NO.

```
      SUBROUTINE PLOTAF (THETA,Y,YSMO,YPS,YPPS,NP,TITLE,IPLOT)
C
C     THIS ROUTINE PLOTS INPUT AND SMOOTHED Y/C, SMOOTHED YPS, AND
C     SMOOTHED YPPS VERSUS THETA. ALSO PLOTS INPUT AND SMOOTHED Y/C
C     VERSUS X/C.
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION TITLE(8), THETA(1), Y(1), YSMO(1), YPS(1), YPPS(1)
C
      COMMON /HLM/ XI(363),YI(363),TI(363)
C
      COMMON /SMY/ YPSI(363)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
      DATA NM/361/,SIZ/.40/,ISIZ/3/
C
C     SINH(X)=(EXP(X)-EXP(-X))/2.
C
C        INTERPOLATE NM SMOOTHED COORDINATES Y/C AND YPS VALUES
C
      YMAX=0.0
      DP=(THETA(NP)-THETA(1))/FLOAT(NM-1)
      YP=THETA(1)-DP
      M=2
      DO 5 I=1,NM
      YP=YP+DP
      IF (YP.LT.THETA(1)) YP=THETA(1)
      IF (YP.GT.THETA(NP)) YP=THETA(NP)
      TI(I)=YP*RAD
      IF (M.LT.2) M=2
      TP=ABS(YP)
      IF (TP.LE.PI2) GO TO 1
      XI(I)=CONS*(ATAN(SINH(TP-PI2))+1.)
      GO TO 2
    1 XI(I)=CONS*(1.-COS(TP))
    2 DO 3 K=M,NP
```

CARD NO.

```
              J=K-1
              IF (YP.GE.THETA(J).AND.YP.LE.THETA(K)) GO TO 4
    3         CONTINUE
    4         M=J
              DELTA=THETA(J+1)-THETA(J)
              X2=THETA(J+1)-YP
              X1=YP-THETA(J)
              YI(I)=YPPS(J)*(X2**3/(6.*DELTA)-X2*DELTA/6.)+YPPS(J+1)*(X1**3/(6.*
             1DELTA)-X1*DELTA/6.)+(YSMO(J)*X2+YSMO(J+1)*X1)/DELTA
              YPSI(I)=YPPS(J)*(DELTA/6.-X2*X2/(2.*DELTA))+YPPS(J+1)*(X1*X1/(2.*D
             1ELTA)-DELTA/6.)+(YSMO(J+1)-YSMO(J))/DELTA
              IF (ABS(YI(I)).GE.YMAX) YMAX=ABS(YI(I))
    5         CONTINUE
    C
    C            PRINT INTERPOLATED Y/C-COORDINATES
    C
              IF (IPRINT.NE.0) GO TO 6
              WRITE (JWRITE,15) TITLE
              WRITE (JWRITE,16) (I,TI(I),XI(I),YI(I),I=1,NM)
    C
    C            DETERMINE SCALING FACTOR FOR Y/C AXIS
    C
    6         YSCALE=0.1
              IF (YMAX.LE.0.06) YSCALE=0.01
              IF ((YMAX.GT.0.06).AND.(YMAX.LE.0.12)) YSCALE=0.02
              IF ((YMAX.GT.0.12).AND.(YMAX.LE.0.24)) YSCALE=0.04
              IF ((YMAX.GT.0.24).AND.(YMAX.LE.0.30)) YSCALE=0.05
              YMIN=-6.*YSCALE
              YSAV=YSCALE
    C
    C            DRAW AND LABEL Y/C AND THETA AXIS
    C
              IF (IPLOT.EQ.2) GO TO 11
              CALL CALPLT (2.,1.,-3)
              CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)
              CALL AXES (0.,2.,0.,36.,-180.,10.,-2.,1.,10HTHETA,DEG.,SIZ,-10,0)
              CALL AXES (0.,2.,90.,12.,YMIN,YSCALE,-1.,0.,3HY/C,SIZ,3,2)
              CALL NOTATE (1.0,13.1,.4,2,0.,-1)
              CALL NOTATE (1.5,12.9,SIZ,8HSMOOTHED,0.,8)
              CALL NOTATE (1.0,13.7,.4,3,0.,-1)
```

CARD NO.

```
            CALL NOTATE (1.5,13.5,SIZ,5HINPUT,0.,5)
            CALL CALPLT (0.,8.,-3)
      C
      C        PLOT INPUT Y/C-COORDINATES VS THETA
      C
            DO 7 I=1,NP
            TP=THETA(I)*RAD/10.+18.0
            YP=Y(I)/YSCALE
            CALL PNTPLT (TP,YP,22,ISIZ)
      7     CONTINUE
      C
      C        PLOT SMOOTHED Y/C-COORDINATES VS THETA
      C
            TI(NM+1)=-180.0
            TI(NM+2)=10.0
            YI(NM+1)=0.
            YI(NM+2)=YSCALE
            CALL LINE (TI,YI,NM,1,0,0,0.)
      C
      C        DETERMINE SCALING FACTOR FOR FIRST DERIVATIVE AXIS (YP AXIS)
      C
            YMAX=0.0
            DO 8 I=1,NM
            IF (ABS(YPSI(I)).GT.YMAX) YMAX=ABS(YPSI(I))
      8     CONTINUE
            CSCALE=.1
            IF ((YMAX.LE.0.30).AND.(YMAX.GT.0.24)) CSCALE=.05
            IF ((YMAX.LE.0.24).AND.(YMAX.GT.0.12)) CSCALE=.04
            IF ((YMAX.LE.0.12).AND.(YMAX.GT.0.06)) CSCALE=.02
            IF ((YMAX.LE.0.06).AND.(YMAX.GE.0.00)) CSCALE=.01
            CMIN=-6.*CSCALE
      C
      C        DETERMINE SCALING FACTOR FOR SECOND DERIVATIVE AXIS (YPP AXIS)
      C
            YMAX=0.0
            DO 9 I=1,NP
            IF (ABS(YPPS(I)).GT.YMAX) YMAX=ABS(YPPS(I))
      9     CONTINUE
            YSCALE=1.
            IF ((YMAX.LE.3.00).AND.(YMAX.GT.2.40)) YSCALE=.5
```

CARD NO.

```
              IF ((YMAX.LE.2.40).AND.(YMAX.GT.1.20)) YSCALE=.4
              IF ((YMAX.LE.1.20).AND.(YMAX.GT.0.60)) YSCALE=.2
              IF ((YMAX.LE.0.60).AND.(YMAX.GT.0.30)) YSCALE=.1
              IF ((YMAX.LE.0.30).AND.(YMAX.GT.0.24)) YSCALE=.05
              IF ((YMAX.LE.0.24).AND.(YMAX.GT.0.12)) YSCALE=.04
              IF ((YMAX.LE.0.12).AND.(YMAX.GT.0.06)) YSCALE=.02
              IF ((YMAX.LE.0.06).AND.(YMAX.GE.0.00)) YSCALE=.01
              YMIN=-6.*YSCALE
        C
        C        DRAW AND LABEL YP, YPP, AND THETA AXIS
        C
              CALL CALPLT (0.,8.,-3)
              CALL AXES (0.,0.,0.,36.,-180.,10.,-2.,1.,10HTHETA,DEG.,SIZ,-10,0)
        C             DRAW AND LABEL YPS AXES
              CALL AXES (0.,0.,90.,12.,CMIN,CSCALE,-1.,0.,3HYPS,SIZ,3,2)
        C             DRAW AND LABEL YPPS AXES
              CALL AXES (36.,0.,90.,12.,YMIN,YSCALE,-1.,0.,4HYPPS,SIZ,-4,2)
              CALL NOTATE (1.0,11.1,.4,3,0.,-1)
              CALL NOTATE (1.5,10.9,SIZ,4HYPPS,0.,4)
              CALL NOTATE (1.0,11.7,.4,2,0.,-1)
              CALL NOTATE (1.5,11.5,SIZ,3HYPS,0.,3)
              CALL CALPLT (0.,6.,-3)
        C
        C        PLOT SMOOTHED FIRST DERIVATIVES YP VS THETA
        C
              YPSI(NM+1)=0.0
              YPSI(NM+2)=CSCALE
              CALL LINE (TI,YPSI,NM,1,0,0,0.)
        C
        C        PLOT SMOOTHED SECOND DERIVATIVES YPP VS THETA
        C
              THETA(NP+1)=-PI
              THETA(NP+2)=10./RAD
              YPPS(NP+1)=0.0
              YPPS(NP+2)=YSCALE
              CALL LINE (THETA,YPPS,NP,1,0,0,0.)
              DO 10 I=1,NP
              TP=THETA(I)*RAD/10.+18.0
              YP=YPPS(I)/YSCALE
              CALL PNTPLT (TP,YP,22,ISIZ)
```

CARD NO.

```
      10    CONTINUE
            CALL NFRAME
      C           CHECK PLOT OPTION
            IF (IPLOT.EQ.1.OR.IPLOT.EQ.6) RETURN
            IF (IPLOT.EQ.8) RETURN
      C
      C        DETERMINE SCALING FACTOR FOR Y/C AXIS
      C
      11    IF (YSAV.EQ.0.01) YMAX=8
            IF (YSAV.EQ.0.02) YMAX=12
            IF (YSAV.EQ.0.04) YMAX=20
            IF (YSAV.EQ.0.05) YMAX=24
            YMIN=-0.0125*YMAX
      C
      C        PLOT INPUT AND SMOOTHED Y/C-COORDINATES VS X/C
      C
      C        DRAW AND LABEL Y/C AND X/C AXIS
      C
            CALL CALPLT (2.,2.,-3)
            CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)
            CALL CALPLT (0.,2.,-3)
            CALL AXES (0.,0.,0.,40.,0.,.025,-2.,1.,3HX/C,SIZ,-3,2)
            CALL AXES (0.,0.,90.,YMAX,YMIN,0.025,-2.,1.,3HY/C,SIZ,3,2)
            YP=YMAX-0.9
            CALL NOTATE (1.0,YP,SIZ,2,0.,-1)
            YP=YMAX-1.1
            CALL NOTATE (1.5,YP,SIZ,8HSMOOTHED,0.,8)
            YP=YMAX-.3
            CALL NOTATE (1.0,YP,SIZ,3,0.,-1)
            YP=YMAX-.5
            CALL NOTATE (1.5,YP,SIZ,5HINPUT,0.,5)
            YP=0.5*YMAX
            CALL CALPLT (0.,YP,-3)
      C
      C        PLOT INPUT Y/C-COORDINATES
      C
            DO 14 I=1,NP
            TP=ABS(THETA(I))
            IF (TP.LE.PI2) GO TO 12
            XP=CONS*(ATAN(SINH(TP-PI2))+1.)/.025
```

CARD NO.

```
              GO TO 13
       12     XP=CONS*(1.-COS(TP))/.025
       13     YP=Y(I)/.025
              CALL PNTPLT (XP,YP,22,ISIZ)
       14     CONTINUE
      C
      C          PLOT SMOOTHED Y/C-COORDINATES
      C
              XI(NM+1)=YI(NM+1)=0.0
              XI(NM+2)=YI(NM+2)=.025
              CALL LINE (XI,YI,NM,1,0,0,0.)
              CALL NFRAME
      C
      C          RETURN TO CALLING PROGRAM
      C
              RETURN
      C
       15     FORMAT (1H1,1X,7HTITLE--,2X,8A10//49X,28H--INTERPOLATED COORDINATE
             1S--/10X,1HI,3X,5HTHETA,5X,3HX/C,7X,3HY/C,12X,1HI,3X,5HTHETA,5X,3HX
             2/C,7X,3HY/C,12X,1HI,3X,5HTHETA,5X,3HX/C,7X,3HY/C/)
       16     FORMAT (3(7X,I4,F8.2,2F10.6))
              END
```

CARD NO.

```
      SUBROUTINE PLOTCK (THETA,YSMO,YPS,YPPS,NP,TITLE)
C
C     ROUTINE TO PLOT SQUARE ROOT OF SMOOTHED CURVATURE VERSUS THETA
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION THETA(1), YSMO(1), YPS(1), YPPS(1), TITLE(8)
C
      COMMON /HLM/ TI(723)
      COMMON /SMY/ CURV(723)
      COMMON /BLK1/ PI,PI2,RAD,CONS
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
      DATA NM/721/,SIZ/.40/,ISIZ/3/
C
C     SINH(X)=0.5*(EXP(X)-EXP(-X))
C     COSH(X)=0.5*(EXP(X)+EXP(-X))
C
C        INTERPOLATE NM CURVATURE POINTS
C
      IF (IPRINT.NE.0) GO TO 1
      WRITE (JWRITE,15) TITLE
    1 DP=(THETA(NP)-THETA(1))/FLOAT(NM-1)
      TDEL=THETA(1)-DP
      M=2
      DO 8 I=1,NM
      TDEL=TDEL+DP
      IF (TDEL.LT.THETA(1)) TDEL=THETA(1)
      IF (TDEL.GT.THETA(NP)) TDEL=THETA(NP)
      TI(I)=TDEL*RAD
      TP=TDEL
      IF (M.LT.2) M=2
      DO 2 K=M,NP
      J=K-1
      IF (TP.GE.THETA(J).AND.TP.LE.THETA(K)) GO TO 3
    2 CONTINUE
    3 M=J
      DELTA=THETA(J+1)-THETA(J)
      T2=THETA(J+1)-TP
      T1=TP-THETA(J)
```

CARD NO.

```
                YI=YPPS(J)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPS(J+1)*(T1**3/(6.*DEL PC  41
               1TA)-T1*DELTA/6.)+(YSMO(J)*T2+YSMO(J+1)*T1)/DELTA                  PC  42
                YPI=YPPS(J)*(DELTA/6.-T2*T2/(2.*DELTA))+YPPS(J+1)*(T1*T1/(2.*DELTA PC  43
               1)-DELTA/6.)+(YSMO(J+1)-YSMO(J))/DELTA                             PC  44
                YPPI=(YPPS(J)*T2+YPPS(J+1)*T1)/DELTA                               PC  45
                DELTA=YPI                                                          PC  46
                IF (TP.LE.0.0) DELTA=-DELTA                                        PC  47
                TP=ABS(TP)                                                         PC  48
                IF (TP.GT.PI2) GO TO 4                                             PC  49
                GP=CONS*SIN(TP)                                                    PC  50
                GPP=CONS*COS(TP)                                                   PC  51
                XI=CONS*(1.-COS(TP))                                               PC  52
                GO TO 5                                                            PC  53
      4         T1=COSH(TP-PI2)                                                    PC  54
                T2=SINH(TP-PI2)                                                    PC  55
                XI=CONS*(ATAN(T2)+1.)                                              PC  56
                GP=CONS/T1                                                         PC  57
                GPP=-CONS*T2/(T1*T1)                                               PC  58
      5         IF (TP.LE.0.0.OR.GP.EQ.0.0) GO TO 6                                PC  59
                DYDX=DELTA/GP                                                      PC  60
                DY2DX=(YPPI*GP-DELTA*GPP)/(GP**3)                                  PC  61
                CURV(I)=ABS(DY2DX)/(SQRT(1.+DYDX**2)**3)                           PC  62
                GO TO 7                                                            PC  63
      6         DYDX=0.1E99                                                        PC  64
                DY2DX=0.1E99                                                       PC  65
                CURV(I)=CONS/(DELTA*DELTA)                                         PC  66
      7         IF (IPRINT.NE.0) GO TO 8                                           PC  67
                WRITE (JWRITE,16) I,TI(I),XI,YI,YPI,YPPI,DYDX,DY2DX,CURV(I)        PC  68
      8         CURV(I)=SQRT(CURV(I))                                              PC  69
      C                                                                            PC  70
      C            DETERMINE SCALING FACTOR FOR CURVATURE AXES                     PC  71
      C                                                                            PC  72
                CMAX=0.0                                                           PC  73
                DO 9 I=1,NM                                                        PC  74
                IF (CURV(I).GT.CMAX) CMAX=CURV(I)                                  PC  75
      9         CONTINUE                                                           PC  76
                M=IFIX(CMAX)+1                                                     PC  77
                CMAX=FLOAT(M)/20.                                                  PC  78
      C                                                                            PC  79
      C            DRAW AND LABEL CURVATURE AND THETA AXES                         PC  80
```

```
C
      CALL GRIDCK
      CALL CALPLT (2.,2.,-3)
      CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)
      CALL CALPLT (0.,2.,-3)
      CALL AXES (0.,0.,0.,36.,-180.,10.,-2.,1.,10HTHETA,DEG.,SIZ,-10,0)
      CALL AXES (0.,0.,90.,20.,0.,CMAX,-2.,1.,15HSQRT(CURVATURE),SIZ,15,
     12)
C
C         PLOT INTERPOLATED CURVATURE POINTS
C
      TI(NM+1)=-180.0
      CURV(NM+1)=0.0
      TI(NM+2)=10.
      CURV(NM+2)=CMAX
      CALL LINE (TI,CURV,NM,1,0,0,0.0)
C
C         COMPUTE AND PLOT CURVATURE AT INPUT THETA POINTS
C
      DO 14 I=1,NP
      DELTA=YPS(I)
      IF (THETA(I).LE.0.0) DELTA=-DELTA
      TP=ABS(THETA(I))
      IF (TP.GT.PI2) GO TO 10
      GP=CONS*SIN(TP)
      GPP=CONS*COS(TP)
      GO TO 11
   10 T1=COSH(TP-PI2)
      T2=SINH(TP-PI2)
      GP=CONS/T1
      GPP=-CONS*T2/(T1*T1)
   11 IF (TP.LE.0.0.OR.GP.EQ.0.0) GO TO 12
      DYDX=DELTA/GP
      DY2DX=(YPPS(I)*GP-DELTA*GPP)/(GP**3)
      T1=ABS(DY2DX)/(SQRT(1.+DYDX**2)**3)
      GO TO 13
   12 T1=CONS/(DELTA*DELTA)
   13 T2=THETA(I)*RAD/10.+18.0
      T1=SQRT(T1)/CMAX
      CALL PNTPLT (T2,T1,22,ISIZ)
```

CARD NO.

```
      14    CONTINUE                                                        PC 121
      C                                                                     PC 122
      C        ADVANCE TO NEXT FRAME AND RETURN                             PC 123
      C                                                                     PC 124
            CALL NFRAME                                                     PC 125
            RETURN                                                          PC 126
      C                                                                     PC 127
      C                                                                     PC 128
      15    FORMAT (1H1,1X,7HTITLE--,2X,8A10//36X,26H--INTERPOLATED CURVATURE- PC 129
           1-/3X,1HI,6X,5HTHETA,5X,3HX/C,7X,3HY/C,6X,5HDY/DT,5X,6HDY2/DT,7X,5H PC 130
           2DY/DX,7X,11HD(DY/DX)/DX,5X,9HCURVATURE/)                        PC 131
      16    FORMAT (I5,F10.2,4F10.6,3E15.6)                                 PC 132
            END                                                             PC 133-
```

CARD NO.

```
            SUBROUTINE CAMTK (THETA,YSMO,YPPS,NOSE,NP,EPS,KPLOT,IPUNCH,TITLE) CK   1
      C                                                                        CK   2
      C     THIS SUBROUTINE COMPUTES THE THICKNESS AND CAMBER DISTRIBUTIONS    CK   3
      C     OF THE SMOOTHED AIRFOIL                                            CK   4
      C                                                                        CK   5
      C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          CK   6
      C                                                                        CK   7
            DIMENSION TITLE(8), THETA(1), YSMO(1), YPPS(1)                     CK   8
      C                                                                        CK   9
            COMMON /SMY/ TU(100),YPPU(100),TL(100),YPPL(100),DYXU(100),LX(101) CK  10
           1,XLS(101),YLS(101),TH(101),XU(102),YU(102),XL(102),YL(102),XC(103) CK  11
           2,YC(103),TK(103)                                                   CK  12
      C                                                                        CK  13
            COMMON /BLK1/ PI,PI2,RAD,CONS                                      CK  14
      C                                                                        CK  15
            COMMON /INOUT/ JREAD,JWRITE,IPRINT                                 CK  16
      C                                                                        CK  17
            DATA NM/2001/,SIZ/.40/,ISIZ/3/                                     CK  18
      C                                                                        CK  19
      C     COSH(X)=0.5*(EXP(X)+EXP(-X))                                       CK  20
      C     SINH(X)=0.5*(EXP(X)-EXP(-X))                                       CK  21
      C                                                                        CK  22
            F(X1,X2,X3,X4,X5,X6,X7,X8,X9)=X1*(X5*X9-X6*X8)+X2*(X6*X7-X4*X9)+X3 CK  23
           1*(X4*X8-X5*X7)                                                     CK  24
      C                                                                        CK  25
      C        LOAD THETA, X/C, Y/C, AND SECOND DERIVATIVES INTO SEPARATE      CK  26
      C        ARRAYS FOR UPPER AND LOWER SURFACES                             CK  27
      C                                                                        CK  28
            J=0                                                                CK  29
            NU=NP-NOSE+1                                                       CK  30
            DO 2 I=NOSE,NP                                                     CK  31
            J=J+1                                                              CK  32
            TU(J)=THETA(I)                                                     CK  33
            YU(J)=YSMO(I)                                                      CK  34
            TP=ABS(THETA(I))                                                   CK  35
            IF (TP.GT.PI2) GO TO 1                                             CK  36
            XU(J)=CONS*(1.-COS(TP))                                            CK  37
            GO TO 2                                                            CK  38
      1     XU(J)=CONS*(ATAN(SINH(TP-PI2))+1.)                                 CK  39
      2     YPPU(J)=YPPS(I)                                                    CK  40
```

CARD NO.

```
      NL=NOSE
      J=NOSE+1
      DO 4 I=1,NOSE
      J=J-1
      TL(J)=THETA(I)
      YL(J)=YSMO(I)
      TP=ABS(THETA(I))
      IF (TP.GT.PI2) GO TO 3
      XL(J)=CONS*(1.-COS(TP))
      GO TO 4
    3 XL(J)=CONS*(ATAN(SINH(TP-PI2))+1.)
    4 YPPL(J)=YPPS(I)
C              COMPUTE FIRST DERIVATIVES OF UPPER SURFACE
      DO 5 I=2,NU
      DELTA=TU(I)-TU(I-1)
      DYXU(I)=YPPU(I)*DELTA/3.+YPPU(I-1)*DELTA/6.+(YU(I)-YU(I-1))/DELTA
      IF (TU(I).LE.PI2) DYXU(I)=DYXU(I)/(CONS*SIN(TU(I)))
      IF (TU(I).GT.PI2) DYXU(I)=DYXU(I)*COSH(TU(I)-PI2)/CONS
    5 CONTINUE
      DYXU(1)=0.1E99
C
C        COMPUTE THICKNESS AND CAMBER DISTRIBUTIONS BY FINDING LOWER
C        SURFACE COORDINATE (XLS,YLS) CORRESPONDING TO INPUT UPPER
C        SURFACE COORDINATE (XU,YU)
C
      NT=0
      KSAVE=1
      NS=1
      NL1=NL-1
      NM1=NM-1
      A1=PI/FLOAT(NM1)
      DEL=1./(FLOAT(NM1)**2)
      DO 12 I=1,NU
C              LOAD XU AND YU
      IJ=NU+1-I
      XXU=XU(IJ)
      YYU=YU(IJ)
      DYU=DYXU(IJ)
      NN=1
C              FIND XLS
```

