COMPUTER PROGRAMS FOR SMOOTHING AND SCALING AIRFOIL COORDINATES

Harry L. Morgan, Jr.

July 1983
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 polynomial 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

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

**Subscripts:**

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>camber</td>
</tr>
<tr>
<td>$i$</td>
<td>iteration or element number</td>
</tr>
<tr>
<td>$l$</td>
<td>lower surface</td>
</tr>
<tr>
<td>$u$</td>
<td>upper surface</td>
</tr>
</tbody>
</table>
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

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

(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

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

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:

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

where \( \frac{\pi}{2} \leq \theta \leq \pi \). The constant \( K \) was determined by specifying that at \( \theta = \pi \) equals unity; therefore,

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

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

where \( 0 \leq \theta \leq \pi/2 \).

The first and second derivatives of equation (3) are

\[ \frac{dx}{d\theta} = \frac{K}{\cosh \left( \theta - \frac{\pi}{2} \right)} \] and

\[ \frac{d^2x}{d\theta^2} = -\frac{K \sinh \left( \theta - \frac{\pi}{2} \right)}{\cosh^2 \left( \theta - \frac{\pi}{2} \right)} , \] (7)

respectively, and of equation (5) are

\[ \frac{d\bar{x}}{d\theta} = K \sin (\theta) \] and

\[ \frac{d^2\bar{x}}{d\theta^2} = K \cos (\theta) \] ,

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

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

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

\[ \theta = \cos^{-1}(1 - \frac{x}{K}). \]  

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 \( \theta \) value using the following relationships:

\[ \frac{d\bar{y}}{d\bar{x}} = \bar{y}' \frac{1}{d\bar{x}/d\theta} \]  

\[ \frac{\partial^2\bar{y}}{\partial\bar{x}^2} = \bar{y}''(\frac{d\bar{x}}{d\theta}) - \bar{y}' \left( \frac{\partial^2\bar{x}}{\partial\theta^2} \right) \left( \frac{d\bar{x}}{d\theta} \right)^2 \]  

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 sur-
face function run from 0 to +π. 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 \( \ddot{y} \) 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

\[
g(\theta)_i = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i, \tag{14}
\]

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

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 \( \delta f_i \) 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:
\[ y_i = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \quad (17) \]
\[ y'_i = 3a_i h_i^2 + 2b_i h_i + c_i \quad (18) \]
and \[ y''_i = 6a_i h_i + 2b_i \quad (19) \]
where \( h_i = (\theta - \