CARD NO.

```
      DO 9 K=NS,NM
      TP=A1*FLOAT(NM-K)
      IF (K.EQ.1) TP=ABS(TL(NL))
      IF (K.EQ.NM) TP=ABS(TL(1))
      IF (TP.LE.PI2) XXL=CONS*(1.-COS(TP))
      IF (TP.GT.PI2) XXL=CONS*(ATAN(SINH(TP-PI2))+1.)
      IF (NN.EQ.NL) NN=NL1
      DO 6 J=NN,NL1
      J2=NL-J
      J1=J2+1
      IF (TP.GE.ABS(TL(J2)).AND.TP.LE.ABS(TL(J1))) GO TO 7
    6 CONTINUE
    7 DELTA=TL(J2)-TL(J1)
      T1=-TP-TL(J1)
      T2=TL(J2)+TP
      YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DE
     1LTA)-T1*DELTA/6.)+(YL(J1)*T2+YL(J2)*T1)/DELTA
      DYL=YPPL(J1)*(DELTA/6.-T2*T2/(2.*DELTA))+YPPL(J2)*(T1*T1/(2.*DELTA
     1)-DELTA/6.)+(YL(J2)-YL(J1))/DELTA
      IF (TP.LE.PI2) DELTA=CONS*SIN(TP)
      IF (TP.GT.PI2) DELTA=CONS/COSH(TP-PI2)
      IF (TP.LE.0.0) DYL=0.1E99
      IF (TP.GT.0.0) DYL=-DYL/DELTA
      NN=NL+1-J1
      D=SQRT((XXL-XXU)**2+(YYL-YYU)**2)
      IF (I.EQ.1.AND.D.LE.DEL) GO TO 10
      IF (D.LE.DEL) GO TO 9
      COST=(YYU-YYL)/D
      SINT=(XXL-XXU)/D
      IF (DYU.NE.0.1E99) DU=(COST*DYU-SINT)/(SINT*DYU+COST)
      IF (DYU.EQ.0.1E99.AND.SINT.NE.0.0) DU=COST/SINT
      IF (DYU.EQ.0.1E99.AND.SINT.EQ.0.0) DU=0.1E99
      IF (DYL.NE.0.1E99) DL=-(COST*DYL-SINT)/(SINT*DYL+COST)
      IF (DYL.EQ.0.1E99.AND.SINT.NE.0.0) DL=-COST/SINT
      IF (DYL.EQ.0.1E99.AND.SINT.EQ.0.0) DL=-0.1E99
      IF (K.EQ.NS) GO TO 8
      DKL=(DL-DLP)/(XXL-XP)
      DKU=(DU-DUP)/(XXL-XP)
      IF (DKU.EQ.DKL) GO TO 8
      XK=XP+(DLP-DUP)/(DKU-DKL)
```

CARD NO.

```
              IF (XK.LE.XP+DEL.AND.XK.GE.XXL-DEL) GO TO 11
       8      KSAVE=K
              XP=XXL
              DUP=DU
              DLP=DL
       9      CONTINUE
              IF (I.GT.1) GO TO 12
      10      XK=XL(NL)
              KSAVE=NS
      11      NT=NT+1
              LX(NT)=IJ
              XLS(NT)=XK
              NS=KSAVE
      12      CONTINUE
C                       COMPUTE YLS FOR EACH XLS AND PRINT RESULTS
              WRITE (JWRITE,44) TITLE
              DO 19 I=1,NT
              IJ=LX(I)
              DELTA=XLS(I)
              IF (DELTA.GT.1.) DELTA=1.
              IF (DELTA.LE.CONS) GO TO 13
              DELTA=TAN(DELTA/CONS-1.)
              TP=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))
              GO TO 14
      13      TP=ACOS(1.-DELTA/CONS)
      14      DO 15 J=1,NL1
              J2=NL-J
              J1=J2+1
              IF (TP.GE.ABS(TL(J2)).AND.TP.LE.ABS(TL(J1))) GO TO 16
      15      CONTINUE
      16      DELTA=TL(J2)-TL(J1)
              T1=-TP-TL(J1)
              T2=TL(J2)+TP
              YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DE
             1LTA)-T1*DELTA/6.)+(YL(J1)*T2+YL(J2)*T1)/DELTA
              YLS(I)=YYL
              XC(I)=(XU(IJ)+XLS(I))/2.
              YC(I)=(YU(IJ)+YYL)/2.
              TK(I)=0.5*SQRT((XU(IJ)-XLS(I))**2+(YU(IJ)-YYL)**2)
              IF (YU(IJ).EQ.YYL) TH(I)=0.0
```

CARD NO.

```
      IF (YU(IJ).NE.YYL) TH(I)=ATAN((XLS(I)-XU(IJ))/(YU(IJ)-YYL))        CK 161
      IF (TK(I).LE.0.0) GO TO 17                                         CK 162
      DYL=YPPL(J1)*(DELTA/6.-T2*T2/(2.*DELTA))+YPPL(J2)*(T1*T1/(2.*DELTA CK 163
     1)-DELTA/6.)+(YL(J2)-YL(J1))/DELTA                                  CK 164
      IF (TP.LE.PI2) DELTA=CONS*SIN(TP)                                  CK 165
      IF (TP.GT.PI2) DELTA=CONS/COSH(TP-PI2)                             CK 166
      IF (TP.LE.0.0) DYL=0.1E99                                          CK 167
      IF (TP.GT.0.0) DYL=-DYL/DELTA                                      CK 168
      COST=(YU(IJ)-YYL)/(2.*TK(I))                                       CK 169
      SINT=(XLS(I)-XU(IJ))/(2.*TK(I))                                    CK 170
      DU=(COST*DYXU(IJ)-SINT)/(SINT*DYXU(IJ)+COST)                       CK 171
      DL=(COST*DYL-SINT)/(SINT*DYL+COST)                                 CK 172
      T2=ABS(ABS(DU)-ABS(DL))                                            CK 173
      GO TO 18                                                           CK 174
   17 T2=0.0                                                             CK 175
   18 T1=TH(I)*RAD                                                       CK 176
      WRITE (JWRITE,45) I,XU(IJ),YU(IJ),XLS(I),YYL,XC(I),YC(I),TK(I),T1, CK 177
     1T2                                                                 CK 178
   19 CONTINUE                                                           CK 179
C                                                                        CK 180
C        COMPUTE STARTING LOCATION OF CAMBER DISTRIBUTION (I.E.          CK 181
C        THICKNESS = 0) BY FITTING SECOND ORDER CURVE TO LAST THREE      CK 182
C        COMPUTED CAMBER LINE COORDINATES AND THEN DETERMINING           CK 183
C        INTERSECTION OF THAT CURVE WITH AIRFOIL SURFACE                 CK 184
C                                                                        CK 185
      ISYM=1                                                             CK 186
      DO 20 I=1,5                                                        CK 187
      IF (ABS(XU(I)-XL(I)).GT.EPS) ISYM=0                                CK 188
      IF (ABS(YU(I)+YL(I)).GT.EPS) ISYM=0                                CK 189
   20 CONTINUE                                                           CK 190
      IF (ISYM.EQ.1) GO TO 30                                            CK 191
      IF (XC(NT).LE.DEL) GO TO 31                                        CK 192
      X1=XC(NT)**2                                                       CK 193
      X2=XC(NT-1)**2                                                     CK 194
      X3=XC(NT-2)**2                                                     CK 195
      D=F(X1,XC(NT),1.,X2,XC(NT-1),1.,X3,XC(NT-2),1.)                    CK 196
      A1=F(YC(NT),XC(NT),1.,YC(NT-1),XC(NT-1),1.,YC(NT-2),XC(NT-2),1.)/D CK 197
      A2=F(X1,YC(NT),1.,X2,YC(NT-1),1.,X3,YC(NT-2),1.)/D                 CK 198
      A3=YC(NT)-A1*X1-A2*XC(NT)                                          CK 199
      NM1=NM/4                                                           CK 200
```

CARD NO.

```
      D=XC(NT)/FLOAT(NM1)                                               CK 201
      X=0.0                                                             CK 202
      XP=X                                                              CK 203
      YYUP=YU(1)                                                        CK 204
      YYLP=YL(1)                                                        CK 205
      YYCP=(A1*X+A2)*X+A3                                               CK 206
      NM1=NM1+1                                                         CK 207
      DO 27 I=2,NM1                                                     CK 208
      X=X+D                                                             CK 209
      IF (X.GT.CONS) GO TO 27                                           CK 210
      TP=ACOS(1.-X/CONS)                                                CK 211
      DO 21 K=2,NU                                                      CK 212
      K1=K-1                                                            CK 213
      K2=K                                                              CK 214
      IF (TP.GE.TU(K1).AND.TP.LE.TU(K2)) GO TO 22                       CK 215
   21 CONTINUE                                                          CK 216
   22 DELTA=TU(K2)-TU(K1)                                               CK 217
      T1=TP-TU(K1)                                                      CK 218
      T2=TU(K2)-TP                                                      CK 219
      YYU=YPPU(K1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPU(K2)*(T1**3/(6.*DE CK 220
     1LTA)-T1*DELTA/6.)+(YU(K2)*T1+YU(K1)*T2)/DELTA                     CK 221
      DO 23 J=2,NL                                                      CK 222
      J2=J-1                                                            CK 223
      J1=J                                                              CK 224
      IF (TP.GE.ABS(TL(J2)).AND.TP.LE.ABS(TL(J1))) GO TO 24             CK 225
   23 CONTINUE                                                          CK 226
   24 DELTA=TL(J2)-TL(J1)                                               CK 227
      T1=-TP-TL(J1)                                                     CK 228
      T2=TL(J2)+TP                                                      CK 229
      YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DE CK 230
     1LTA)-T1*DELTA/6.)+(YL(J1)*T2+YL(J2)*T1)/DELTA                     CK 231
      YYC=(A1*X+A2)*X+A3                                                CK 232
      DKC=(YYC-YYCP)/(X-XP)                                             CK 233
      DKU=(YYU-YYUP)/(X-XP)                                             CK 234
      IF (DKU.EQ.DKC) GO TO 25                                          CK 235
      XKU=XP+(YYCP-YYUP)/(DKU-DKC)                                      CK 236
      IF (XKU.GE.XP.AND.XKU.LE.X) GO TO 28                              CK 237
   25 DKL=(YYL-YYLP)/(X-XP)                                             CK 238
      IF (DKL.EQ.DKC) GO TO 26                                          CK 239
      XKL=XP+(YYCP-YYLP)/(DKL-DKC)                                      CK 240
```

CARD NO.

```
            IF (XKL.GE.XP.AND.XKL.LE.X) GO TO 29                                 CK 241
   26       XP=X                                                                 CK 242
            YYLP=YYL                                                             CK 243
            YYUP=YYU                                                             CK 244
            YYCP=YYC                                                             CK 245
   27       CONTINUE                                                             CK 246
            GO TO 31                                                             CK 247
   28       NT=NT+1                                                              CK 248
            LX(NT)=0                                                             CK 249
            XLS(NT)=XKU                                                          CK 250
            XC(NT)=XKU                                                           CK 251
            DU=(A1*XKU+A2)*XKU+A3                                                CK 252
            TK(NT)=0.                                                            CK 253
            TH(NT)=ATAN(2.*A1*XKU+A2)                                            CK 254
            TP=ACOS(1.-XKU/CONS)                                                 CK 255
            DELTA=TU(K2)-TU(K1)                                                  CK 256
            T1=TP-TU(K1)                                                         CK 257
            T2=TU(K2)-TP                                                         CK 258
            YYU=YPPU(K1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPU(K2)*(T1**3/(6.*DE CK 259
           1LTA)-T1*DELTA/6.)+(YU(K2)*T1+YU(K1)*T2)/DELTA                        CK 260
            YLS(NT)=YYU                                                          CK 261
            YC(NT)=YLS(NT)                                                       CK 262
            D=ABS(ABS(DU)-ABS(YC(NT)))                                           CK 263
            T1=TH(NT)*RAD                                                        CK 264
            WRITE (JWRITE,45) NT,XLS(NT),YLS(NT),XLS(NT),YLS(NT),XC(NT),YC(NT) CK 265
           1,TK(NT),T1,D                                                         CK 266
            GO TO 31                                                             CK 267
   29       NT=NT+1                                                              CK 268
            LX(NT)=0                                                             CK 269
            XLS(NT)=XKL                                                          CK 270
            XC(NT)=XKL                                                           CK 271
            DL=(A1*XKL+A2)*XKL+A3                                                CK 272
            TK(NT)=0.                                                            CK 273
            TH(NT)=ATAN(2.*A1*XKL+A2)                                            CK 274
            TP=ACOS(1.-XKL/CONS)                                                 CK 275
            DELTA=TL(J2)-TL(J1)                                                  CK 276
            T1=-TP-TL(J1)                                                        CK 277
            T2=TL(J2)+TP                                                         CK 278
            YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DE CK 279
           1LTA)-T1*DELTA/6.)+(YL(J1)*T2+YL(J2)*T1)/DELTA                        CK 280
```

CARD NO.

```
              YLS(NT)=YYL                                                       CK 281
              YC(NT)=YLS(NT)                                                    CK 282
              D=ABS(ABS(DL)-ABS(YC(NT)))                                        CK 283
              T1=TH(NT)*RAD                                                     CK 284
              WRITE (JWRITE,45) NT,XLS(NT),YLS(NT),XLS(NT),YLS(NT),XC(NT),YC(NT) CK 285
             1,TK(NT),T1,D                                                      CK 286
              GO TO 31                                                          CK 287
       30     IF (LX(NT).EQ.1) GO TO 31                                         CK 288
              NT=NT+1                                                           CK 289
              LX(NT)=1                                                          CK 290
              XC(NT)=0.0                                                        CK 291
              YC(NT)=YU(1)                                                      CK 292
              XLS(NT)=0.0                                                       CK 293
              YLS(NT)=YL(1)                                                     CK 294
              TK(NT)=0.0                                                        CK 295
              TH(NT)=0.0                                                        CK 296
              D=0.0                                                             CK 297
              WRITE (JWRITE,45) NT,XC(NT),YC(NT),XLS(NT),YLS(NT),XC(NT),YC(NT),T CK 298
             1K(NT),TH(NT),D                                                    CK 299
       C                                                                        CK 300
       C         PUNCH CAMBER AND THICKNESS DISTRIBUTIONS                       CK 301
       C                                                                        CK 302
       31     IF (IPUNCH.NE.5) GO TO 33                                         CK 303
              WRITE (1,46) TITLE                                                CK 304
              WRITE (JWRITE,41) IPUNCH,TITLE,NT                                 CK 305
       C                                                                        CK 306
              D=FLOAT(NT)                                                       CK 307
              WRITE (1,42) D                                                    CK 308
       C                                                                        CK 309
              DO 32 I=1,NT                                                      CK 310
              J=NT+1-I                                                          CK 311
              WRITE (JWRITE,43) XC(J),YC(J),TK(J),TH(J)                         CK 312
              WRITE (1,47) XC(J),YC(J),TK(J),TH(J)                              CK 313
       32     CONTINUE                                                          CK 314
       C                                                                        CK 315
       C         PLOT CAMBER AND THICKNESS DISTRIBUTIONS                        CK 316
       C                                                                        CK 317
       33     IF (KPLOT.EQ.0) RETURN                                            CK 318
       C                 PLOT CAMBER                                            CK 319
              CALL CALPLT (4.,2.,-3)                                            CK 320
```

CARD NO.

```
              CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)                         CK 321
              CALL CALPLT (0.,2.,-3)                                      CK 322
              CALL AXES (0.,0.,0.,20.,0.,.05,-2.,1.,3HX/C,SIZ,-3,1)       CK 323
              DU=0.0                                                      CK 324
              DO 34 I=1,NT                                                CK 325
              IF (ABS(YC(I)).GT.DU) DU=ABS(YC(I))                         CK 326
        34    CONTINUE                                                    CK 327
              D=.1                                                        CK 328
              IF (DU.LE.0.2.AND.DU.GT.0.08) D=.05                         CK 329
              IF (DU.LE.0.08.AND.DU.GT.0.04) D=.02                        CK 330
              IF (DU.LE.0.04) D=.01                                       CK 331
              DL=-4.*D                                                    CK 332
              CALL AXES (0.,0.,90.,8.,DL,D,-1.,0.,3HY/C,SIZ,3,2)          CK 333
              CALL CALPLT (0.,4.,-3)                                      CK 334
              XC(NT+1)=YC(NT+1)=0.0                                       CK 335
              XC(NT+2)=.05                                                CK 336
              YC(NT+2)=D                                                  CK 337
              DO 35 I=1,NT                                                CK 338
              XU1=XC(I)/.05                                               CK 339
              YU1=YC(I)/D                                                 CK 340
              CALL PNTPLT (XU1,YU1,22,ISIZ)                               CK 341
        35    CONTINUE                                                    CK 342
              CALL LINE (XC,YC,NT,1,0,0,0.)                               CK 343
        C               PLOT THICKNESS                                    CK 344
              CALL CALPLT (0.,6.,-3)                                      CK 345
              CALL AXES (0.,0.,0.,20.,0.,.05,-2.,1.,3HX/C,SIZ,-3,1)       CK 346
              DU=0.0                                                      CK 347
              DO 36 I=1,NT                                                CK 348
              IF (ABS(TK(I)).GT.DU) DU=ABS(TK(I))                         CK 349
        36    CONTINUE                                                    CK 350
              D=.1                                                        CK 351
              IF (DU.LE.0.06) D=.01                                       CK 352
              IF (DU.GT.0.06.AND.DU.LE.0.12) D=.02                        CK 353
              IF (DU.GT.0.12.AND.DU.LE.0.24) D=.04                        CK 354
              IF (DU.GT.0.24.AND.DU.LE.0.30) D=.05                        CK 355
              CALL AXES (0.,0.,90.,6.,0.,D,-1.,0.,5HT/C/2,SIZ,5,2)        CK 356
              TK(NT+1)=0.0                                                CK 357
              TK(NT+2)=D                                                  CK 358
              DO 37 I=1,NT                                                CK 359
              XU1=XC(I)/.05                                               CK 360
```

CARD NO.

```
            YU1=TK(I)/D                                                 CK 361
            CALL PNTPLT (XU1,YU1,22,ISIZ)                               CK 362
   37       CONTINUE                                                    CK 363
            CALL LINE (XC,TK,NT,1,0,0,0.)                               CK 364
   C                PLOT INPUT AIRFOIL AND AIRFOIL GENERATED BY COMBINING   CK 365
   C                THICKNESS AND CAMBER DISTRIBUTIONS                  CK 366
            CALL CALPLT (0.,8.,-3)                                      CK 367
            CALL AXES (0.,0.,0.,20.,0.,.05,-2.,1.,3HX/C,SIZ,-3,1)       CK 368
            CALL AXES (0.,0.,90.,8.,-.2,.05,-2.,1.,3HY/C,SIZ,3,1)       CK 369
            CALL CALPLT (0.,4.,-3)                                      CK 370
            XU(NU+1)=YU(NU+1)=0.0                                       CK 371
            XU(NU+2)=YU(NU+2)=.05                                       CK 372
            CALL LINE (XU,YU,NU,1,0,0,0.)                               CK 373
            XL(NL+1)=YL(NL+1)=0.0                                       CK 374
            XL(NL+2)=YL(NL+2)=.05                                       CK 375
            CALL LINE (XL,YL,NL,1,0,0,0.)                               CK 376
            DO 40 I=1,NT                                                CK 377
            IJ=LX(I)                                                    CK 378
            IF (IJ.EQ.0) GO TO 38                                       CK 379
            XU1=XU(IJ)/.05                                              CK 380
            YU1=YU(IJ)/.05                                              CK 381
            XL1=XLS(I)/.05                                              CK 382
            YL1=YLS(I)/.05                                              CK 383
            GO TO 39                                                    CK 384
   38       XU1=XL1=XLS(I)/.05                                          CK 385
            YU1=YL1=YLS(I)/.05                                          CK 386
   39       CONTINUE                                                    CK 387
            CALL PNTPLT (XU1,YU1,22,ISIZ)                               CK 388
            CALL PNTPLT (XL1,YL1,22,ISIZ)                               CK 389
   40       CONTINUE                                                    CK 390
            CALL NFRAME                                                 CK 391
            RETURN                                                      CK 392
   41       FORMAT (1H1,5X,47HTHE FOLLOWING CAMBERLINE DATA HAVE BEEN PUNCHED, CK 393
           15X,7HIPUNCH=,I4//5X,8A10//5X,4HNT =,I4//9X,3HX/C,7X,3HY/C,5X,5HT/C CK 394
           2/2,5X,5HSLOPE)                                              CK 395
   42       FORMAT (F10.2)                                              CK 396
   43       FORMAT (5X,4F10.6)                                          CK 397
   44       FORMAT (1H1,1X,7HTITLE--,2X,8A10//32X,37H--THICKNESS AND CAMBER DI CK 398
           1STRIBUTION--//4X,1HI,5X,4HXU/C,6X,4HYU/C,6X,4HXL/C,6X,4HYL/C,6X,3H CK 399
           2X/C,7X,3HY/C,6X,5HT/C/2,5X,5HSLOPE,10X,5HERROR/)            CK 400
```

```
 401      45    FORMAT (I5,7F10.6,F10.4,5X,F10.6)                CK 401
          46    FORMAT (8A10)                                    CK 402
          47    FORMAT (4F10.6)                                  CK 403
                END                                              CK 404-
```

CARD NO.

```
      SUBROUTINE INTP (THETA,X,YSMO,YPPS,NP,NOSE,CHORD,TITLE,NINT,XINT,C
     1NEW,INTR,IPUNCH)
C
C     ROUTINE TO INTERPOLATE ADDITIONAL UPPER AND LOWER SURFACE
C     COORDINATES
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION TITLE(8), THETA(1), X(1), YSMO(1), YPPS(1), XINT(1)
C
      DIMENSION XSAV(57)
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /HLM/ XU(100),YU(100),XL(100),YL(100),TLS(100)
C
C          STANDARD X/C COORDINATE INTERPOLATION VALUES
      DATA (XSAV(I),I=1,57)/0.0,.00025,.0005,.00075,.001,.0015,.002,.002
     15,.005,.01,.02,.03,.04,.05,.06,.07,.08,.09,.1,.125,.15,.175,.2,.22
     25,.25,.275,.3,.325,.35,.375,.4,.425,.45,.475,.5,.525,.55,.575,.6,.
     3625,.65,.675,.7,.725,.75,.775,.8,.825,.85,.875,.9,.925,.95,.97,.98
     4,.99,1.0/
C
C     IF INTR EQUAL 1, LOAD STANDARD X/C COORDINATE VALUES
C
      IF (INTR.EQ.0) RETURN
      IF (INTR.EQ.2) GO TO 2
      NINT=57
      DO 1 I=1,NINT
    1 XINT(I)=XSAV(I)
C
C     INTERPOLATE UPPER SURFACE COORDINATES
C
    2 WRITE (JWRITE,7) TITLE
      XUP=X(NP)*CHORD
      XNOSE=X(NOSE)*CHORD
      XLO=X(1)*CHORD
      RATIO=CNEW/CHORD
```

CARD NO.

```
      DO 5 I=1,NINT
      XU(I)=XINT(I)*CHORD+XNOSE
      XL(I)=XU(I)
      IF (XU(I).GT.XUP) XU(I)=XUP
      IF (XL(I).GT.XLO) XL(I)=XLO
      XU(I)=(XU(I)-XNOSE)*RATIO
      XL(I)=(XL(I)-XNOSE)*RATIO
      DELTA=XINT(I)
      IF (DELTA.LE.CONS) GO TO 3
      DELTA=TAN(DELTA/CONS-1.)
      TU=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))
      GO TO 4
    3 TU=ACOS(1.-DELTA/CONS)
    4 TL=-TU
      IF (TL.LT.THETA(1)) TL=THETA(1)
      IF (TU.GT.THETA(NP)) TU=THETA(NP)
      TLS(I)=TL
      CALL COORD (THETA,YPPS,YSMO,NP,TU,YU(I),DYDX,DY2DX,CURV)
      YU(I)=YU(I)*CNEW
      WRITE (JWRITE,8) I,XU(I),YU(I),DYDX,DY2DX,CURV
    5 CONTINUE
      WRITE (JWRITE,9) CNEW
C
C     INTERPOLATE LOWER SURFACE COORDINATES
C
      WRITE (JWRITE,10) TITLE
      DO 6 I=1,NINT
      TL=TLS(I)
      CALL COORD (THETA,YPPS,YSMO,NP,TL,YL(I),DYDX,DY2DX,CURV)
      YL(I)=YL(I)*CNEW
      WRITE (JWRITE,8) I,XL(I),YL(I),DYDX,DY2DX,CURV
    6 CONTINUE
C
C     PUNCH COORDINATES
C
      IF (IPUNCH.NE.6) RETURN
      WRITE (JWRITE,11) CNEW,TITLE
      WRITE (1,12) TITLE
      WRITE (JWRITE,13) NINT
      XNT=FLOAT(NINT)
```

```
      WRITE (1,14) XNT
      WRITE (JWRITE,15) (XU(I),I=1,NINT)
      WRITE (JWRITE,16) (YU(I),I=1,NINT)
      WRITE (1,17) (XU(I),YU(I),I=1,NINT)
      WRITE (JWRITE,18) NINT
      WRITE (1,14) XNT
      WRITE (JWRITE,19) (XL(I),I=1,NINT)
      WRITE (JWRITE,20) (YL(I),I=1,NINT)
      WRITE (1,17) (XL(I),YL(I),I=1,NINT)
C
C     RETURN TO CALLING PROGRAM
C
      RETURN
C
    7 FORMAT (1H1,5X,9HTITLE--  ,8A10//26X,42H--UPPER SURFACE INTERPOLAT
     1ED COORDINATES--//9X,1HI,10X,2HXU,13X,2HYU,11X,5HDY/DX,6X,11HD(DY/
     2DX)/DX,6X,9HCURVATURE)
    8 FORMAT (I10,2F15.6,3E15.6)
    9 FORMAT (/10X,7HCHORD =,F10.6)
   10 FORMAT (1H1,5X,9HTITLE--  ,8A10//26X,42H--LOWER SURFACE INTERPOLAT
     1ED COORDINATES--//9X,1HI,10X,2HXL,13X,2HYL,11X,5HDY/DX,6X,11HD(DY/
     2DX)/DX,6X,9HCURVATURE)
   11 FORMAT (1H1,10X,50HTHE FOLLOWING DATA HAVE BEEN PUNCHED FOR A CHOR
     1D =,F10.6//3X,9HTITLE--  ,8A10)
   12 FORMAT (8A10)
   13 FORMAT (5X,4HNU =,I4)
   14 FORMAT (F10.2)
   15 FORMAT (5X,4HXU =,8F10.6/(9X,8F10.6))
   16 FORMAT (5X,4HYU =,8F10.6/(9X,8F10.6))
   17 FORMAT (2F10.6)
   18 FORMAT (5X,4HNL =,I4)
   19 FORMAT (5X,4HXL =,8F10.6/(9X,8F10.6))
   20 FORMAT (5X,4HYL =,8F10.6/(9X,8F10.6))
      END
```

CARD NO.

```
      SUBROUTINE COORD (THETA,YPPS,YSMO,NP,TI,YI,DYDX,DY2DX,CURV)
C
C     ROUTINE TO COMPUTE THE Y COORDINATE, DY/DX, D(DY/DX)/DX, AND
C     CURVATURE AT A GIVEN VALUE OF THETA
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION THETA(1), YPPS(1), YSMO(1)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
C     COSH(X)=(EXP(X)+EXP(-X))/2.
C     SINH(X)=(EXP(X)-EXP(-X))/2.
C
      DO 1 K=2,NP
      J=K-1
      IF (TI.GE.THETA(J).AND.TI.LE.THETA(K)) GO TO 2
    1 CONTINUE
    2 DELTA=THETA(J+1)-THETA(J)
      T2=THETA(J+1)-TI
      T1=TI-THETA(J)
      YI=YPPS(J)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPS(J+1)*(T1**3/(6.*DEL
     1TA)-T1*DELTA/6.)+(YSMO(J)*T2+YSMO(J+1)*T1)/DELTA
      YPI=YPPS(J)*(DELTA/6.-T2*T2/(2.*DELTA))+YPPS(J+1)*(T1*T1/(2.*DELTA
     1)-DELTA/6.)+(YSMO(J+1)-YSMO(J))/DELTA
      YPPI=(YPPS(J)*T2+YPPS(J+1)*T1)/DELTA
      DELTA=YPI
      IF (TI.LE.0.0) DELTA=-DELTA
      TP=ABS(TI)
      IF (TP.GT.PI2) GO TO 3
      GP=CONS*SIN(TP)
      GPP=CONS*COS(TP)
      GO TO 4
    3 T1=COSH(TP-PI2)
      T2=SINH(TP-PI2)
      GP=CONS/T1
      GPP=-CONS*T2/(T1*T1)
    4 IF (TP.LE.0.0.OR.GP.EQ.0.0) GO TO 5
      DYDX=DELTA/GP
      DY2DX=(YPPI*GP-DELTA*GPP)/(GP**3)
```

CARD NO.

```
              CURV=ABS(DY2DX)/(SQRT(1.+DYDX**2)**3)                CD  41
              RETURN                                               CD  42
        5     DYDX=0.1E99                                          CD  43
              DY2DX=0.1E99                                         CD  44
              CURV=CONS/(DELTA*DELTA)                              CD  45
              RETURN                                               CD  46
              END                                                  CD  47-
```

LISTING OF DECK: SINH

CARD NO.

```
      FUNCTION SINH(X)                                                  SH    1
C     HYPERBOLIC SINE                                                   SH    2
      SINH=0.5*(EXP(X)-EXP(-X))                                         SH    3
      RETURN                                                            SH    4
      END                                                               SH    5-
```

CARD NO.

```
                  FUNCTION COSH(X)                                      CH   1
        C         HYPERBOLIC COSINE                                     CH   2
                  COSH=0.5*(EXP(X)+EXP(-X))                             CH   3
                  RETURN                                                CH   4
                  END                                                   CH   5-
```

APPENDIX B

COMPUTER LISTING OF AIRFOIL SCALING PROGRAM AFSCL

This appendix contains a computer listing of the airfoil scaling program AFSCL which consists of a main program and two subroutines.

CARD NO.

```
      PROGRAM SCALE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)         SC   1
C                                                                       SC   2
C     THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF  SC   3
C     AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED    SC   4
C     OUTPUT MAXIMUM THICKNESS RATIO                                    SC   5
C                                                                       SC   6
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982         SC   7
C                                                                       SC   8
C*********************************************************************** SC   9
C*                                                                    * SC  10
C*               DESCRIPTION OF INPUT CARDS FOR SCALING PROGRAM       * SC  11
C*                                                                    * SC  12
C*.....................................................................* SC  13
C*                                                                    * SC  14
C* CARD NUMBER                  DESCRIPTION                           * SC  15
C*                                                                    * SC  16
C*.....................................................................* SC  17
C*   1                                 FORMAT(8A10)                   * SC  18
C*              TITLE CARD                                            * SC  19
C*.....................................................................* SC  20
C*   2                                 FORMAT(4F10.0)                 * SC  21
C*              NT - NUMBER OF INPUT CAMBER, THICKNESS, AND SLOPE     * SC  22
C*                   POINTS                                           * SC  23
C*              IPLOT - PLOT OPTION                                   * SC  24
C*                      0 - NO PLOTS DESIRED                          * SC  25
C*                      1 - PLOTS DESIRED                             * SC  26
C*              IPUNCH - PUNCH OUTPUT OPTION                          * SC  27
C*                       0 - NO PUNCHED OUTPUT DESIRED                * SC  28
C*                       1 - PUNCH COORDINATES OF SCALED AIRFOIL      * SC  29
C*              IOP - SLOPE OF CAMBERLINE OPTION                      * SC  30
C*                    0 - SLOPES INPUT ON CARD 3                      * SC  31
C*                    1 - SLOPES COMPUTED BY PROGRAM                  * SC  32
C*.....................................................................* SC  33
C*   3                                 FORMAT(4F10.0)                 * SC  34
C*              XC - X/C COORDINATES OF CAMBERLINE                    * SC  35
C*              YC - Y/C COORDINATES OF CAMBERLINE                    * SC  36
C*              TK - T/C/2 THICKNESS DISTRIBUTION                     * SC  37
C*              TH - SLOPE OF CAMBERLINE IN RADIANS                   * SC  38
C*                   NOTE -- CARD 3 IS READ NT TIMES                  * SC  39
C*.....................................................................* SC  40
```

CARD NO.

```
C*   4                           FORMAT(F10.0)                          * SC  41
C*            LT - NUMBER OF DESIRED OUTPUT MAXIMUM THICKNESS RATIOS    * SC  42
C*......................................................................* SC  43
C*   5                           FORMAT(F10.0)                          * SC  44
C*            TKNEW - DESIRED OUTPUT MAXIMUM THICKNESS RATIO            * SC  45
C*                 NOTE -- CARD 5 IS READ LT TIMES                      * SC  46
C*......................................................................* SC  47
C*                                                                      * SC  48
C*       RESTRICTIONS:                                                  * SC  49
C*              NT NOT GREATER THAN 101                                 * SC  50
C*              LT NOT GREATER THAN 10                                  * SC  51
C*              XC MUST BE MONOTONICALLY INCREASING                     * SC  52
C*                                                                      * SC  53
C*********************************************************************** SC  54
C                                                                         SC  55
      DIMENSION XC(101), YC(101), TK(101), TH(101), THETA(101), YPP(101)  SC  56
     1, TKNEW(10), TITLE(8), VAR(4)                                       SC  57
C                                                                         SC  58
      COMMON /HLM/ WK(404,3)                                              SC  59
C                                                                         SC  60
      COMMON /BLK1/ PI,PI2,RAD,CONS                                       SC  61
C                                                                         SC  62
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                  SC  63
C                                                                         SC  64
      SINH(X)=0.5*(EXP(X)-EXP(-X))                                        SC  65
C                                                                         SC  66
C        INITIALIZE PROGRAM CONSTANTS                                     SC  67
C                                                                         SC  68
      JWRITE=6                                                            SC  69
      JREAD=5                                                             SC  70
      IPRINT=0                                                            SC  71
      NTMAX=101                                                           SC  72
      PI=ACOS(-1.)                                                        SC  73
      PI2=PI/2.                                                           SC  74
      RAD=180./PI                                                         SC  75
      CONS=1./(1.+ATAN(SINH(PI2)))                                        SC  76
C                                                                         SC  77
C        READ AND PRINT INPUT DATA                                        SC  78
C                                                                         SC  79
1     READ (JREAD,26) TITLE                                               SC  80
```

CARD NO.

```
            IF (EOF(JREAD)) 25,2
      2     READ (JREAD,27) VAR
            NT=IFIX(VAR(1))
            IF (NT.GT.NTMAX) GO TO 24
            IPLOT=IFIX(VAR(2))
            IF (IPLOT.NE.0) IPLOT=1
            IPUNCH=IFIX(VAR(3))
            IF (IPUNCH.NE.0) IPUNCH=1
            IOP=IFIX(VAR(4))
            IF (IOP.NE.0) IOP=1
            WRITE (JWRITE,28) TITLE,NT,IPLOT,IPUNCH,IOP
            READ (JREAD,29) (XC(I),YC(I),TK(I),TH(I),I=1,NT)
            WRITE (JWRITE,30) (XC(I),I=1,NT)
            WRITE (JWRITE,31) (YC(I),I=1,NT)
            WRITE (JWRITE,32) (TK(I),I=1,NT)
            IF (IOP.EQ.0) WRITE (JWRITE,33) (TH(I),I=1,NT)
            READ (JREAD,34) VAR(1)
            LT=IFIX(VAR(1))
            IF (LT.LE.0) GO TO 1
            IF (LT.GT.10) LT=10
            READ (JREAD,34) (TKNEW(I),I=1,LT)
            WRITE (JWRITE,35) LT,(TKNEW(I),I=1,LT)
      C
      C        INITIALIZE PLOTTING DEVICE
      C
            CALL PSEUDO
            CALL LEROY
      C
      C        CHECK FOR INCREASING XC
      C
            DO 3 I=2,NT
            IF (XC(I).LE.XC(I-1)) GO TO 4
      3     CONTINUE
            GO TO 5
      4     WRITE (JWRITE,36)
            GO TO 1
      C
      C        FIND MAXIMUM THICKNESS RATIO OF INPUT AIRFOIL
      C
      C              COMPUTE THETA EQUIVALENT OF XC
```

CARD NO.

```
      5     CHORD=XC(NT)-XC(1)                                               SC 121
            DO 7 I=1,NT                                                      SC 122
            DELTA=(XC(I)-XC(1))/CHORD                                        SC 123
            IF (DELTA.LE.CONS) GO TO 6                                       SC 124
            DELTA=TAN(DELTA/CONS-1.)                                         SC 125
            THETA(I)=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))                    SC 126
            GO TO 7                                                          SC 127
      6     THETA(I)=ACOS(1.-DELTA/CONS)                                     SC 128
      7     CONTINUE                                                         SC 129
      C              FIT CUBIC SPLINE THRU TK VS THETA                       SC 130
            CALL CUBSPL (THETA,TK,YPP,NT,WK)                                 SC 131
      C              FIND LOCATIONS WHERE D(TK)/D(THETA) = 0.0               SC 132
            KRT=0                                                            SC 133
            N1=NT-1                                                          SC 134
            DO 12 I=1,N1                                                     SC 135
            DELTA=THETA(I+1)-THETA(I)                                        SC 136
            AA=(YPP(I)-YPP(I+1))/(2.*DELTA)                                  SC 137
            BB=(YPP(I+1)*THETA(I)-YPP(I)*THETA(I+1))/DELTA                   SC 138
            CC=(YPP(I)*THETA(I+1)**2-YPP(I+1)*THETA(I)**2)/(2.*DELTA)+(YPP(I+1 SC 139
           1)-YPP(I))*DELTA/6.-(TK(I+1)-TK(I))/DELTA                         SC 140
            GP=BB*BB-4.*AA*CC                                                SC 141
            IF (GP) 12,8,8                                                   SC 142
      8     GP=SQRT(GP)                                                      SC 143
            T1=(-BB+GP)/(2.*AA)                                              SC 144
            T2=(-BB-GP)/(2.*AA)                                              SC 145
            IF (T1.GE.THETA(I).AND.T1.LE.THETA(I+1)) GO TO 9                 SC 146
            GO TO 10                                                         SC 147
      9     KRT=KRT+1                                                        SC 148
            WK(KRT,1)=T1                                                     SC 149
      10    IF (T2.GE.THETA(I).AND.T2.LE.THETA(I+1)) GO TO 11                SC 150
            GO TO 12                                                         SC 151
      11    KRT=KRT+1                                                        SC 152
            WK(KRT,1)=T2                                                     SC 153
      12    CONTINUE                                                         SC 154
            IF (KRT.EQ.0) GO TO 16                                           SC 155
      C              COMPUTE XC LOCATIONS WHERE D(TK)/D(THETA) = 0.0         SC 156
            DO 15 I=1,KRT                                                    SC 157
            T1=ABS(WK(I,1))                                                  SC 158
            IF (T1.LE.PI2) WK(I,2)=CONS*(1.-COS(T1))                         SC 159
            IF (T1.GT.PI2) WK(I,2)=CONS*(ATAN(SINH(T1-PI2))+1.)              SC 160
```

CARD NO.

```
              DO 13 J=1,N1
              J1=J
              J2=J+1
              IF (WK(I,1).GE.THETA(J).AND.WK(I,1).LE.THETA(J+1)) GO TO 14
     13       CONTINUE
     14       AA=THETA(J2)-WK(I,1)
              BB=WK(I,1)-THETA(J1)
              DELTA=THETA(J2)-THETA(J1)
     15       WK(I,3)=YPP(J1)*(AA**3/(6.*DELTA)-AA*DELTA/6.)+YPP(J2)*(BB**3/(6.*
             1DELTA)-BB*DELTA/6.)+(TK(J1)*AA+TK(J2)*BB)/DELTA
     16       CONTINUE
     C                COMPUTE AND PRINT MAXIMUM THICKNESS RATIO
              IF (KRT.EQ.0) GO TO 23
              TKMAX=0.0
              DO 18 I=1,KRT
              IF (WK(I,3).GE.TKMAX) GO TO 17
              GO TO 18
     17       N1=I
              TKMAX=WK(I,3)
     18       CONTINUE
              TKMAX=2.*TKMAX
              DELTA=WK(N1,2)*CHORD+XC(1)
              WRITE (JWRITE,37) TKMAX,DELTA
              IF (TKMAX.LE.0.0) GO TO 1
     C
     C           IF IOP=1, COMPUTE SLOPES OF CAMBERLINE
     C
              IF (IOP.NE.1) GO TO 21
              CALL CUBSPL (XC,YC,YPP,NT,WK)
              DO 20 I=1,NT
              IF (I.EQ.NT) GO TO 19
              DELTA=XC(I+1)-XC(I)
              TH(I)=-YPP(I)*DELTA/3.-YPP(I+1)*DELTA/6.+(YC(I+1)-YC(I))/DELTA
              GO TO 20
     19       DELTA=XC(NT)-XC(NT-1)
              TH(I)=YPP(NT-1)*DELTA/6.+YPP(NT)*DELTA/3.+(YC(NT)-YC(NT-1))/DELTA
     20       TH(I)=ATAN(TH(I))
     C
     C           COMPUTE AND PRINT COORDINATES OF INPUT AIRFOIL
     C
```

CARD NO.

```
21    CALL SCTK (XC,YC,TK,TH,NT,TITLE,TKMAX,TKMAX,IPUNCH,IPLOT,IERR)
      IF (IERR.NE.0) GO TO 1
C
C        COMPUTE AND PRINT COORDINATES OF SCALED AIRFOILS
C
      DO 22 I=1,LT
      CALL SCTK (XC,YC,TK,TH,NT,TITLE,TKNEW(I),TKMAX,IPUNCH,IPLOT,IERR)
      IF (IERR.NE.0) GO TO 1
22    CONTINUE
C
C        READ NEXT CASE
C
      GO TO 1
C
C        PRINT ERROR MESSAGE
C
23    WRITE (JWRITE,38)
      GO TO 1
24    WRITE (JWRITE,39) NTMAX
      GO TO 1
C
C        FINALIZE PLOTTING DEVICE
C
25    CALL CALPLT (0.,0.,999)
      STOP
C
26    FORMAT (8A10)
27    FORMAT (4F10.6)
28    FORMAT (1H1,57X,14H--INPUT DATA--//5X,7HTITLE--,2X,8A10//5X,3HNT=,
     1I3,5X,6HIPLOT=,I3,5X,7HIPUNCH=,I3,5X,4HIOP=,I3)
29    FORMAT (4F10.6)
30    FORMAT (/4X,4HX/C=,8E15.6/(8X,8E15.6))
31    FORMAT (/4X,4HY/C=,8E15.6/(8X,8E15.6))
32    FORMAT (/2X,6HT/C/2=,8E15.6/(8X,8E15.6))
33    FORMAT (/2X,6HSLOPE=,8E15.6/(8X,8E15.6))
34    FORMAT (F10.2)
35    FORMAT (/2X,3HLT=,I3,5X,9HNEW T/C =,10F12.6)
36    FORMAT (/5X,40HXC ARRAY IS NOT MONOTONICALLY INCREASING)
37    FORMAT (/5X,28H(T/C)MAX FOR INPUT AIRFOIL =,F10.6,2X,8HAT X/C =,
     1F10.6)
```

CARD NO.

```
241        38    FORMAT (//5X,64H(T/C)MAX OF INPUT AIRFOIL WAS NOT FOUND -- CHECK Y SC 241
                1OUR INPUT DATA)                                                     SC 242
           39    FORMAT (//5X,35HINPUT CARD ERROR - NT GREATER THAN ,I4)              SC 243
                 END                                                                 SC 244-
```

CARD NO.

```
      SUBROUTINE SCTK (XC,YC,TK,TH,NT,TITLE,TKNEW,TKMAX,IPUNCH,IPLOT,IER
     1R)
C
C     THIS SUBROUTINE SCALES THE COORDINATES OF AN AIRFOIL FROM A BASIC
C     MAXIMUM THICKNESS RATIO (TKMAX) TO A NEW MAXIMUM THICKNESS RATIO
C     (TKNEW)
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION XC(1), YC(1), TK(1), TH(1), TITLE(8)
C
      COMMON /HLM/ X(220),Y(220),XU(110),YU(110),XL(110),YL(110),XPRT(11
     10),YPRT(110),TPRT(110)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
C        SCALE THICKNESS AND COMPUTE UPPER AND LOWER SURFACE COORDINATES
C        OF NEW AIRFOIL
C
      IERR=0
      DELT1=TKNEW/TKMAX
      DO 1 I=1,NT
      DELT2=COS(TH(I))
      DELT4=SIN(TH(I))
      XU(I)=XC(I)-TK(I)*DELT4*DELT1
      YU(I)=YC(I)+TK(I)*DELT2*DELT1
      XL(I)=XC(I)+TK(I)*DELT4*DELT1
    1 YL(I)=YC(I)-TK(I)*DELT2*DELT1
C
C        LOAD SURFACE COORDINATES INTO X AND Y ARRAYS
C
      DO 2 I=1,NT
      J=NT+1-I
      X(I)=XL(J)
    2 Y(I)=YL(J)
      N=NT
      M=1
      IF (XU(1).EQ.XL(1).AND.YU(1).EQ.YL(1)) M=2
```

CARD NO.

```
          DO 3 I=M,NT
          N=N+1
          X(N)=XU(I)
    3     Y(N)=YU(I)
C
C            INTERPOLATE OR EXTRAPOLATE TRAILING EDGE COORDINATES
C
          IF (X(1)-X(N)) 4,6,5
    4     DELT1=X(2)-X(1)
          DELT2=X(3)-X(1)
          DELT3=Y(2)-Y(1)
          DELT4=Y(3)-Y(1)
          Y(1)=Y(1)+(X(N)-X(1))*((DELT3*DELT2-DELT4*DELT1)*(X(N)-X(1))+(DELT
         14*DELT1*DELT1-DELT3*DELT2*DELT2))/(DELT2*DELT1*DELT1-DELT1*DELT2*D
         2ELT2)
          X(1)=X(N)
          GO TO 6
    5     DELT1=X(N-1)-X(N-2)
          DELT2=X(N)-X(N-2)
          DELT3=Y(N-1)-Y(N-2)
          DELT4=Y(N)-Y(N-2)
          Y(N)=Y(N-2)+(X(1)-X(N-2))*((DELT3*DELT2-DELT4*DELT1)*(X(1)-X(N-2))
         1+(DELT4*DELT1*DELT1-DELT3*DELT2*DELT2))/(DELT2*DELT1*DELT1-DELT1*D
         2ELT2*DELT2)
          X(N)=X(1)
C
C            COMPUTE LONGEST CHORD
C
    6     CHORD=0.0
          DO 8 I=2,N
          DELT=X(1)-X(I)
          IF (DELT.GT.CHORD) GO TO 7
          GO TO 8
    7     CHORD=DELT
          NOSE=I
    8     CONTINUE
C
C            ADJUST COORDINATES FOR LONGEST CHORD
C
          DELT=X(NOSE)
```

CARD NO.

```
              DO 9 I=1,N
              X(I)=(X(I)-DELT)/CHORD
        9     Y(I)=Y(I)/CHORD
        C
        C        CHECK UPPER AND LOWER SURFACE X VALUES TO DETECT CROSSOVER OF
        C        PERPENDICULARS TO CAMBERLINE AND TO FIND NOSE POINT
        C
              DO 10 I=2,NOSE
              IF (X(I)-X(I-1)) 10,20,20
        10    CONTINUE
              J=NOSE+1
              DO 11 I=J,N
              IF (X(I)-X(I-1)) 20,20,11
        11    CONTINUE
        C
        C        LOAD COORDINATES INTO UPPER AND LOWER SURFACE ARRAYS
        C
              DO 12 I=1,NOSE
              J=NOSE+1-I
              XL(I)=X(J)
        12    YL(I)=Y(J)
              DO 13 I=NOSE,N
              J=I+1-NOSE
              XU(J)=X(I)
        13    YU(J)=Y(I)
              NL=NOSE
              NU=N-NOSE+1
        C
        C        PRINT SCALED SURFACE COORDINATES
        C
              WRITE (JWRITE,21) TITLE,TKNEW
              J=NU
              IF (NL.GT.NU) J=NL
              DO 14 I=1,J
              IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,22) I,XU(I),YU(I),XL(I),YL(
             1I)
              IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,22) I,XU(I),YU(I)
              IF (I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,23) I,XL(I),YL(I)
        14    CONTINUE
        C
```

CARD NO.

```
      C         PRINT CAMBER AND THICKNESS DISTRIBUTIONS                     SK 121
      C                                                                      SK 122
            WRITE (JWRITE,24) TITLE,TKNEW                                    SK 123
            DELT4=TKNEW/TKMAX                                                SK 124
            DO 15 I=1,NT                                                     SK 125
            XPRT(I)=(XC(I)-DELT)/CHORD                                       SK 126
            YPRT(I)=YC(I)/CHORD                                              SK 127
            TPRT(I)=TK(I)*DELT4                                              SK 128
            DELT3=2.0*TPRT(I)                                                SK 129
            DELT1=TH(I)*RAD                                                  SK 130
      15    WRITE (JWRITE,25) I,XPRT(I),YPRT(I),DELT1,DELT3                  SK 131
      C                                                                      SK 132
      C         PUNCH DESIRED OUTPUT DATA                                    SK 133
      C                                                                      SK 134
            IF (IPUNCH.EQ.0) GO TO 16                                        SK 135
            WRITE (JWRITE,26) (TITLE(I),I=1,6),TKNEW                         SK 136
            WRITE (1,27) (TITLE(I),I=1,6),TKNEW                              SK 137
            WRITE (JWRITE,28) NU                                             SK 138
            DELT1=FLOAT(NU)                                                  SK 139
            WRITE (1,29) DELT1                                               SK 140
            WRITE (JWRITE,30) (XU(I),I=1,NU)                                 SK 141
            WRITE (JWRITE,31) (YU(I),I=1,NU)                                 SK 142
            WRITE (1,32) (XU(I),YU(I),I=1,NU)                                SK 143
            WRITE (JWRITE,33) NL                                             SK 144
            DELT1=FLOAT(NL)                                                  SK 145
            WRITE (1,29) DELT1                                               SK 146
            WRITE (JWRITE,34) (XL(I),I=1,NL)                                 SK 147
            WRITE (JWRITE,35) (YL(I),I=1,NL)                                 SK 148
            WRITE (1,32) (XL(I),YL(I),I=1,NL)                                SK 149
      16    IF (IPLOT.EQ.0) RETURN                                           SK 150
      C                                                                      SK 151
      C         PLOT AIRFOIL SHAPE AND CAMBER AND THICKNESS DISTRIBUTIONS    SK 152
      C                                                                      SK 153
      C                LABEL PLOT                                            SK 154
            CALL CALPLT (2.,0.,-3)                                           SK 155
            CALL NOTATE (0.,0.,.40,44HPLOT OF AIRFOIL GENERATED BY SCALING PRO SK 156
           1GRAM,0.,44)                                                      SK 157
            CALL NOTATE (16.0,0.,.40,10H(T/C)MAX =,0.,10)                    SK 158
            CALL NUMBER (20.0,0.,.40,TKNEW,0.0,3)                            SK 159
            CALL NOTATE (0.,1.,.40,TITLE,0.,80)                              SK 160
```

CARD NO.

```
      C           PLOT AIRFOIL                                          SK 161
            CALL AXES (0.,4.,0.,20.,0.0,.05,-2.,1.,3HX/C,.40,-3,1)      SK 162
            CALL AXES (0.,4.,90.,8.,-.2,.05,-2.,1.,3HY/C,.40,3,1)       SK 163
            CALL CALPLT (0.,8.,-3)                                      SK 164
            X(N+1)=Y(N+1)=0.0                                           SK 165
            X(N+2)=Y(N+2)=.05                                           SK 166
            CALL LINE (X,Y,N,1,0,0,0.0)                                 SK 167
      C           PLOT CAMBER DISTRIBUTION                              SK 168
            CALL CALPLT (0.,6.,-3)                                      SK 169
            CALL AXES (0.,0.,0.,20.,0.,.05,-2.,1.,3HX/C,.40,-3,1)       SK 170
            DELT1=0.0                                                   SK 171
            DO 17 I=1,NT                                                SK 172
            IF (ABS(YPRT(I)).GT.DELT1) DELT1=ABS(YPRT(I))               SK 173
      17    CONTINUE                                                    SK 174
            DELT2=.1                                                    SK 175
            IF (DELT1.LE.0.2.AND.DELT1.GT.0.08) DELT2=.05               SK 176
            IF (DELT1.LE.0.08.AND.DELT1.GT.0.04) DELT2=.02              SK 177
            IF (DELT1.LE.0.04) DELT2=.01                                SK 178
            DELT1=-4.*DELT2                                             SK 179
            CALL AXES (0.,0.,90.,8.,DELT1,DELT2,-1.,0.,3HY/C,.40,3,2)   SK 180
            CALL CALPLT (0.,4.,-3)                                      SK 181
            XPRT(NT+1)=YPRT(NT+1)=0.0                                   SK 182
            XPRT(NT+2)=.05                                              SK 183
            YPRT(NT+2)=DELT2                                            SK 184
            DO 18 I=1,NT                                                SK 185
            DELT3=XPRT(I)/.05                                           SK 186
            DELT4=YPRT(I)/DELT2                                         SK 187
            CALL PNTPLT (DELT3,DELT4,22,3)                              SK 188
      18    CONTINUE                                                    SK 189
            CALL LINE (XPRT,YPRT,NT,1,0,0,0.)                           SK 190
      C           PLOT THICKNESS DISTRIBUTION                           SK 191
            CALL CALPLT (0.,6.,-3)                                      SK 192
            CALL AXES (0.,0.,0.,20.,0.,.05,-2.,1.,3HX/C,.40,-3,1)       SK 193
            CALL AXES (0.,0.,90.,7.,0.,.02,-1.,0.,5HT/C/2,.40,5,2)      SK 194
            TPRT(NT+1)=0.0                                              SK 195
            TPRT(NT+2)=.02                                              SK 196
            DO 19 I=1,NT                                                SK 197
            DELT3=XPRT(I)/.05                                           SK 198
            DELT4=TPRT(I)/.02                                           SK 199
            CALL PNTPLT (DELT3,DELT4,22,3)                              SK 200
```

CARD NO.

```
 19      CONTINUE
         CALL LINE (XPRT,TPRT,NT,1,0,0,0.0)
         CALL NFRAME
         RETURN
C
C           PRINT ERROR MESSAGE
C
 20      WRITE (JWRITE,36) TITLE,TKNEW
         WRITE (JWRITE,37) (I,X(I),Y(I),I=1,N)
         IERR=1
         RETURN
C
 21      FORMAT (1H1,3X,7HTITLE--,2X,8A10//12X,33HSCALED COORDINATES FOR (T
        1/C)MAX =,F7.4//24X,5HUPPER,20X,5HLOWER//9X,1HI,10X,3HX/C,7X,3HY/C,
        212X,3HX/C,7X,3HY/C)
 22      FORMAT (5X,I5,5X,2F10.6,5X,2F10.6)
 23      FORMAT (5X,I5,30X,2F10.6)
 24      FORMAT (1H1,3X,7HTITLE--,2X,8A10//12X,49HCAMBER AND THICKNESS DIST
        1RIBUTIONS FOR (T/C)MAX =,F7.4//34X,6HCAMBER,22X,9HTHICKNESS//9X,1H
        2I,10X,3HX/C,12X,3HY/C,11X,5HSLOPE,10X,5HT/C/2)
 25      FORMAT (5X,I5,2(5X,F10.6),5X,F10.4,5X,F10.6)
 26      FORMAT (1H1,10X,36HTHE FOLLOWING DATA HAVE BEEN PUNCHED//5X,9HTITL
        1E--  ,6A10,10H(T/C)MAX =,F10.6)
 27      FORMAT (6A10,10H(T/C)MAX =,F10.6)
 28      FORMAT (/5X,4HNU =,I4)
 29      FORMAT (F10.2)
 30      FORMAT (/5X,4HXU =,8F10.6/(9X,8F10.6))
 31      FORMAT (/5X,4HYU =,8F10.6/(9X,8F10.6))
 32      FORMAT (2F10.6)
 33      FORMAT (/5X,4HNL =,I4)
 34      FORMAT (/5X,4HXL =,8F10.6/(9X,8F10.6))
 35      FORMAT (/5X,4HYL =,8F10.6/(9X,8F10.6))
 36      FORMAT (1H1,3X,7HTITLE--,2X,8A10//3X,38HATTEMPT TO SCALE AIRFOIL T
        1O (T/C)MAX =,F7.4,2X,55HFAILED DUE TO CROSSOVER OF PERPENDICULARS
        2TO CAMBERLINE//9X,1HI,9X,3HX/C,13X,3HY/C)
 37      FORMAT (5X,I5,5X,F10.6,5X,F10.6)
         END
```

CARD NO.

```
      SUBROUTINE CUBSPL (X,Y,YPP,N,A)                                   CB   1
C                                                                       CB   2
C     THIS SUBROUTINE FITS A CUBIC SPLINE TO A SET OF Y VS X INPUT      CB   3
C     POINTS                                                            CB   4
C                                                                       CB   5
C        CODED BY -- HARRY MORGAN      NASA/LARC/TAD/AAB      1982      CB   6
C                                                                       CB   7
C        IN CALLING PROGRAM DIMENSION X, Y, AND YPP BY N AND A BY 2*N   CB   8
C                                                                       CB   9
      DIMENSION X(1), Y(1), YPP(1), A(N,2)                              CB  10
C                                                                       CB  11
C     COMPUTE SECOND DERIVATIVE AT END POINTS BY FITTING                CB  12
C     Y=A*X**2+B*X+C TO THE LAST THREE POINTS AND SOLVE FOR A.          CB  13
C     SECOND DERIVATIVE AT END POINT IS THEN EQUAL TO 2.*A              CB  14
C                                                                       CB  15
      H1=X(2)-X(3)                                                      CB  16
      H2=X(3)-X(1)                                                      CB  17
      H3=X(1)-X(2)                                                      CB  18
      YPP(1)=2.*(Y(1)*H1+Y(2)*H2+Y(3)*H3)/(H1*X(1)**2+H2*X(2)**2+H3*X(3) CB  19
     1**2)                                                              CB  20
      H1=X(N-1)-X(N)                                                    CB  21
      H2=X(N)-X(N-2)                                                    CB  22
      H3=X(N-2)-X(N-1)                                                  CB  23
      YPP(N)=2.*(Y(N-2)*H1+Y(N-1)*H2+Y(N)*H3)/(H1*X(N-2)**2+H2*X(N-1)**2 CB  24
     1+H3*X(N)**2)                                                      CB  25
C                                                                       CB  26
C     PERFORM FORWARD ELIMINATION                                       CB  27
C                                                                       CB  28
      A(1,1)=0.0                                                        CB  29
      A(1,2)=YPP(1)                                                     CB  30
      N1=N-1                                                            CB  31
      DO 1 I=2,N1                                                       CB  32
      H1=X(I)-X(I-1)                                                    CB  33
      H2=X(I+1)-X(I)                                                    CB  34
      H3=(Y(I+1)-Y(I))/H2-(Y(I)-Y(I-1))/H1                              CB  35
      D=H1*(2.-A(I-1,1))+2.*H2                                          CB  36
      A(I,1)=H2/D                                                       CB  37
    1 A(I,2)=(6.*H3-H1*A(I-1,2))/D                                      CB  38
C                                                                       CB  39
C     PERFORM BACK SUBSTITUTION                                         CB  40
```

CARD NO.

```
      C                                                                  CB  41
            J=N                                                          CB  42
            DO 2 I=2,N1                                                  CB  43
            J=J-1                                                        CB  44
      2     YPP(J)=A(J,2)-A(J,1)*YPP(J+1)                                CB  45
      C                                                                  CB  46
      C     RETURN TO CALLING PROGRAM                                    CB  47
      C                                                                  CB  48
            RETURN                                                       CB  49
            END                                                          CB  50-
```

APPENDIX C

DESCRIPTION OF INPUT FOR AIRFOIL SMOOTHING PROGRAM AFSMO

This appendix contains a description of the input requirements for the airfoil smoothing program AFSMO. All variables are input with a card format of 8F10.0, except the title card which has a format of 8A10.

CARD	VARIABLE	VALUE	DESCRIPTION
1	TITLE	-	80-column title
2	ITER	-	Maximum number of smoothing iteratives
2	IPLOT	0	No plots desired
		1	Plot smoothed and unsmoothed $\bar{y}$ and smoothed $\bar{y}'$ and $\bar{y}''$ versus θ
		2	Plot smoothed and unsmoothed $\bar{y}$ versus $\bar{x}$
		3	Plot smoothed curvature versus θ
		4	Plot camber and thickness distribution versus $\bar{x}$ (ICAMTK must equal 1)
		5	Plot combined options 1 and 2
		6	Plot combined options 1 and 3
		7	Plot combined options 1, 2, and 3
		8	Plot combined options 1 and 4
		9	Plot combined options 1, 2, and 4
		10	Plot combined options 1, 2, 3, and 4
2	IPUNCH	0	No punched output desired
		1	Punch smoothed x, y, and w
		2	Punch smoothed θ, $\bar{y}$, and w
		3	Punch smoothed θ, $\bar{y}'$, and w (YLTE, YNOSE, YUTE also punched)
		4	Punch smoothed θ, $\bar{y}''$, and w (YLTE, YNOSE, YUTE also punched)
		5	Punch x_c, y_c, t/c/2, and ϕ of camber and thickness distribution (ICAMTK must equal 1)
		6	Punch interpolated x and y coordinates (INTR must equal 1 or 2)
2	IOP	0	Upper and lower surface x, y, and w input

CARD	VARIABLE	VALUE	DESCRIPTION
		1	Upper and lower surface θ, $\bar{y}$, and w input
		2	Upper and lower surface θ, $\bar{y}'$, and w input
		3	Upper and lower surface θ, $\bar{y}''$, and w input
2	ICAMTK	0	Do not compute camber and thickness distribution
		1	Compute camber and thickness distribution
2	IBAD	0	Do not check for bad input coordinates
		1	Check for bad input coordinates
2	ITRN	0	Do not translate and rotate input coordinates
		1	Translate and rotate input coordinates so that x-axis corresponds to longest chordline
2	INTR	0	No coordinate interpolation desired
		1	Interpolate smoothed $\bar{y}$ coordinates for standard set of 57 $\bar{x}$ coordinates defined in subroutine INTP
		2	Interpolate smoothed $\bar{y}$ coordinates at input $\bar{x}$ coordinates (must specify NINT, XINT, and CNEW quantities)
3	NU	-	Number of input upper surface points
4	XU, YU, WU	0	Upper surface x, y, and w
		1	Upper surface θ, $\bar{y}$, and w
		2	Upper surface θ, $\bar{y}'$, and w
		3	Upper surface θ, $\bar{y}''$, and w (card 4 must be input NU times and x or θ runs from nose to trailing edge)
5	NL	-	Number of input lower surface points
6	XL, YL, WL	0	Lower surface x, y, and w
		1	Lower surface θ, $\bar{y}$, and w
		2	Lower surface θ, $\bar{y}'$, and w
		3	Lower surface θ, $\bar{y}''$, and w (card 6 must be input NL times and x or θ runs from nose to trailing edge)

CARD	VARIABLE	VALUE	DESCRIPTION
7	YLTE, YNOSE, YUTE	-	Lower surface trailing edge, nose, and upper surface trailing-edge $\bar{y}$ coordinates (Skip this card if IOP=0 or 1)
8	NINT	-	Number of desired interpolation $\bar{x}$ coordinates (Skip this card if INTR = 0 or 1)
9	XINT	-	Interpolation $\bar{x}$ coordinates (must be input NINT times with 8 values per card, but skip if INTR = 0 or 1)
10	CNEW	-	Desired chord length of interpolated $\bar{x}$ and $\bar{y}$ coordinates. (must be greater than zero, but skip if INTR = 0 or 1)

The primary restrictions on the input data are that the input value of the variables ITER not exceed 300 and the values of NU, NL and NINT not exceed 100. If the user desires to input a weighting value of 1.0 for any input point, the WU and WL columns may be left blank. The variables WU and WL are checked in subroutine INPUT to determine if the weighting value is less than 1.0 and, if so, a value of 1.0 is substituted. The coordinates and derivatives for the upper and lower surfaces must be input from the nose to the trailing edge for each surface and must be in monotonically increasing order.

APPENDIX D

DESCRIPTION OF OUTPUT FOR AIRFOIL SMOOTHING PROGRAM AFSMO

This appendix contains a description of the output for the airfoil smoothing program AFSMO. Presented in table II is the sample 12-page output for the smoothing program utilizing the plot, punch, camber and thickness, bad-point search, translation and rotation, and interpolation options.

A summary of the input data is printed on page 1 and all of the quantities printed are described in Appendix C. If the IBAD option is exercised and bad coordinates are found, the bad points and the corresponding replacement values will be printed on page 2. The allowable deviation (TOLR) and the surface identifier are printed at the top of page 2. If the ITRN option is exercised, pages 3 and 4 will be printed. Page 3 contains a listing of the input prior to translation and rotation and page 4 contains a listing after translation and rotation. On each page the upper surface coordinates are listed on the left and lower surface listed on the right. The coordinates of the leading edge of the longest chord (XNOSE and YNOSE) in the input axis-system and the angle (ANGLE) between the longest chord and the input x-axis are printed at the bottom of page 4. A summary of the input nondimensionalized $\bar{x}$ and $\bar{y}$ coordinates (X/C and Y/C), θ-transformation values (THETA), and weighting factors (W) are printed on page 5. All data are printed in the reordered format from the lower surface trailing-edge point clockwise around the airfoil to the upper surface trailing-edge point. If the IOP parameter equals 2, the input first derivative $\bar{y}'$ (YPS) will be printed instead of the $\bar{y}$ coordinate and, likewise if the IOP equals 3, the input second derivative $\bar{y}''$ (YPPS) will be printed. The value of the computed chord (CHORD) is printed at the bottom of page 5.

A summary of the results from the iterative smoothing process is printed on page 6. The sum-of-squares differences generated during the iterative least-squares polynomial smoothing process are printed initially. The differences are printed 10 to a line with iteration 1 to 10 on line 1, 11 to 20 on line 2, 21 to 30 on line 3, and so on. Immediately following the printout of the differences, a message is printed that states whether the smoothing process converged either within a specified number of iterations or tolerance, or began to oscillate during the smoothing process. The next message printed is the sum-of-squares difference for the least-squares cubic-spline smoothing process and should always be equal to the number of coordinates (NP) times the square of the allowable deviation (DF). The last line printed on page 6 is the result of the iteration procedure in subroutine YNEW to match the upper and lower surface slopes at the nose. The magnitude listed for DELTA is the incremental value added to all of the smoothed second derivative values.

A summary of the smoothed airfoil properties are printed on page 7. The quantities listed under the THETA, X/C, and Y/C headings are the θ-transformation values and the input $\bar{x}$ and $\bar{y}$ coordinates, respectively. The quantities listed under the YT/C heading are the partially smoothed $\bar{y}$ coordinates generated during the least-squares polynomial smoothing process and under the YSMO/C heading the final smoothed values following the solution of the cubic-spline matrix. The quantity listed under the DELTA heading are the differences between the input and final smoothed $\bar{y}$ coordinates (Y/C - YSMO/C). The quantities listed under the YPS, YPPS, DY/DX, D(DY/DX)/DX and CURVATURE headings are $\bar{y}'$, $\bar{y}''$, dy/dx, d^2y/dx^2, and k, respectively. The value of the leading-edge radius is printed next and is simply the reciprocal of the curvature at the nose. The locations of the upper and lower surface inflection points are printed at the bottom of page 7. A summary of the check of the final smoothed $\bar{y}$ and $\bar{y}''$ values is printed on page 8. The check values are obtained by making a call to the least-squares polynomial smoothing subroutine LSQSMO input with the final smoothed $\bar{y}$ coordinates and a uniform weighting factor of 1.0.

A summary of the desired punched data is printed on page 9. The upper surface quantities are listed first and then the lower surface quantities. The values listed adjacent to the DX heading are the x coordinates if IPUNCH equals 1 and the θ-values if IPUNCH is greater than 1. The values adjacent to the DY heading are y, $\bar{y}$, $\bar{y}'$, or $\bar{y}''$ if IPUNCH equals 1, 2, 3, or 4, respectively.

A summary of the camber and thickness distribution data is printed on page 10. The quantities listed under the XU/C and YU/C headings are the smoothed upper surface $\bar{x}$ and $\bar{y}$ coordinates input during the search for the camberline. The quantities listed under the XL/C and YL/C headings are the corresponding lower surface points located during the search. The quantities listed under the X/C, Y/C, T/C/2, and SLOPE headings are the x_c and y_c coordinates of the camberline, the local half thickness-chord ratio t/c/2, and the local slope of the camberline ϕ, respectively. The quantity listed under the ERROR heading are the absolute values of the difference between the local slopes of the upper and lower surface coordinates with respect to the local camberline-axis system.

The results of the interpolation process are printed on pages 11 and 12 for the upper and lower surfaces, respectively. The x and y coordinate values are listed under the XU and YU or XL and YL headings and are based on a chord equal to the value of the input parameter CNEW. The quantities listed under the DY/DX, D(DY/DX)/DX, and CURVATURE headings are dy/dx, d^2y/dx^2, and k, respectively.

APPENDIX E

DESCRIPTION OF INPUT FOR AIRFOIL SCALING PROGRAM AFSCL

This appendix contains a description of the input requirements for the airfoil scaling program AFSCL. All variables are input with a card format of 8F10.0, except the title card which has a format of 8A10.

CARD	VARIABLE	VALUE	DESCRIPTION
1	TITLE	-	80-column title
2	NT	-	Number of input thickness and camber points
2	IPLOT	0 1	No plots desired Plot scaled airfoil and its thickness and camber distributions
2	IPUNCH	0 1	No punched output desired Punch $\bar{x}$ and $\bar{y}$ coordinates of scaled airfoil
2	IOP	0 1	Slopes of camberline ϕ (TH array) are input on card 3 Slopes of camberline to be computed by scaling program
3	XC, YC, TK, TH	-	y_c coordinates of camberline (YC), the half thickness distribution t/c/2 (TK), and slope of camberline ϕ (TH) versus x_c coordinate (XC). (Card 3 is input NT times)
4	LT	-	Number of scaled maximum thickness-chord ratios
5	TKNEW	-	Scaled maximum thickness-chord ratios (Card 5 is input LT times)

The input data restrictions are that the variable NT not exceed 101, the variable LT not exceed 10, and that the coordinates for the camberline and thickness distribution be input in a monotonically increasing order from nose to trailing edge.

APPENDIX F

DESCRIPTION OF OUTPUT FOR AIRFOIL SCALING PROGRAM AFSCL

This appendix contains a description of the output for the airfoil scaling program AFSCL. Presented in table III is a sample 3-page output for the scaling program. A summary of the input data is printed on page 1. A description of the input parameters is presented in Appendix E. The guantities listed adjacent to the X/C, Y/C, and SLOPE headings are the x_c and y_c coordinates and local slopes ϕ (XC, YC, and TH arrays) of the camberline and adjacent to the T/C/2 heading are the half thickness distribution values t/c/2 (TK array). The values listed adjacent to the heading NEW T/C are the desired scaled maximum thickness-chord ratios (TKNEW array). The value of the maximum thickness-chord ratio for the input airfoil and its $\bar{x}$ coordinate are printed on the last line of page 1.

Page 2 and 3 are then output for the input airfoil and each airfoil for a desired scaled maximum thickness-chord ratio. A summary of the upper and lower surface $\bar{x}$ and $\bar{y}$ coordinates of the scaled airfoil is presented on page 2 and the corresponding camber and thickness distributions on page 3. The slopes of the camberline in degrees are also printed on page 3.

REFERENCES

1. Abbott, Ira H.; and VonDoenhoff, Albert E.: Theory of Wing Sections. Doven Publications, Inc., New York, 1959.
2. Garabedian, P.R.: Transonic Airfoil Codes. NASA CP-2045, Advanced Technology Airfoil Research, Volume I, Part 1, 1979.
3. Stevens, W. A.; Goradia, S. H.; and Braden, J.A.: Mathematical Model for Two-Dimensional Multi-Component Airfoils in Viscous Flow. NASA CR-1843, 1971.
4. Eppler, Richard; and Somers, Dan M.: A Computer Program for the Design and Analysis of Low-Speed Airfoils. NASA TM 80210, 1980.
5. Reinsch, Christian H.: "Smoothing by Spline Functions" Numerische Mathematik, Vol. 10, no. 3, 1967, pp. 177-183.
6. Hildebrand, Francis B.: Methods of Applied Mathematics. Second Edition, Prentice-Hall, Inc., 1965.

TABLE I. - TRANSFORMATION FUNCTION AND FIRST AND SECOND DERIVATIVES

θ, deg (+/-)	x/c	$d(x/c)/d\theta$	$d^2(x/c)/d\theta^2$
0	0.00000	0.00000	.46278
1	.00007	.00808	.46270
2	.00028	.01615	.46249
3	.00063	.02422	.46214
4	.00113	.03228	.46165
5	.00176	.04033	.46101
6	.00254	.04837	.46024
7	.00345	.05640	.45933
8	.00450	.06441	.45827
9	.00570	.07239	.45708
10	.00703	.08036	.45574
11	.00850	.08830	.45427
12	.01011	.09622	.45266
13	.01186	.10410	.45091
14	.01375	.11196	.44903
15	.01577	.11978	.44701
16	.01793	.12756	.44485
17	.02022	.13530	.44255
18	.02265	.14301	.44013
19	.02521	.15066	.43756
20	.02791	.15828	.43487
21	.03074	.16584	.43204
22	.03370	.17336	.42908
23	.03679	.18082	.42599
24	.04001	.18823	.42277
25	.04336	.19558	.41942
26	.04684	.20287	.41594
27	.05044	.21010	.41234
28	.05417	.21726	.40861
29	.05802	.22436	.40475
30	.06200	.23139	.40078
31	.06610	.23835	.39668
32	.07032	.24523	.39246
33	.07466	.25205	.38812
34	.07912	.25878	.38366
35	.08369	.26544	.37908
36	.08838	.27201	.37439
37	.09319	.27851	.36959
38	.09810	.28491	.36467
39	.10313	.29123	.35964
40	.10827	.29747	.35451
41	.11351	.30361	.34926
42	.11887	.30966	.34391
43	.12432	.31561	.33845
44	.12988	.32147	.33289
45	.13554	.32723	.32723

TABLE I. - CONTINUED

θ, deg (+/-)	x/c	d(x/c)/dθ	$d^2(x/c)/d\theta^2$
46	.14130	.33289	.32147
47	.14716	.33845	.31561
48	.15312	.34391	.30966
49	.15917	.34926	.30361
50	.16531	.35451	.29747
51	.17154	.35964	.29123
52	.17786	.36467	.28491
53	.18427	.36959	.27851
54	.19076	.37439	.27201
55	.19734	.37908	.26544
56	.20399	.38366	.25878
57	.21073	.38812	.25205
58	.21754	.39246	.24523
59	.22443	.39668	.23835
60	.23139	.40078	.23139
61	.23842	.40475	.22436
62	.24552	.40861	.21726
63	.25268	.41234	.21010
64	.25991	.41594	.20287
65	.26720	.41942	.19558
66	.27455	.42277	.18823
67	.28195	.42599	.18082
68	.28942	.42908	.17336
69	.29693	.43204	.16584
70	.30450	.43487	.15828
71	.31211	.43756	.15066
72	.31977	.44013	.14301
73	.32747	.44255	.13530
74	.33522	.44485	.12756
75	.34300	.44701	.11978
76	.35082	.44903	.11196
77	.35867	.45091	.10410
78	.36656	.45266	.09622
79	.37447	.45427	.08830
80	.38242	.45574	.08036
81	.39038	.45708	.07239
82	.39837	.45827	.06441
83	.40638	.45933	.05640
84	.41440	.46024	.04837
85	.42244	.46101	.04033
86	.43049	.46165	.03228
87	.43856	.46214	.02422
88	.44662	.46249	.01615
89	.45470	.46270	.00808
90	.46278	.46278	.00000

TABLE I. - CONTINUED

θ, deg (+/-)	x/c	d(x/c)/dθ	$d^2(x/c)/d\theta^2$
91	.47085	.46270	-.00807
92	.47893	.46249	-.01614
93	.48700	.46214	-.02418
94	.49506	.46165	-.03218
95	.50311	.46102	-.04013
96	.51115	.46025	-.04802
97	.51917	.45934	-.05584
98	.52718	.45830	-.06358
99	.53517	.45712	-.07122
100	.54314	.45582	-.07876
101	.55108	.45438	-.08618
102	.55900	.45281	-.09347
103	.56689	.45111	-.10063
104	.57474	.44930	-.10765
105	.58257	.44736	-.11451
106	.59036	.44530	-.12122
107	.59811	.44313	-.12775
108	.60583	.44084	-.13411
109	.61350	.43845	-.14029
110	.62113	.43595	-.14628
111	.62872	.43334	-.15208
112	.63626	.43064	-.15768
113	.64375	.42784	-.16308
114	.65119	.42495	-.16827
115	.65858	.42197	-.17326
116	.66592	.41890	-.17803
117	.67320	.41575	-.18260
118	.68043	.41253	-.18695
119	.68760	.40923	-.19108
120	.69472	.40586	-.19500
121	.70177	.40242	-.19871
122	.70876	.39892	-.20220
123	.71569	.39537	-.20548
124	.72256	.39175	-.20855
125	.72937	.38809	-.21140
126	.73611	.38437	-.21406
127	.74279	.38062	-.21650
128	.74940	.37682	-.21874
129	.75594	.37298	-.22079
130	.76241	.36911	-.22264
131	.76882	.36521	-.22430
132	.77516	.36128	-.22577
133	.78143	.35733	-.22706
134	.78764	.35336	-.22818
135	.79377	.34937	-.22911

TABLE I. - CONCLUDED

θ, deg (+/-)	x/c	d(x/c)/dθ	$d^2(x/c)/d\theta^2$
136	.79983	.34536	-.22988
137	.80582	.34134	-.23049
138	.81175	.33732	-.23093
139	.81760	.33328	-.23123
140	.82338	.32925	-.23137
141	.82909	.32521	-.23137
142	.83473	.32117	-.23123
143	.84030	.31714	-.23096
144	.84580	.31311	-.23056
145	.85123	.30909	-.23004
146	.85659	.30508	-.22940
147	.86188	.30108	-.22865
148	.86710	.29710	-.22779
149	.87225	.29313	-.22683
150	.87733	.28918	-.22577
151	.88235	.28525	-.22462
152	.88729	.28134	-.22338
153	.89217	.27746	-.22206
154	.89698	.27359	-.22066
155	.90172	.26975	-.21919
156	.90639	.26594	-.21764
157	.91100	.26216	-.21604
158	.91554	.25840	-.21437
159	.92002	.25467	-.21264
160	.92443	.25098	-.21086
161	.92878	.24731	-.20904
162	.93307	.24368	-.20716
163	.93729	.24008	-.20525
164	.94145	.23652	-.20329
165	.94554	.23299	-.20131
166	.94958	.22949	-.19928
167	.95356	.22603	-.19724
168	.95747	.22261	-.19516
169	.96133	.21922	-.19306
170	.96512	.21587	-.19094
171	.96886	.21255	-.18881
172	.97254	.20928	-.18666
173	.97617	.20604	-.18449
174	.97973	.20284	-.18231
175	.98325	.19967	-.18013
176	.98670	.19655	-.17794
177	.99011	.19346	-.17575
178	.99346	.19041	-.17355
179	.99676	.18740	-.17135
180	1.00000	.18443	-.16915

TABLE II.- SAMPLE OUTPUT FOR AIRFOIL SMOOTHING PROGRAM

PAGE 1 OUTPUT

```
                                        --INPUT DATA--

   TITLE--  //    GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS                                 //

   ITER = 100   IPLOT = 10   IPUNCH =  1   IOP =  0   ICAMTK =  1   IBAD =  1   ITRN =  1   INTR =  2

   NU =  40

   XU=   0.             .200000E-02   .500000E-02   .125000E-01   .250000E-01   .375000E-01   .500000E-01   .750000E-01
          .100000E+00   .125000E+00   .150000E+00   .175000E+00   .200000E+00   .250000E+00   .300000E+00   .350000E+00
          .400000E+00   .450000E+00   .500000E+00   .550000E+00   .575000E+00   .600000E+00   .625000E+00   .650000E+00
          .675000E+00   .700000E+00   .725000E+00   .750000E+00   .775000E+00   .800000E+00   .825000E+00   .850000E+00
          .875000E+00   .900000E+00   .925000E+00   .950000E+00   .975000E+00   .990000E+00   .995000E+00   .100000E+01

   YU=   0.             .130000E-01   .204000E-01   .307000E-01   .617000E-01   .496500E-01   .558900E-01   .655100E-01
          .730000E-01   .790000E-01   .840000E-01   .884000E+00   .920000E-01   .977000E-01   .101600E+00   .104000E+00
          .104910E+00   .104450E+00   .102580E+00   .991000E-01   .966800E-01   .937100E-01   .900600E-01   .859900E-01
          .813600E-01   .763400E-01   .709200E-01   .651300E-01   .590700E-01   .528600E-01   .464600E-01   .398800E-01
          .331500E-01   .263900E-01   .196100E-01   .128700E-01   .609000E-02   .200000E-02   .700000E-03  -.700000E-03

   WU=    .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01

   NL =  40

   XL=   0.             .200000E-02   .500000E-02   .125000E-01   .250000E-01   .375000E-01   .500000E-01   .750000E-01
          .100000E+00   .125000E+00   .150000E+00   .175000E+00   .200000E+00   .250000E+00   .300000E+00   .350000E+00
          .400000E+00   .450000E+00   .500000E+00   .550000E+00   .575000E+00   .600000E+00   .625000E+00   .650000E+00
          .675000E+00   .700000E+00   .725000E+00   .750000E+00   .775000E+00   .800000E+00   .825000E+00   .850000E+00
          .875000E+00   .900000E+00   .925000E+00   .950000E+00   .975000E+00   .990000E+00   .995000E+00   .100000E+01

   YL=   0.            -.930000E-02  -.138000E-01  -.205000E-01  -.269000E-01  -.319000E-01  -.358000E-01  -.421000E-01
         -.470000E-01  -.510000E-01  -.543000E-01  -.570000E-01  -.593000E-01  -.627000E-01  -.645000E-01  -.652000E-01
         -.649000E-01  -.635000E-01  -.610000E-01  -.570000E-01  -.540000E-01  -.508000E-01  -.469000E-01  -.428000E-01
         -.384000E-01  -.340000E-01  -.294000E-01  -.249000E-01  -.204000E-01  -.160000E-01  -.120000E-01  -.860000E-02
         -.580000E-02  -.360000E-02  -.250000E-02  -.260000E-02  -.400000E-02  -.570000E-02  -.670000E-02  -.800000E-02

   WL=    .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01
          .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01   .100000E+01

 NINT =  17

 XINT=   0.             .100000E-02   .200000E-02   .500000E-02   .100000E-01   .500000E-01   .800000E-01   .100000E+00
          .200000E+00   .300000E+00   .400000E+00   .500000E+00   .600000E+00   .700000E+00   .800000E+00   .900000E+00
          .100000E+01

 CNEW =    10.000
```

PAGE 2 OUTPUT

```
WARNING -- BAD POINTS HAVE BEEN FOUND ON THE UPPER SURFACE BASED ON AN EDIT TOLERANCE OF   .010000

BAD POINT AT I=  12     X =    .175000       Y =    .884000      REPLACED WITH Y =     .088310

BAD POINT AT I=   5     X =    .025000       Y =    .061700      REPLACED WITH Y =     .041720
```

TABLE II. - CONTINUED

PAGE 3 OUTPUT

--INPUT COORDINATES--

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS

I	XU	YU	XL	YL
1	0.000000	0.000000	0.000000	0.000000
2	.002000	.013000	.002000	-.009300
3	.005000	.020400	.005000	-.013800
4	.012500	.030700	.012500	-.020500
5	.025000	.041720	.025000	-.026900
6	.037500	.049650	.037500	-.031900
7	.050000	.055890	.050000	-.035800
8	.075000	.065510	.075000	-.042100
9	.100000	.073000	.100000	-.047000
10	.125000	.079000	.125000	-.051000
11	.150000	.084000	.150000	-.054300
12	.175000	.088310	.175000	-.057000
13	.200000	.092000	.200000	-.059300
14	.250000	.097700	.250000	-.062700
15	.300000	.101600	.300000	-.064500
16	.350000	.104000	.350000	-.065200
17	.400000	.104910	.400000	-.064900
18	.450000	.104450	.450000	-.063500
19	.500000	.102580	.500000	-.061000
20	.550000	.099100	.550000	-.057000
21	.575000	.096680	.575000	-.054000
22	.600000	.093710	.600000	-.050800
23	.625000	.090060	.625000	-.046900
24	.650000	.085990	.650000	-.042800
25	.675000	.081360	.675000	-.038400
26	.700000	.076340	.700000	-.034000
27	.725000	.070920	.725000	-.029400
28	.750000	.065130	.750000	-.024900
29	.775000	.059070	.775000	-.020400
30	.800000	.052860	.800000	-.016000
31	.825000	.046460	.825000	-.012000
32	.850000	.039880	.850000	-.008600
33	.875000	.033150	.875000	-.005800
34	.900000	.026390	.900000	-.003600
35	.925000	.019610	.925000	-.002500
36	.950000	.012870	.950000	-.002600
37	.975000	.006090	.975000	-.004000
38	.990000	.002000	.990000	-.005700
39	.995000	.000700	.995000	-.006700
40	1.000000	-.000700	1.000000	-.008000

PAGE 4 OUTPUT

--TRANSLATED AND ROTATED COORDINATES--

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS

I	XU	YU	XL	YL
1	0.000000	0.000000	0.000000	0.000000
2	.001943	.013009	.002040	-.009291
3	.004911	.020422	.005060	-.013778
4	.012366	.030754	.012589	-.020445
5	.024818	.041828	.025117	-.026791
6	.037284	.049813	.037638	-.031737
7	.049756	.056107	.050155	-.035582
8	.074714	.065836	.075182	-.041773
9	.099682	.073434	.100204	-.046565
10	.124655	.079543	.125221	-.050456
11	.149633	.084652	.150235	-.053647
12	.174614	.089070	.175246	-.056238
13	.199598	.092869	.200256	-.058429
14	.249573	.098787	.250270	-.061612
15	.299555	.102904	.300278	-.063194
16	.349544	.105522	.350280	-.063677
17	.399540	.106649	.400279	-.063159
18	.449541	.106406	.450272	-.061542
19	.499549	.104754	.500261	-.058824
20	.549564	.101492	.550243	-.054607
21	.574574	.099180	.575229	-.051498
22	.599587	.096319	.600215	-.048190
23	.624602	.092778	.625198	-.044181
24	.649620	.088817	.650180	-.039972
25	.674640	.084295	.675161	-.035463
26	.699661	.079384	.700141	-.030955
27	.724685	.074073	.725121	-.026246
28	.749710	.068392	.750101	-.021637
29	.774736	.062441	.775081	-.017029
30	.799762	.056339	.800062	-.012520
31	.824790	.050048	.825044	-.008411
32	.849818	.043577	.850029	-.004902
33	.874848	.036956	.875017	-.001994
34	.899877	.030305	.900007	.000315
35	.924906	.023634	.925002	.001524
36	.949935	.017002	.950002	.001532
37	.974964	.010331	.975008	.000241
38	.989982	.006306	.990015	-.001393
39	.994988	.005028	.995020	-.002372
40	.999994	.003650	1.000025	-.003650

XNOSE = 0.000000 YNOSE = 0.000000 ANGLE = -.249

TABLE II. - CONTINUED

PAGE 5 OUTPUT

--SUMMARY OF INPUT DATA--

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //

I	X/C	Y/C	THETA	YPS	YPPS	W
1	1.000000	-.003650	-180.00			1.00
2	.994995	-.002372	-178.46			1.00
3	.989990	-.001393	-176.97			1.00
4	.974983	.000241	-172.67			1.00
5	.949978	.001532	-166.10			1.00
6	.924979	.001524	-160.12			1.00
7	.899984	.000315	-154.63			1.00
8	.874995	-.001994	-149.54			1.00
9	.850008	-.004902	-144.77			1.00
10	.825023	-.008411	-140.29			1.00
11	.800042	-.012520	-136.03			1.00
12	.775062	-.017028	-131.98			1.00
13	.750082	-.021637	-128.10			1.00
14	.725103	-.026245	-124.37			1.00
15	.700124	-.030954	-120.77			1.00
16	.675144	-.035463	-117.27			1.00
17	.650164	-.039971	-113.86			1.00
18	.625182	-.044180	-110.53			1.00
19	.600200	-.048188	-107.27			1.00
20	.575215	-.051497	-104.06			1.00
21	.550229	-.054606	-100.89			1.00
22	.500248	-.058823	-94.64			1.00
23	.450261	-.061540	-88.45			1.00
24	.400268	-.063158	-82.24			1.00
25	.350271	-.063675	-75.93			1.00
26	.300270	-.063193	-69.44			1.00
27	.250264	-.061610	-62.66			1.00
28	.200251	-.058428	-55.44			1.00
29	.175242	-.056237	-51.59			1.00
30	.150231	-.053646	-47.52			1.00
31	.125217	-.050454	-43.16			1.00
32	.100201	-.046563	-38.42			1.00
33	.075181	-.041772	-33.12			1.00
34	.050154	-.035581	-26.92			1.00
35	.037637	-.031736	-23.27			1.00
36	.025116	-.026790	-18.96			1.00
37	.012589	-.020445	-13.39			1.00
38	.005060	-.013778	-8.48			1.00
39	.002040	-.009291	-5.38			1.00
40	0.000000	0.000000	0.00			1.00
41	.001943	.013008	5.25			1.00
42	.004911	.020421	8.35			1.00
43	.012366	.030753	13.28			1.00
44	.024818	.041827	18.85			1.00
45	.037283	.049811	23.16			1.00
46	.049755	.056106	26.81			1.00
47	.074712	.065834	33.01			1.00
48	.099679	.073432	38.32			1.00
49	.124652	.079541	43.06			1.00
50	.149629	.084650	47.42			1.00
51	.174610	.089068	51.49			1.00
52	.199593	.092867	55.34			1.00
53	.249566	.098784	62.57			1.00
54	.299548	.102901	69.35			1.00
55	.349535	.105519	75.84			1.00
56	.399530	.106646	82.15			1.00
57	.449530	.106404	88.36			1.00
58	.499536	.104751	94.56			1.00
59	.549550	.101489	100.81			1.00
60	.574559	.099178	103.98			1.00
61	.599571	.096317	107.19			1.00
62	.624587	.092776	110.45			1.00
63	.649603	.088814	113.79			1.00
64	.674623	.084293	117.20			1.00
65	.699644	.079382	120.70			1.00
66	.724666	.074071	124.31			1.00
67	.749691	.068390	128.04			1.00
68	.774716	.062439	131.93			1.00
69	.799742	.056338	135.99			1.00
70	.824769	.050047	140.24			1.00
71	.849797	.043576	144.73			1.00
72	.874825	.036955	149.50			1.00
73	.899854	.030304	154.61			1.00
74	.924883	.023633	160.10			1.00
75	.949911	.017002	166.08			1.00
76	.974940	.010331	172.66			1.00
77	.989957	.006306	176.96			1.00
78	.994962	.005028	178.45			1.00
79	.999968	.003650	179.99			1.00

CHORD = 1.000025

TABLE II. - CONTINUED

PAGE 6 OUTPUT

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //

--SUM OF SQUARES GENERATED DURING SMOOTHING PROCESS--

1	1.2812335	.0077146	.0008193	.0005538	.0003981	.0002971	.0002271	.0001771	.0001406	.0001133
2	.0000927	.0000766	.0000641	.0000542	.0000462	.0000397	.0000344	.0000300	.0000264	.0000234
3	.0000207	.0000186	.0000167	.0000151	.0000137	.0000125	.0000114	.0000105	.0000097	.0000089
4	.0000083	.0000077	.0000072	.0000067	.0000063	.0000059	.0000056	.0000053	.0000050	.0000047
5	.0000045	.0000042	.0000040	.0000038	.0000037	.0000035	.0000033	.0000032	.0000031	.0000029
6	.0000028	.0000027	.0000026	.0000025	.0000024	.0000023	.0000022	.0000021	.0000021	.0000020
7	.0000019	.0000019	.0000018	.0000018	.0000017	.0000016	.0000016	.0000015	.0000015	.0000015
8	.0000014	.0000014	.0000013	.0000013	.0000013	.0000012	.0000012	.0000012	.0000011	.0000011
9	.0000011	.0000010	.0000010	.0000010						

SMOOTHING PROCESS CONVERGED AFTER 84 ITERATIONS

SUM OF SQUARES FROM LEAST SQUARES CUBIC SPLINE SMOOTHING = .7900E-06

ITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND DERIVATIVE CONVERGED IN 2 ITERATIONS AND DELTA = .1554E-03

TABLE II.- CONTINUED

PAGE 7 OUTPUT

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //

--SMOOTHING OUTPUT SUMMARY--

I	THETA	X/C	Y/C	YT/C	YSMO/C	DELTA	YPS	YPPS	DY/DX	D(DY/DX)/DX	CURVATURE
1	-180.00	1.000000	-.003650	-.003634	-.003634	-.000016	.046662	-.251722	-.253004E+00	-.865835E+01	.788889E+01
2	-178.46	.994995	-.002372	-.002482	-.002471	.000100	.040183	-.231661	-.212594E+00	-.751107E+01	.702919E+01
3	-176.97	.989990	-.001393	-.001518	-.001497	.000104	.034363	-.213279	-.177523E+00	-.652514E+01	.622841E+01
4	-172.67	.974983	.000241	.000472	.000520	-.000279	.020044	-.168921	-.967836E-01	-.435650E+01	.429599E+01
5	-166.10	.949978	.001532	.001736	.001805	-.000273	.003201	-.124731	-.139704E-01	-.242847E+01	.242776E+01
6	-160.12	.924979	.001524	.001428	.001491	.000033	-.008931	-.107946	.356487E-01	-.160033E+01	.159728E+01
7	-154.63	.899984	.000315	.000093	.000136	.000179	-.019363	-.109732	.714075E-01	-.127899E+01	.126927E+01
8	-149.54	.874995	-.001994	-.002056	-.002034	.000040	-.029613	-.120810	.101760E+00	-.115470E+01	.113699E+01
9	-144.77	.850008	-.004902	-.004937	-.004924	.000022	-.040022	-.129549	.129104E+00	-.103885E+01	.101341E+01
10	-140.29	.825023	-.008411	-.008471	-.008452	.000041	-.050038	-.126252	.152515E+00	-.845039E+00	.816389E+00
11	-136.03	.800042	-.012520	-.012534	-.012496	-.000023	-.058748	-.108496	.170175E+00	-.582089E+00	.557689E+00
12	-131.98	.775062	-.017028	-.016960	-.016899	-.000129	-.065483	-.082023	.181220E+00	-.314867E+00	.299970E+00
13	-128.10	.750082	-.021637	-.021582	-.021501	-.000136	-.070152	-.055875	.186367E+00	-.106339E+00	.101030E+00
14	-124.37	.725103	-.026245	-.026270	-.026175	-.000070	-.073120	-.035247	.187297E+00	.263583E-01	.250297E-01
15	-120.77	.700124	-.030954	-.030943	-.030835	-.000119	-.074771	-.017226	.185429E+00	.119704E+00	.113785E+00
16	-117.27	.675144	-.035463	-.035542	-.035417	-.000045	-.075079	.007125	.180959E+00	.234592E+00	.223523E+00
17	-113.86	.650164	-.039971	-.039996	-.039845	-.000126	-.073515	.045501	.172834E+00	.411565E+00	.393790E+00
18	-110.53	.625182	-.044180	-.044195	-.044012	-.000167	-.069500	.092741	.159929E+00	.617598E+00	.594639E+00
19	-107.27	.600200	-.048188	-.048010	-.047801	-.000387	-.063212	.128078	.142845E+00	.748502E+00	.726164E+00
20	-104.06	.575215	-.051497	-.051353	-.051136	-.000361	-.055751	.138268	.124115E+00	.751767E+00	.734725E+00
21	-100.89	.550229	-.054606	-.054222	-.054008	-.000598	-.048130	.137405	.105889E+00	.708829E+00	.697072E+00
22	-94.64	.500248	-.058823	-.058651	-.058462	-.000361	-.033768	.126015	.732075E-01	.605127E+00	.600295E+00
23	-88.45	.450261	-.061540	-.061557	-.061407	-.000133	-.021018	.109849	.454342E-01	.510646E+00	.509069E+00
24	-82.24	.400268	-.063158	-.063179	-.063068	-.000089	-.009874	.095697	.215331E-01	.448747E+00	.448435E+00
25	-75.93	.350271	-.063675	-.063676	-.063596	-.000080	.000110	.085695	-.246031E-03	.425414E+00	.425414E+00
26	-69.44	.300270	-.063193	-.063102	-.063045	-.000147	.009507	.080277	-.219416E-01	.446560E+00	.446238E+00
27	-62.66	.250264	-.061610	-.061402	-.061364	-.000246	.018876	.078109	-.459168E-01	.519921E+00	.518281E+00
28	-55.44	.200251	-.058428	-.058381	-.058366	-.000062	.028654	.076969	-.751952E-01	.665834E+00	.660228E+00
29	-51.59	.175242	-.056237	-.056267	-.056267	.000030	.033772	.075308	-.931369E-01	.776423E+00	.766429E+00
30	-47.52	.150231	-.053646	-.053662	-.053681	.000036	.038969	.071015	-.114183E+00	.916081E+00	.898453E+00
31	-43.16	.125217	-.050454	-.050480	-.050519	.000065	.044162	.065608	-.139501E+00	.112455E+01	.109251E+01
32	-38.42	.100201	-.046563	-.046587	-.046646	.000083	.049352	.059800	-.171611E+00	.147545E+01	.141260E+01
33	-33.12	.075181	-.041772	-.041755	-.041836	.000063	.054498	.051449	-.215540E+00	.211150E+01	.197246E+01
34	-26.92	.050154	-.035581	-.035558	-.035667	.000086	.059347	.038221	-.283233E+00	.353240E+01	.314630E+01
35	-23.27	.037637	-.031736	-.031683	-.031809	.000073	.061555	.031007	-.336718E+00	.521141E+01	.443599E+01
36	-18.96	.025116	-.026790	-.026958	-.027087	.000296	.064298	.042033	-.427560E+00	.101328E+02	.787697E+01
37	-13.39	.012589	-.020445	-.020457	-.020541	.000096	.071415	.104413	-.666148E+00	.351781E+02	.202784E+02
38	-8.48	.005060	-.013778	-.013915	-.013927	.000149	.084022	.189553	-.123115E+01	.161688E+03	.405210E+02
39	-5.38	.002040	-.009291	-.009112	-.009085	-.000206	.095459	.233481	-.219907E+01	.661604E+03	.469282E+02
40	0.00	0.000000	0.000000	.000878	.000878	-.000878	.116289	.210007	.100000E+99	.100000E+99	.342209E+02
41	5.25	.001943	.013008	.012368	.012230	.000778	.129279	.073366	.305143E+01	-.742557E+03	.224266E+02
42	8.35	.004911	.020421	.019521	.019288	.001133	.130542	-.026692	.194141E+01	-.202505E+03	.194440E+02
43	13.28	.012366	.030753	.030599	.030278	.000475	.123997	-.125730	.116686E+01	-.576750E+02	.158925E+02
44	18.85	.024818	.041827	.042046	.041716	.000111	.110840	-.144752	.741341E+00	-.209996E+02	.108865E+02
45	23.16	.037283	.049811	.049968	.049664	.000147	.100957	-.118213	.554764E+00	-.106972E+02	.715283E+01
46	26.81	.049755	.056106	.056174	.055883	.000223	.094163	-.094693	.451085E+00	-.644861E+01	.488438E+01
47	33.01	.074712	.065834	.065850	.065548	.000286	.084785	-.078656	.336277E+00	-.329020E+01	.280177E+01
48	38.32	.099679	.073432	.073382	.073060	.000372	.077550	-.077664	.270285E+00	-.213556E+01	.192123E+01
49	43.06	.124652	.079541	.079549	.079212	.000329	.070990	-.080795	.224676E+00	-.157025E+01	.145843E+01
50	47.42	.149629	.084650	.084720	.084372	.000277	.064710	-.084374	.189914E+00	-.123896E+01	.117483E+01
51	51.49	.174610	.089068	.089111	.088754	.000314	.058594	-.087787	.161814E+00	-.102513E+01	.986145E+00
52	55.34	.199593	.092867	.092858	.092493	.000373	.052571	-.091364	.138106E+00	-.881396E+00	.856768E+00
53	62.57	.249566	.098784	.098759	.098380	.000404	.040645	-.097742	.989568E-01	-.704434E+00	.694212E+00
54	69.35	.299548	.102901	.102881	.102493	.000408	.028761	-.103114	.664176E-01	-.607705E+00	.603706E+00
55	75.84	.349535	.105519	.105469	.105077	.000442	.016758	-.108843	.373463E-01	-.561603E+00	.560430E+00
56	82.15	.399530	.106646	.106647	.106248	.000398	.004382	-.115939	.955842E-02	-.554542E+00	.554466E+00
57	88.36	.449530	.106404	.106428	.106023	.000381	-.008696	-.125198	-.187988E-01	-.583912E+00	.583603E+00
58	94.56	.499536	.104751	.104741	.104329	.000423	-.022836	-.136308	-.495027E-01	-.649024E+00	.646646E+00
59	100.81	.549550	.101489	.101418	.101003	.000486	-.038348	-.148059	-.843443E-01	-.750814E+00	.742873E+00
60	103.98	.574559	.099178	.099065	.098652	.000525	-.046684	-.153318	-.103895E+00	-.814662E+00	.801647E+00
61	107.19	.599571	.096317	.096200	.095793	.000524	-.055350	-.155818	-.125027E+00	-.877314E+00	.857137E+00
62	110.45	.624587	.092776	.092782	.092390	.000386	-.063944	-.145695	-.147073E+00	-.886642E+00	.858633E+00
63	113.79	.649603	.088814	.088809	.088439	.000375	-.071706	-.121296	-.168492E+00	-.825262E+00	.791326E+00
64	117.20	.674623	.084293	.084319	.083976	.000317	-.077998	-.090198	-.187890E+00	-.723432E+00	.686747E+00
65	120.70	.699644	.079382	.079375	.079059	.000323	-.082563	-.059161	-.204633E+00	-.611839E+00	.575326E+00
66	124.31	.724666	.074071	.074047	.073758	.000313	-.085430	-.031832	-.218698E+00	-.508800E+00	.474363E+00
67	128.04	.749691	.068390	.068399	.068134	.000256	-.086777	-.009485	-.230394E+00	-.422272E+00	.390751E+00
68	131.93	.774716	.062439	.062463	.062243	.000196	-.086819	.008247	-.240122E+00	-.351437E+00	.323094E+00
69	135.99	.799742	.056338	.056346	.056129	.000209	-.085739	.022266	-.248216E+00	-.291593E+00	.266581E+00
70	140.24	.824769	.050047	.050024	.049831	.000216	-.083657	.033773	-.254843E+00	-.233796E+00	.212738E+00
71	144.73	.849797	.043576	.043554	.043385	.000190	-.080635	.043305	-.259983E+00	-.171947E+00	.155879E+00
72	149.50	.874825	.036955	.036973	.036831	.000124	-.076723	.050687	-.263529E+00	-.105605E+00	.954877E-01
73	154.61	.899854	.030304	.030323	.030208	.000096	-.071994	.055559	-.265400E+00	-.376401E-01	.339867E-01
74	160.10	.924883	.023633	.023646	.023559	.000074	-.066576	.057374	-.265666E+00	.223672E-01	.201922E-01
75	166.08	.949911	.017002	.016977	.016919	.000083	-.060694	.055334	-.264805E+00	.496267E-01	.448297E-01
76	172.66	.974940	.010331	.010329	.010301	.000030	-.054758	.048085	-.264360E+00	-.205638E-01	.185822E-01
77	176.96	.989957	.006306	.006335	.006323	-.000017	-.051450	.040186	-.265752E+00	-.174626E+00	.157636E+00
78	178.45	.994962	.005028	.004996	.004990	.000038	-.050442	.036839	-.266829E+00	-.257499E+00	.232258E+00
79	179.99	.999968	.003650	.003651	.003651	-.000001	-.049504	.033162	-.268368E+00	-.359699E+00	.324066E+00

LEADING-EDGE RADIUS/C= .029222

DY/DX=0. AT X/C= .943926 Y/C= .001847 THETA= -164.604

DY/DX=0. AT X/C= .350849 Y/C= -.063596 THETA= -76.004

DY/DX=0. AT X/C= .416679 Y/C= .106330 THETA= 84.281

TABLE II. - CONTINUED

PAGE 8 OUTPUT

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS

CHECK OF SMOOTHED COORDINATES DF= .000100

I	(YSMO/C-CHECK VALUE)	(YPPS-CHECK VALUE)
1	-.000000	-.000969
2	.000000	-.000343
3	.000000	.000447
4	-.000000	.001928
5	-.000001	.002463
6	-.000000	.001729
7	.000002	.000338
8	.000003	-.001214
9	.000002	-.002315
10	.000001	-.002145
11	-.000000	-.000833
12	-.000001	.000664
13	-.000001	.001267
14	-.000000	.000543
15	.000001	-.001177
16	.000001	-.002626
17	.000000	-.001807
18	-.000001	.002097
19	-.000002	.004562
20	-.000004	.003987
21	-.000004	.002889
22	-.000003	.001281
23	-.000000	-.000125
24	.000000	-.000594
25	.000001	-.000636
26	.000000	-.000322
27	-.000001	.000155
28	-.000002	.000774
29	-.000001	.000666
30	-.000000	.000042
31	.000000	-.000258
32	.000000	.000161
33	-.000001	.000756
34	.000001	-.000461
35	.000002	-.002933
36	.000003	-.005923
37	.000001	-.004470
38	-.000006	.004796
39	-.000005	.011671
40	-.000005	.014370
41	.000002	.000565
42	.000001	-.007354
43	.000005	-.011198
44	-.000001	-.003994
45	-.000001	.001782
46	-.000002	.003287
47	-.000001	.001379
48	.000001	-.000300
49	.000001	-.000540
50	.000000	-.000116
51	-.000000	.000129
52	-.000000	.000010
53	.000000	-.000202
54	.000000	.000024
55	-.000000	.000210
56	-.000000	.000311
57	.000000	.000179
58	.000001	-.000126
59	.000001	-.000797
60	.000002	-.001428
61	.000001	-.002166
62	.000001	-.002127
63	.000000	-.000790
64	-.000000	.000217
65	-.000000	.000592
66	-.000000	.000667
67	-.000000	.000620
68	-.000000	.000463
69	-.000000	.000286
70	-.000000	.000195
71	-.000000	.000218
72	-.000000	.000259
73	.000000	.000272
74	.000000	.000311
75	.000000	.000382
76	.000000	.000389
77	-.000000	.000238
78	-.000000	.000144
79	.000000	.000094
SUM OF SQUARES=	.000000	.000750

PAGE 9 OUTPUT

```
        THE FOLLOWING DATA HAVE BEEN PUNCHED      IPUNCH=   1

//    GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS                              //

IOP =    0

NU =   40

DX =   0.000000   .001943   .004911   .012366   .024818   .037284   .049756   .074714
        .099682   .124655   .149633   .174614   .199596   .249573   .299555   .349544
        .399540   .449541   .499549   .549564   .574574   .599587   .624602   .649620
        .674640   .699661   .724685   .749710   .774736   .799762   .824790   .849818
        .874848   .899877   .924906   .949935   .974964   .989982   .994988   .999994

DY =    .000878   .012230   .019288   .030279   .041717   .049666   .055884   .065550
        .073062   .079214   .084374   .088757   .092496   .098383   .102496   .105080
        .106251   .106026   .104331   .101005   .098655   .095795   .092392   .088442
        .083978   .079061   .073760   .068136   .062245   .056131   .049832   .043387
        .036832   .030209   .023560   .016920   .010301   .006323   .004991   .003651

NL =   40

DX =   0.000000   .002040   .005060   .012589   .025117   .037638   .050155   .075182
        .100204   .125221   .150235   .175246   .200256   .250270   .300278   .350280
        .400279   .450272   .500261   .550243   .575229   .600215   .625198   .650180
        .675161   .700141   .725121   .750101   .775081   .800062   .825044   .850029
        .875017   .900007   .925002   .950002   .975008   .990015   .995020  1.000025

DY =    .000878  -.009085  -.013927  -.020541  -.027087  -.031809  -.035668  -.041837
       -.046647  -.050521  -.053683  -.056268  -.058367  -.061366  -.063047  -.063597
       -.063070  -.061409  -.058463  -.054009  -.051138  -.047802  -.044014  -.039846
       -.035418  -.030836  -.026176  -.021502  -.016900  -.012497  -.008452  -.004924
       -.002034   .000136   .001491   .001805   .000520  -.001497  -.002472  -.003634
```

PAGE 10 OUTPUT

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //

--THICKNESS AND CAMBER DISTRIBUTION--

I	XU/C	YU/C	XL/C	YL/C	X/C	Y/C	T/C/2	SLOPE	ERROR
1	.999968	.003651	.998234	-.003201	.999101	.000225	.003534	-14.2073	.000005
2	.994962	.004990	.993303	-.002122	.994133	.001434	.003652	-13.1308	.000026
3	.989957	.006323	.988331	-.001212	.989144	.002556	.003854	-12.1802	.000023
4	.974940	.010301	.973251	.000681	.974095	.005491	.004883	-9.9589	.000039
5	.949911	.016919	.947878	.001829	.948894	.009374	.007613	-7.6733	.000028
6	.924883	.023559	.922434	.001395	.923658	.012477	.011149	-6.3035	.000019
7	.899854	.030208	.897053	-.000078	.898453	.015065	.015208	-5.2838	.000014
8	.874825	.036831	.871828	-.002362	.873326	.017234	.019653	-4.3739	.000012
9	.849797	.043385	.846803	-.005343	.848300	.019021	.024410	-3.5160	.000001
10	.824769	.049831	.821955	-.008924	.823362	.020454	.029411	-2.7419	.000005
11	.799742	.056129	.797212	-.012980	.798477	.021574	.034578	-2.0968	.000006
12	.774716	.062243	.772501	-.017364	.773609	.022439	.039819	-1.5938	.000000
13	.749691	.068134	.747800	-.021927	.748745	.023104	.045040	-1.2025	.000005
14	.724666	.073758	.723151	-.026541	.723909	.023609	.050155	-.8656	.000003
15	.699644	.079059	.698614	-.031115	.699129	.023972	.055090	-.5352	.000004
16	.674623	.083976	.674209	-.035586	.674416	.024195	.059781	-.1983	.000003
17	.649603	.088439	.649866	-.039897	.649735	.024271	.064168	.1174	.000003
18	.624587	.092390	.625454	-.043969	.625020	.024210	.068181	.3644	.000003
19	.599571	.095793	.600862	-.047706	.600217	.024043	.071753	.5153	.000001
20	.574559	.098652	.576102	-.051026	.575331	.023813	.074843	.5906	.000001
21	.549550	.101003	.551259	-.053898	.550405	.023552	.077455	.6323	.000001
22	.499536	.104329	.501520	-.058368	.500528	.022980	.081355	.6985	.000001
23	.449530	.106023	.451823	-.061336	.450677	.022344	.083687	.7851	.000000
24	.399530	.106248	.402236	-.063025	.400883	.021611	.084647	.9159	.000000
25	.349535	.105077	.352752	-.063595	.351144	.020741	.084351	1.0926	.000000
26	.299548	.102493	.303336	-.063111	.301442	.019691	.082823	1.3106	.000000
27	.249566	.098380	.253936	-.061529	.251751	.018426	.079985	1.5654	.000000
28	.199593	.092493	.204506	-.058680	.202049	.016907	.075626	1.8615	.000000
29	.174610	.088754	.179774	-.056681	.177192	.016037	.072763	2.0338	.000000
30	.149629	.084372	.155057	-.054222	.152343	.015075	.069350	2.2425	.000000
31	.124652	.079212	.130394	-.051227	.127523	.013992	.065282	2.5205	.000000
32	.099679	.073060	.105827	-.047589	.102753	.012736	.060403	2.9170	.000001
33	.074712	.065548	.081471	-.043151	.078092	.011198	.054454	3.5578	.000002
34	.049755	.055883	.057757	-.037724	.053756	.009079	.046974	4.8859	.000002
35	.037283	.049664	.046504	-.034609	.041894	.007528	.042388	6.2449	.000001
36	.024818	.041716	.036053	-.031268	.030435	.005224	.036922	8.7511	.000000
37	.012366	.030278	.026086	-.027496	.019226	.001391	.029691	13.3584	.000003
38	.004911	.019288	.018942	-.024216	.011927	-.002464	.022855	17.8758	.000003
39	.001943	.012230	.014553	-.021786	.008248	-.004778	.018139	20.3388	.000002
40	0.000000	.000878	.007551	-.016593	.003776	-.007858	.009517	23.3746	.000005
41	.002056	-.009120	.002056	-.009120	.002056	-.009120	0.000000	36.7380	.000000

TABLE II. - CONCLUDED

PAGE 11 OUTPUT

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //

--UPPER SURFACE INTERPOLATED COORDINATES--

I	XU	YU	DY/DX	D(DY/DX)/DX	CURVATURE
1	0.000000	.008778	.100000E+99	.100000E+99	.342209E+02
2	.010000	.089074	.417266E+01	-.196293E+04	.248476E+02
3	.020000	.124016	.301025E+01	-.712541E+03	.223259E+02
4	.050000	.194594	.192363E+01	-.197447E+03	.193756E+02
5	.100000	.273409	.132338E+01	-.762275E+02	.167032E+02
6	.500000	.559929	.449512E+00	-.639751E+01	.485424E+01
7	.800000	.672815	.319754E+00	-.296890E+01	.256555E+01
8	1.000000	.731472	.269601E+00	-.212599E+01	.191361E+01
9	2.000000	.925495	.137748E+00	-.879432E+00	.854982E+00
10	3.000000	1.025230	.661428E-01	-.607111E+00	.603149E+00
11	4.000000	1.062524	.929760E-02	-.554703E+00	.554631E+00
12	5.000000	1.043058	-.498038E-01	-.649796E+00	.647386E+00
13	6.000000	.957391	-.125403E+00	-.877407E+00	.857109E+00
14	7.000000	.789865	-.204851E+00	-.610511E+00	.574003E+00
15	8.000000	.560652	-.248291E+00	-.291119E+00	.266134E+00
16	9.000000	.301692	-.265405E+00	-.373856E-01	.337568E-01
17	9.999682	.036510	-.268368E+00	-.359699E+00	.324066E+00

CHORD = 10.000000

PAGE 12 OUTPUT

TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //

--LOWER SURFACE INTERPOLATED COORDINATES--

I	XL	YL	DY/DX	D(DY/DX)/DX	CURVATURE
1	0.000000	.008778	.100000E+99	.100000E+99	.342209E+02
2	.010000	-.063026	-.335261E+01	.191941E+04	.448226E+02
3	.020000	-.089958	-.222619E+01	.681620E+03	.468941E+02
4	.050000	-.138530	-.124092E+01	.164756E+03	.407024E+02
5	.100000	-.186816	-.777581E+00	.526254E+02	.258900E+02
6	.500000	-.356231	-.283778E+00	.354689E+01	.315786E+01
7	.800000	-.428505	-.205762E+00	.195000E+01	.183240E+01
8	1.000000	-.466116	-.171908E+00	.147902E+01	.141580E+01
9	2.000000	-.583469	-.753524E-01	.666760E+00	.661121E+00
10	3.000000	-.630395	-.220622E-01	.446844E+00	.446518E+00
11	4.000000	-.630743	.214127E-01	.448574E+00	.448266E+00
12	5.000000	-.584799	.730575E-01	.604586E+00	.599778E+00
13	6.000000	-.478298	.142695E+00	.748504E+00	.726211E+00
14	7.000000	-.308580	.185414E+00	.120328E+00	.114379E+00
15	8.000000	-.125036	.170199E+00	-.581575E+00	.557190E+00
16	9.000000	.001376	.713875E-01	-.127916E+01	.126944E+01
17	10.000000	-.036343	-.253004E+00	-.865835E+01	.788889E+01

TABLE III.- SAMPLE OUTPUT FOR AIRFOIL SCALING PROGRAM

PAGE 1 OUTPUT

--INPUT DATA--

TITLE-- * GA(W)-1 SMOOTHED *

NT= 41 IPLOT= 1 IPUNCH= 0 IOP= 0

X/C=	.182000E-02	.338100E-02	.791900E-02	.116860E-01	.192200E-01	.306300E-01	.420950E-01	.538920E-01
	.781350E-01	.102793E+00	.127608E+00	.152464E+00	.177313E+00	.202154E+00	.251836E+00	.301530E+00
	.351251E+00	.401010E+00	.450814E+00	.500657E+00	.550503E+00	.575409E+00	.600279E+00	.625084E+00
	.649822E+00	.674533E+00	.699266E+00	.724045E+00	.748863E+00	.773699E+00	.798548E+00	.823428E+00
	.848372E+00	.873407E+00	.898533E+00	.923728E+00	.948946E+00	.974128E+00	.989166E+00	.994153E+00
	.999118E+00							
Y/C=	-.843100E-02	-.739600E-02	-.455900E-02	-.239900E-02	.126700E-02	.502700E-02	.734200E-02	.889900E-02
	.109520E-01	.123760E-01	.135190E-01	.145050E-01	.153780E-01	.161540E-01	.174660E-01	.185130E-01
	.193480E-01	.200120E-01	.205450E-01	.209800E-01	.213350E-01	.214760E-01	.215800E-01	.216200E-01
	.215600E-01	.213760E-01	.210550E-01	.205940E-01	.199850E-01	.192040E-01	.182080E-01	.169490E-01
	.153830E-01	.134750E-01	.111980E-01	.850900E-02	.529300E-02	.127700E-02	-.174200E-02	-.288900E-02
	-.412100E-02							
T/C/2=	0.	.902500E-02	.178730E-01	.226660E-01	.296660E-01	.370210E-01	.424880E-01	.470340E-01
	.544440E-01	.603680E-01	.652520E-01	.693270E-01	.727400E-01	.756000E-01	.799640E-01	.828210E-01
	.843640E-01	.846590E-01	.836790E-01	.813180E-01	.774050E-01	.748020E-01	.717340E-01	.681920E-01
	.642030E-01	.598240E-01	.551240E-01	.501760E-01	.450520E-01	.398330E-01	.346060E-01	.294590E-01
	.244720E-01	.197160E-01	.152540E-01	.111720E-01	.761800E-02	.488500E-02	.385800E-02	.365500E-02
	.353500E-02							
SLOPE=	.592326E+00	.383967E+00	.337573E+00	.299424E+00	.228491E+00	.152664E+00	.108353E+00	.828360E-01
	.576090E-01	.462850E-01	.399860E-01	.355510E-01	.318050E-01	.284960E-01	.229570E-01	.184780E-01
	.148330E-01	.119300E-01	.972200E-02	.807400E-02	.650200E-02	.547000E-02	.388800E-02	.123000E-02
	-.277500E-02	-.780800E-02	-.133180E-01	-.190370E-01	-.252480E-01	-.326600E-01	-.419630E-01	-.533870E-01
	-.665680E-01	-.809090E-01	-.962880E-01	-.114138E+00	-.138752E+00	-.179363E+00	-.217802E+00	-.233989E+00
	-.252162E+00							

LT= 2 NEW T/C = .130000 .200000

(T/C)MAX FOR INPUT AIRFOIL = .169405 AT X/C = .387925

TABLE III.- CONTINUED

PAGE 2 OUTPUT

TITLE-- * GA(W)-1 SMOOTHED *

SCALED COORDINATES FOR (T/C)MAX = .2000

	UPPER		LOWER	
I	X/C	Y/C	X/C	Y/C
1	0.000000	.002481	0.000000	.002481
2	.001540	.015339	.002428	-.008425
3	.004400	.023152	.007977	-.017262
4	.011888	.035353	.015506	-.024450
5	.024575	.048189	.020174	-.027946
6	.037252	.057165	.027742	-.032821
7	.049869	.064188	.037858	-.038142
8	.074987	.075064	.048093	-.042493
9	.100029	.083506	.059051	-.046403
10	.125043	.090425	.082383	-.053177
11	.150050	.096227	.106619	-.058773
12	.175058	.101134	.131197	-.063407
13	.200067	.105290	.155864	-.067239
14	.250087	.111761	.180515	-.070401
15	.300103	.116186	.205149	-.073007
16	.350115	.118846	.254418	-.076856
17	.400120	.119861	.303713	-.079188
18	.450118	.119240	.353067	-.080180
19	.500107	.116891	.402503	-.079868
20	.550096	.112631	.452037	-.078182
21	.575094	.109702	.501656	-.074963
22	.600098	.106187	.551283	-.069994
23	.625114	.102049	.576059	-.066783
24	.650142	.097283	.600756	-.063060
25	.675175	.091932	.625312	-.058842
26	.700204	.086063	.649722	-.054196
27	.725225	.079760	.674073	-.049212
28	.750239	.073100	.698472	-.043985
29	.775248	.066155	.722971	-.038604
30	.800256	.058983	.747555	-.033161
31	.825259	.051639	.772179	-.027777
32	.850250	.044177	.796831	-.022595
33	.875225	.036647	.821550	-.017767
34	.900182	.029101	.846409	-.013434
35	.925129	.021596	.871465	-.009718
36	.950069	.014189	.896722	-.006722
37	.975017	.006946	.922126	-.004591
38	.989999	.002703	.947583	-.003612
39	.994998	.001308	.972961	-.004394
40	1.000000	-.000079	.988032	-.006184
41			.992999	-.007081
42			1.000000	-.008667

TABLE III. - CONCLUDED

PAGE 3 OUTPUT

TITLE-- * GA(W)-1 SMOOTHED *

CAMBER AND THICKNESS DISTRIBUTIONS FOR (T/C)MAX = .2000

		CAMBER		THICKNESS
I	X/C	Y/C	SLOPE	T/C/2
1	.002428	-.008425	33.9378	0.000000
2	.003988	-.007390	21.9997	.021310
3	.008523	-.004555	19.3415	.042202
4	.012287	-.002397	17.1557	.053519
5	.019815	.001266	13.0916	.070048
6	.031216	.005023	8.7470	.087414
7	.042673	.007336	6.2082	.100323
8	.054460	.008892	4.7462	.111057
9	.078685	.010944	3.3008	.128554
10	.103324	.012366	2.6519	.142541
11	.128120	.013509	2.2910	.154073
12	.152957	.014494	2.0369	.163695
13	.177787	.015366	1.8223	.171754
14	.202608	.016142	1.6327	.178507
15	.252252	.017453	1.3153	.188812
16	.301908	.018499	1.0587	.195558
17	.351591	.019333	.8499	.199201
18	.401312	.019997	.6835	.199897
19	.451077	.020529	.5570	.197583
20	.500882	.020964	.4626	.192009
21	.550690	.021319	.3725	.182769
22	.575576	.021459	.3134	.176623
23	.600427	.021563	.2228	.169379
24	.625213	.021603	.0705	.161015
25	.649932	.021543	-.1590	.151597
26	.674624	.021360	-.4474	.141257
27	.699338	.021039	-.7631	.130159
28	.724098	.020578	-1.0907	.118476
29	.748897	.019970	-1.4466	.106377
30	.773714	.019189	-1.8713	.094054
31	.798544	.018194	-2.4043	.081712
32	.823405	.016936	-3.0588	.069559
33	.848329	.015371	-3.8141	.057783
34	.873345	.013465	-4.6357	.046554
35	.898452	.011189	-5.5169	.036018
36	.923628	.008502	-6.5396	.026379
37	.948826	.005289	-7.9499	.017988
38	.973989	.001276	-10.2767	.011534
39	.989015	-.001741	-12.4791	.009110
40	.993998	-.002887	-13.4066	.008630
41	.998960	-.004118	-14.4478	.008347

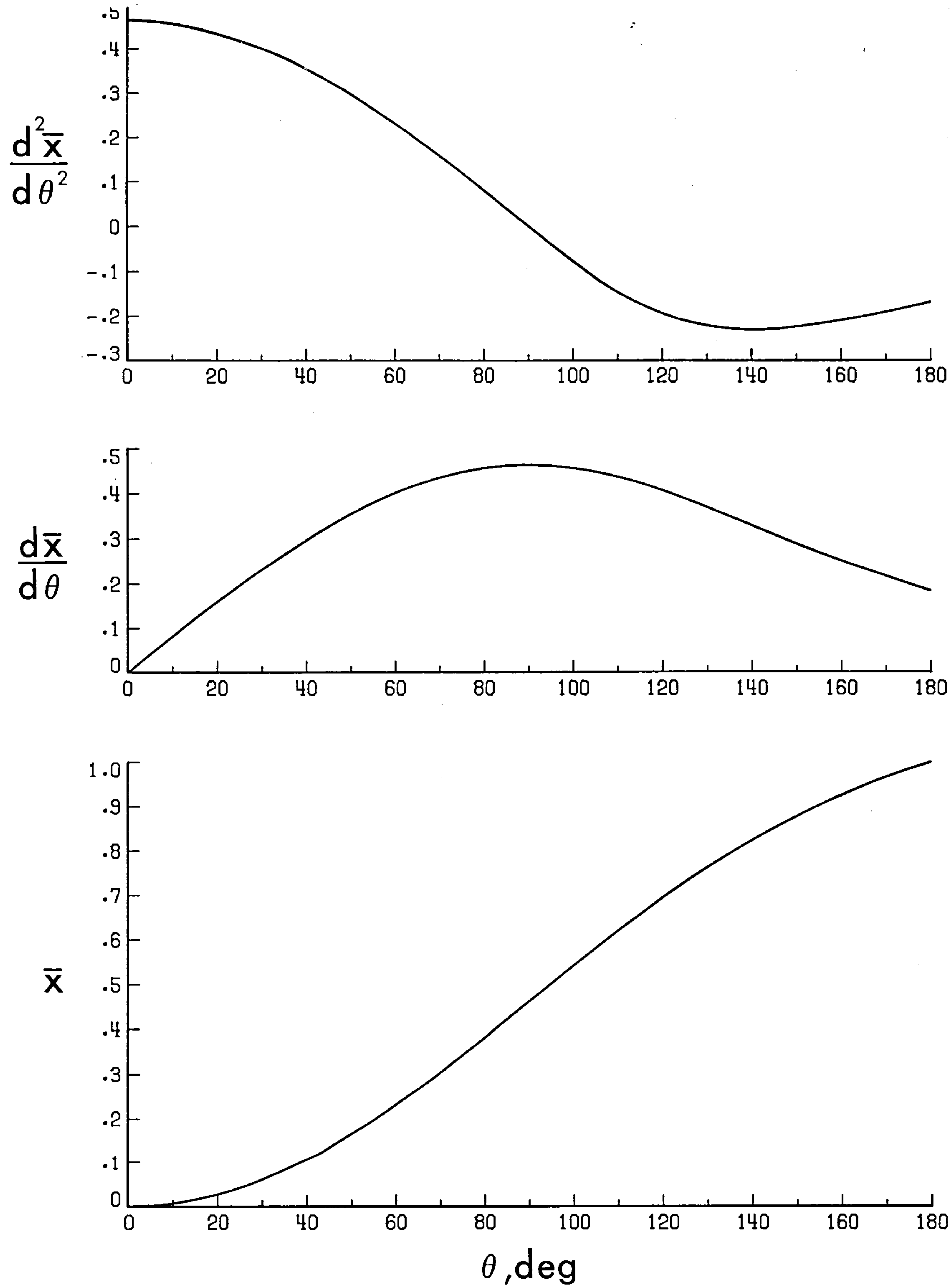


Figure 1.- Properties of θ-transformation function.

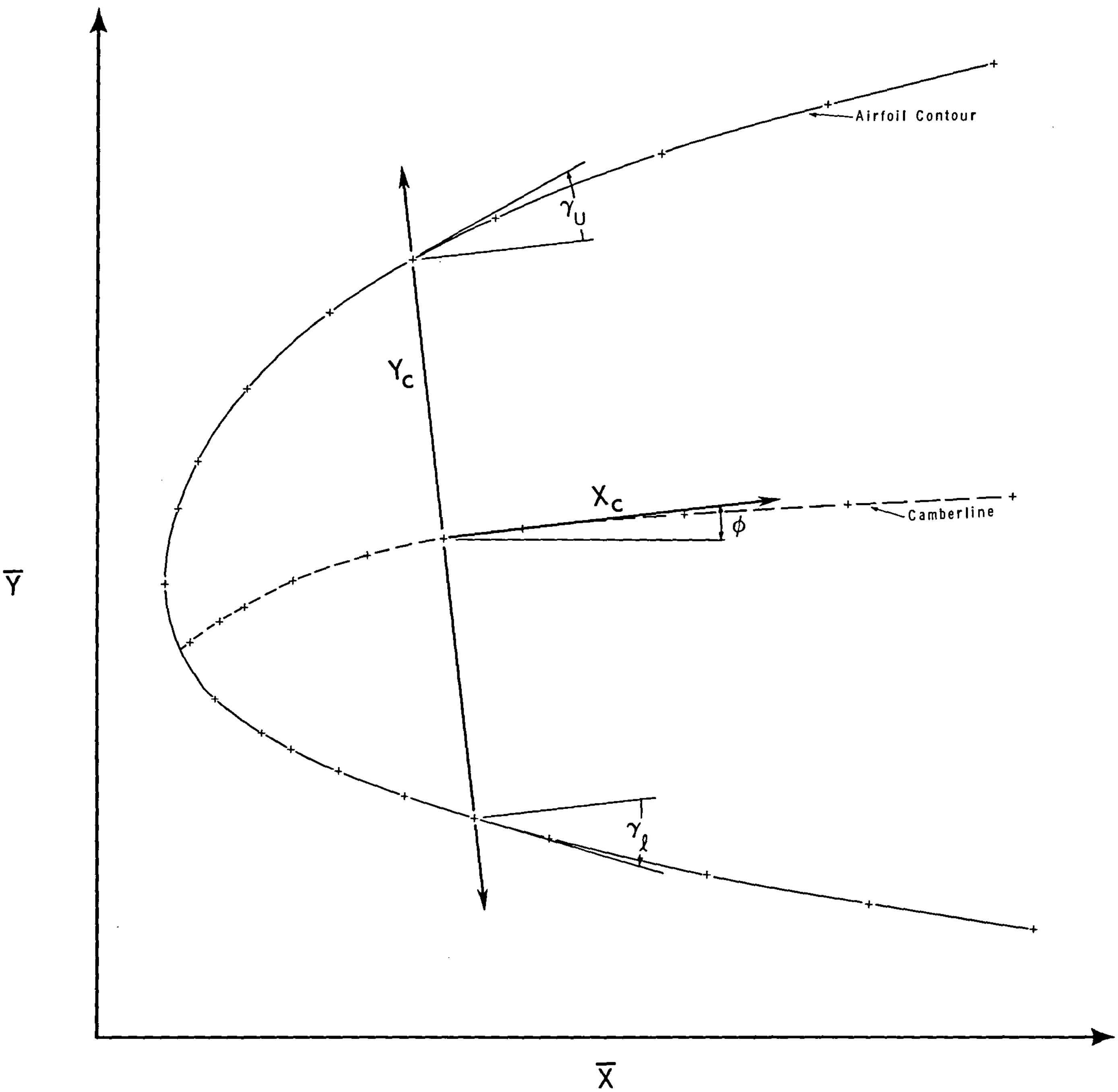


Figure 2. - Camberline axis system.

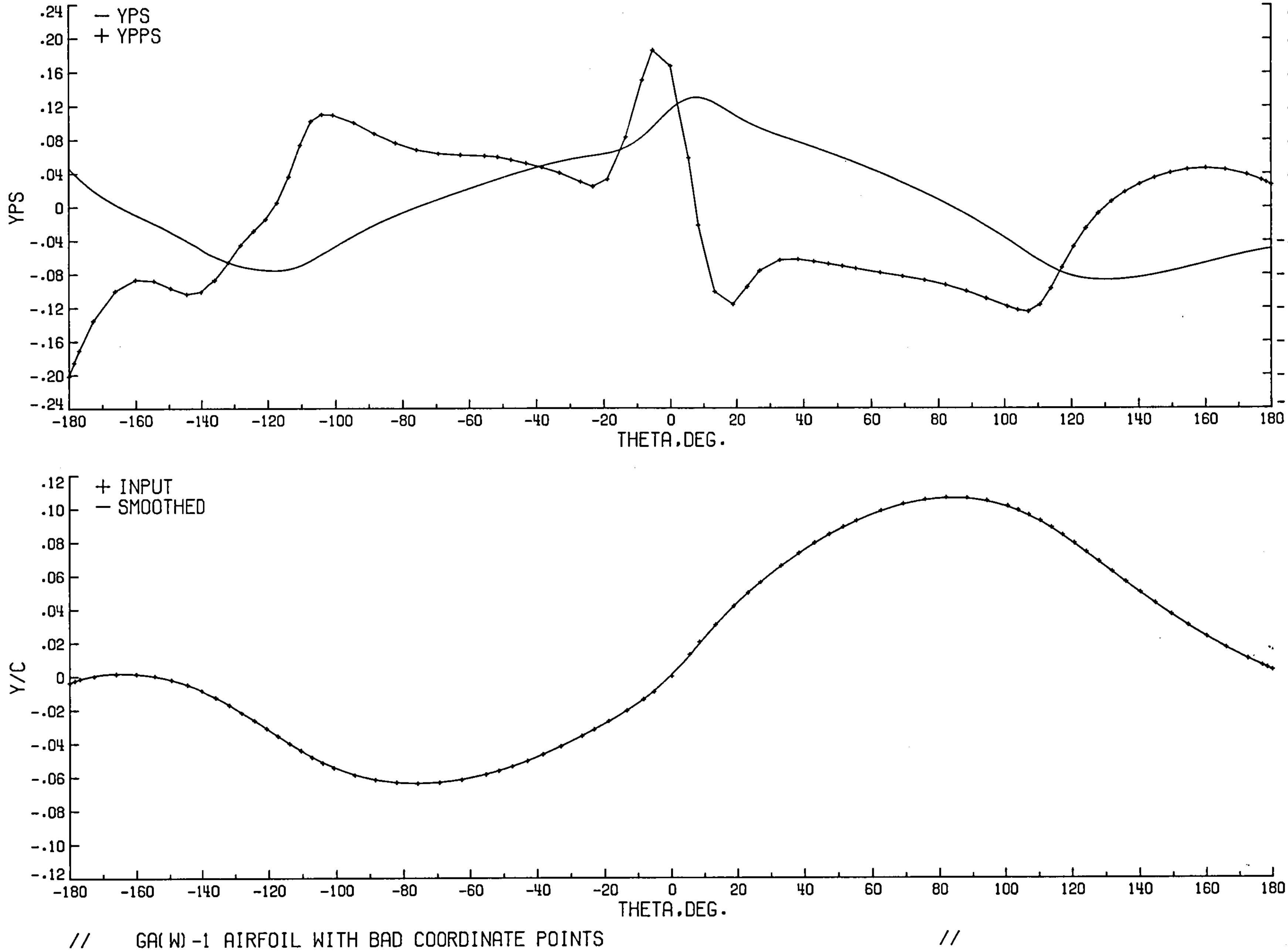


Figure 3. - Sample plot for airfoil smoothing program plotting option 1.

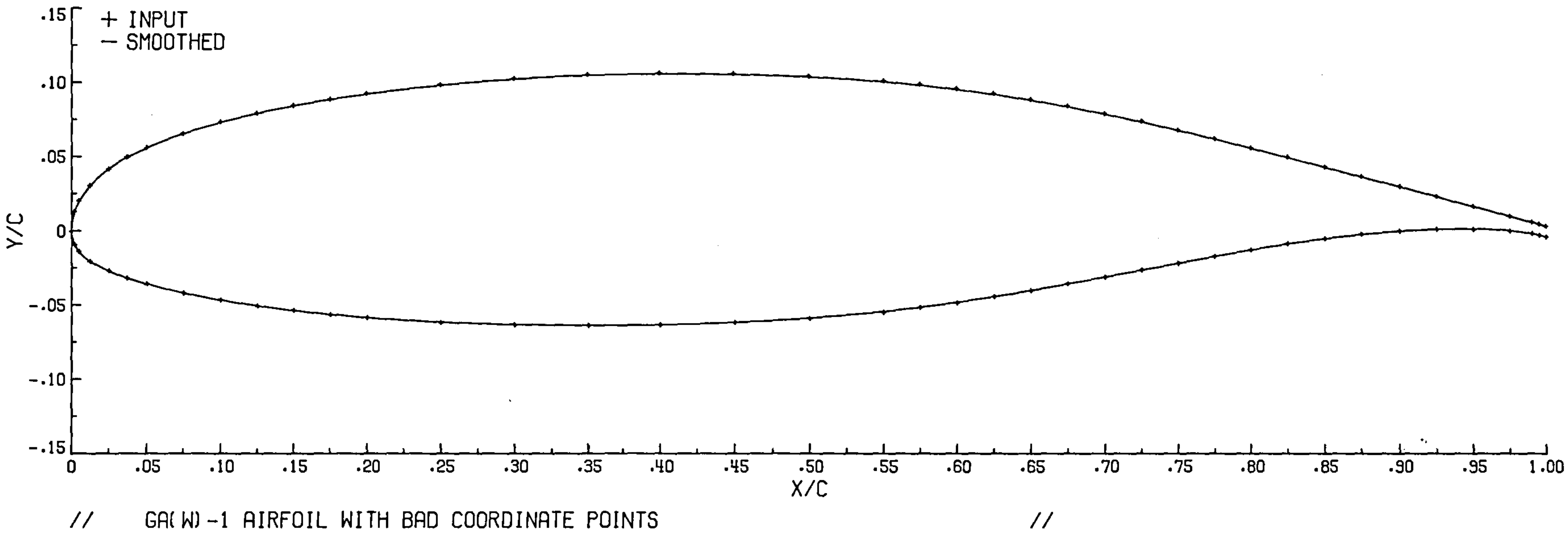


Figure 4. - Sample plot for airfoil smoothing program plotting option 2.

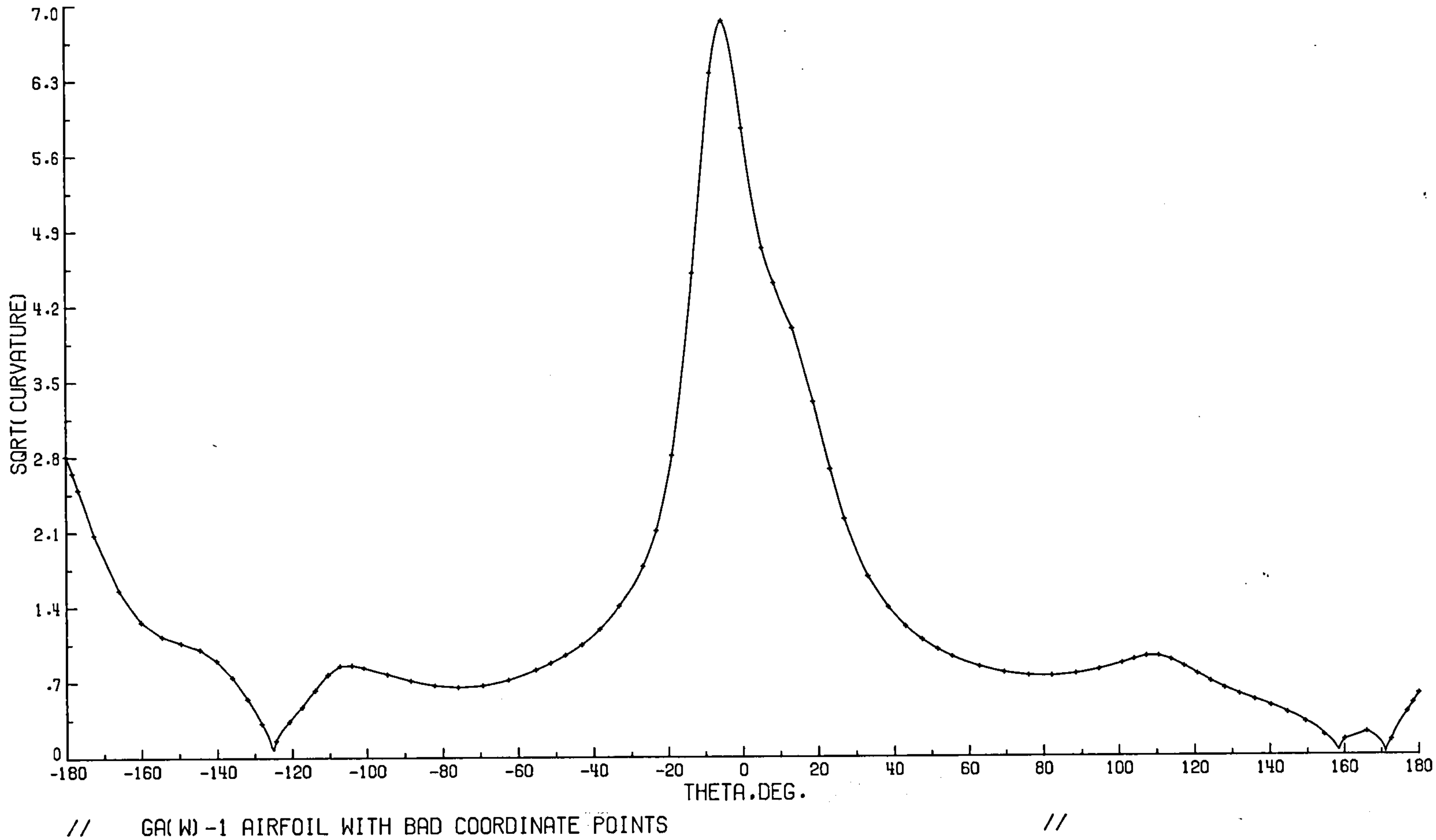


Figure 5.- Sample plot for airfoil smoothing program plotting option 3.

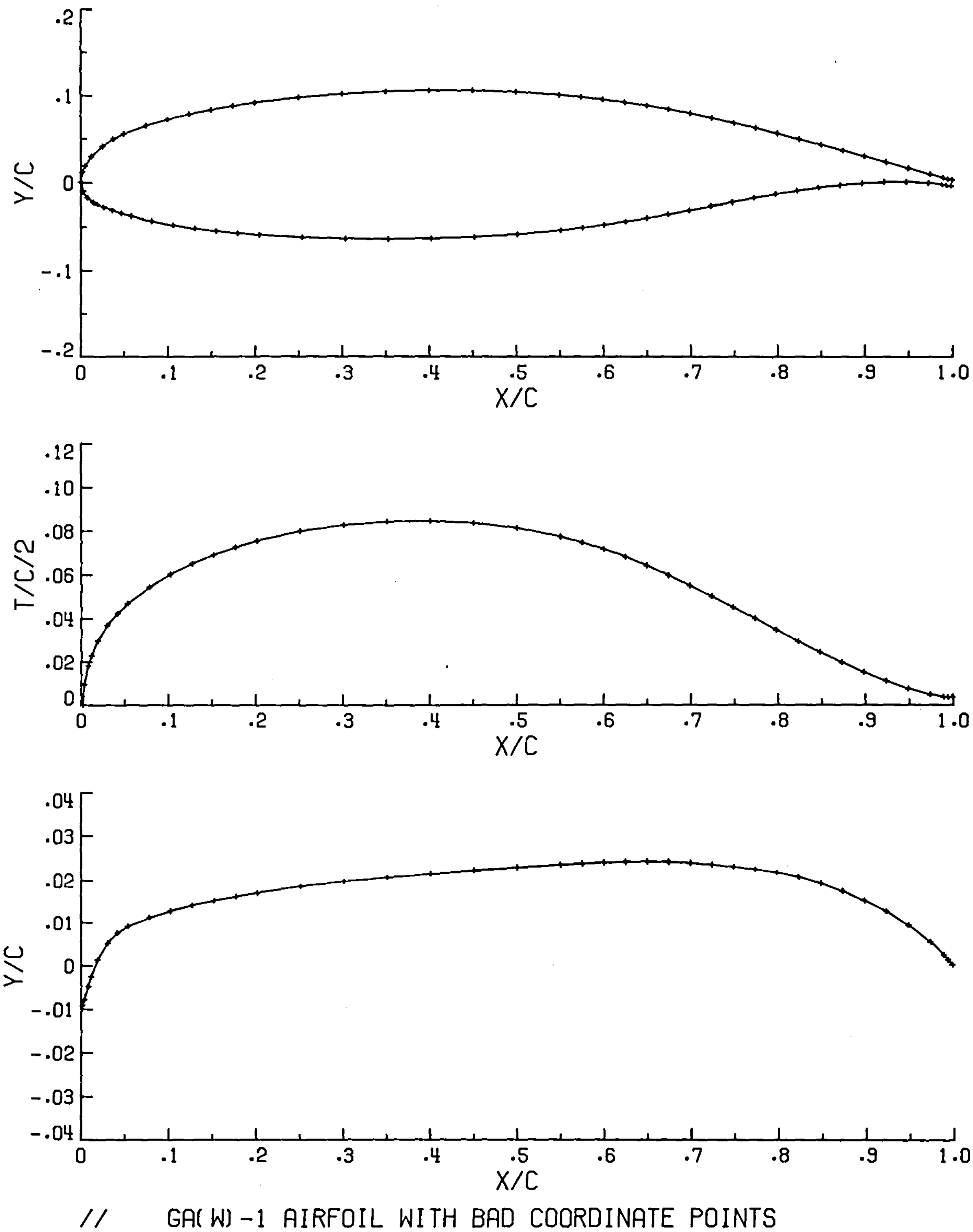


Figure 6.- Sample plot for airfoil smoothing program plotting option 4.

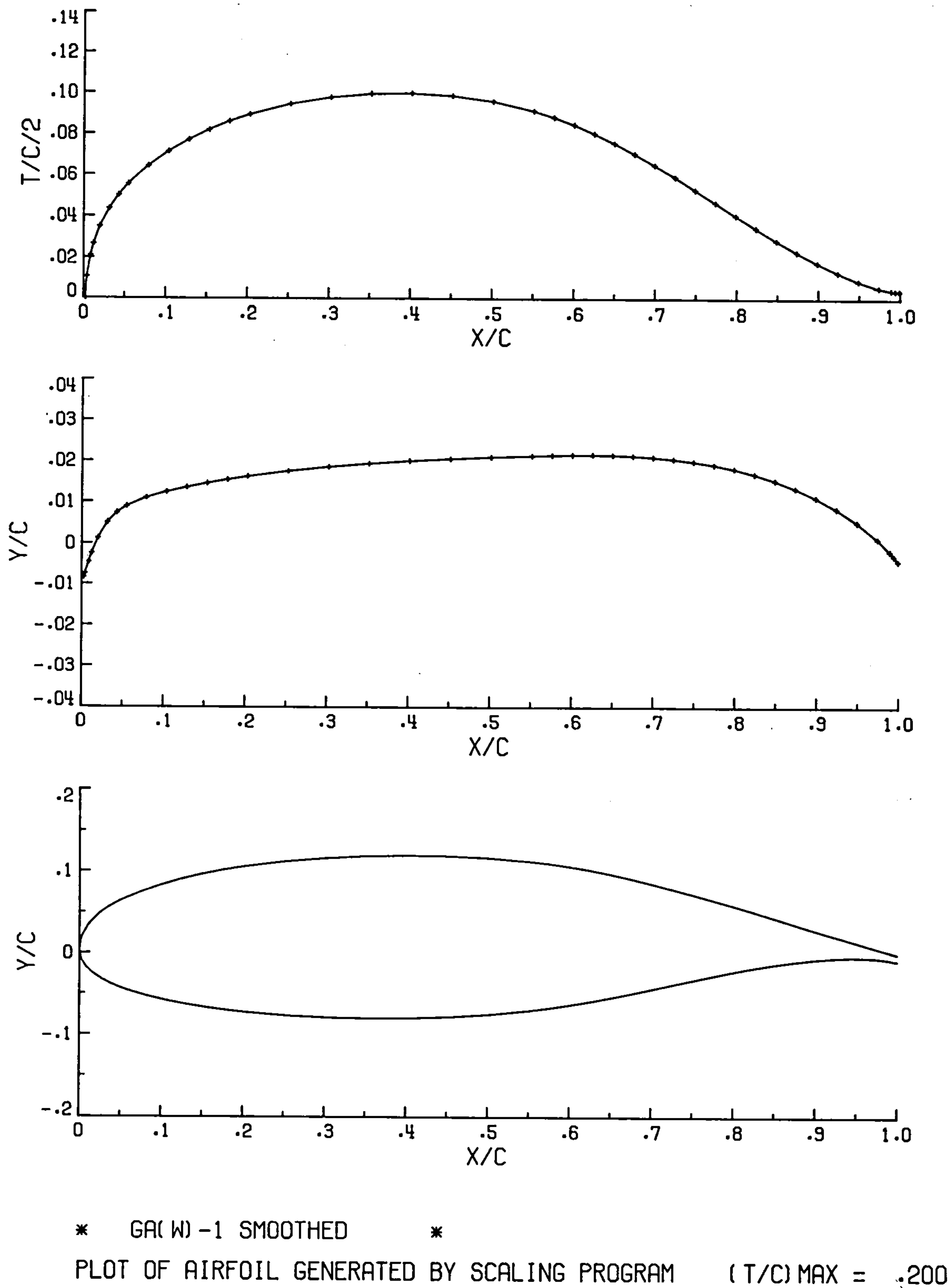
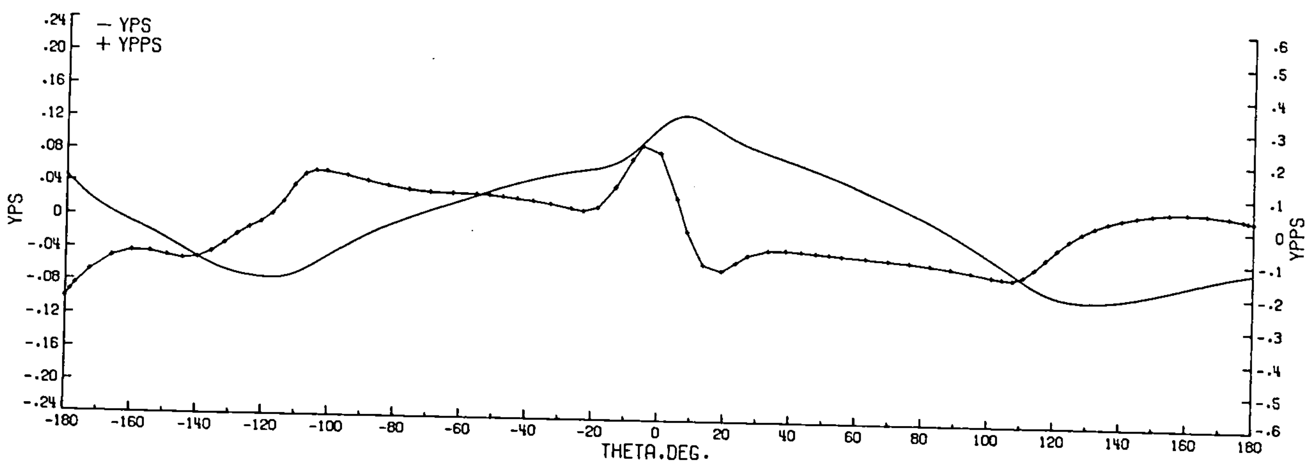
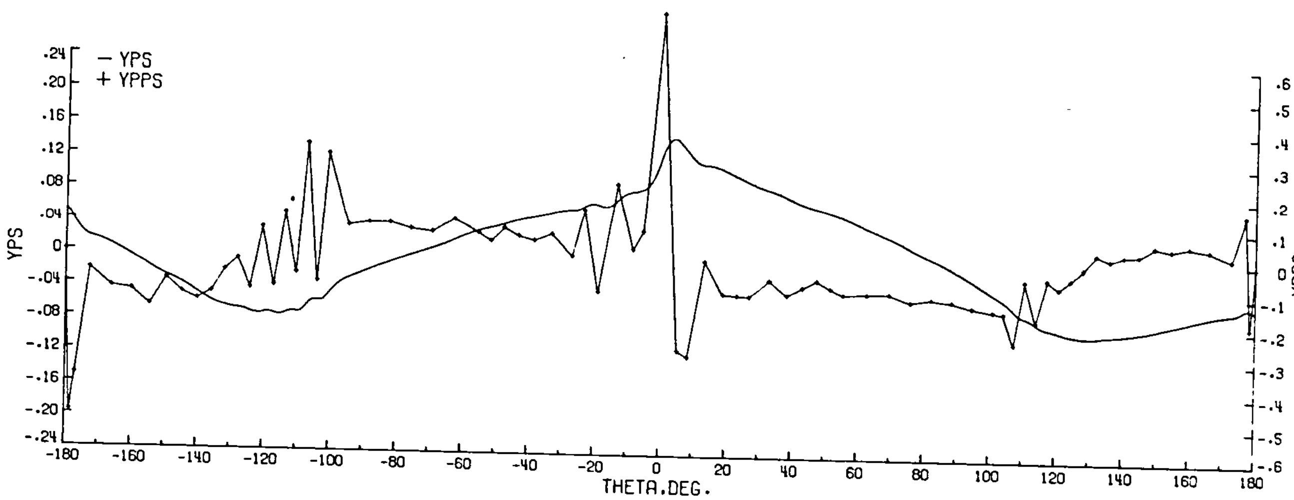


Figure 7. - Sample plot for airfoil scaling program.

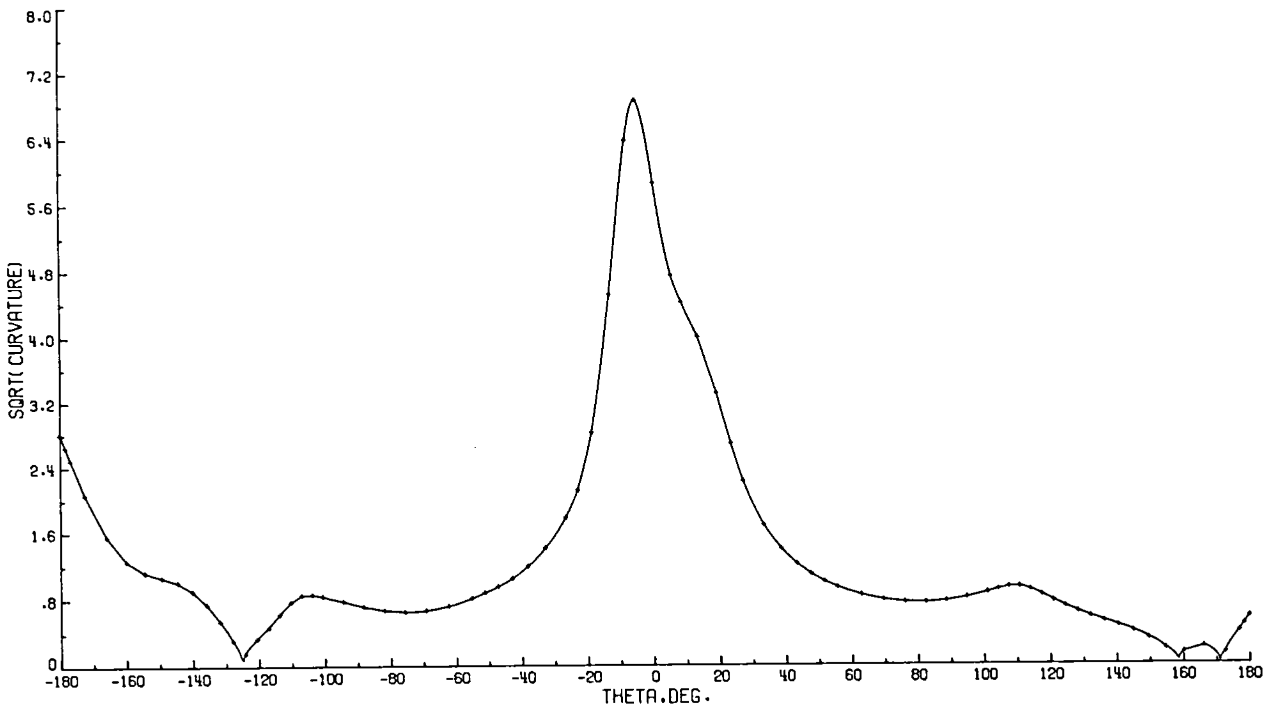


(b) Smoothed

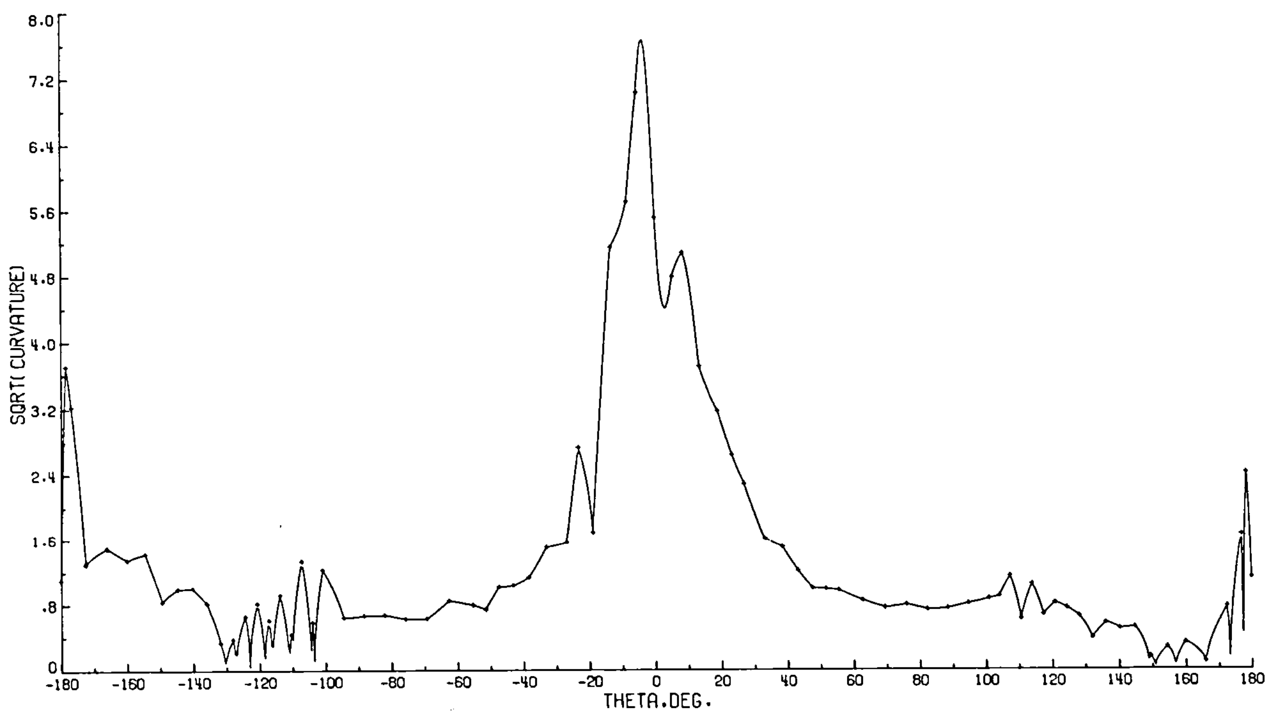


(a) Unsmoothed

Figure 8. - Comparison between unsmoothed and smoothed first (YPS) and second (YPPS) derivatives for a typical airfoil.



(b) Smoothed



(a) Unsmoothed

Figure 9. - Comparison between unsmoothed and smoothed square-root of curvature for a typical airfoil.

1. Report No. NASA TM-84666

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Computer Programs for Smoothing and Scaling Airfoil Coordinates

5. Report Date July 1983

6. Performing Organization Code 505-31-23-06

7. Author(s)

Harry L. Morgan, Jr.

8. Performing Organization Report No.

9. Performing Organization Name and Address

NASA Langley Research Center
Hampton, VA 23665

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered

Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report contains detailed descriptions of the theoretical methods and associated computer codes of a program to smooth and a program to scale arbitrary airfoil coordinates. The smoothing program utilizes both least-squares polynomial and least-squares cubic-spline techniques to smooth iteratively the second derivatives of the y-axis airfoil coordinates with respect to a transformed x-axis system which unwraps the airfoil and stretches the nose and trailing-edge regions. The corresponding smooth airfoil coordinates are then determined by solving a tridiagonal matrix of simultaneous cubic-spline equations relating the y-axis coordinates and their corresponding second derivatives. A technique for computing the camber and thickness distribution of the smoothed airfoil is also discussed.

The scaling program can then be used to scale the thickness distribution generated by the smoothing program to a specified maximum thickness which is then combined with the camber distribution to obtain the final scaled airfoil contour. Computer listings of the smoothing and scaling programs are included as appendices. A user-guide and sample input and output cases for both programs are also included as appendices. Both computer programs are available from COSMIC with identifications LAR-13132 for the airfoil smoothing program "AFSMO" and LAR-13133 for the airfoil scaling program "AFSCL".

17. Key Words (Suggested by Author(s))

Airfoil
Coordinate smoothing
Computer code
Scaling
Camber and thickness distribution

18. Distribution Statement

[redacted]

Subject Category 02

19. Security Classif. (of this report) Unclassified

20. Security Classif. (of this page) Unclassified

21. No. of Pages 186

22. Price*

Available: NASA's Industrial Applications Centers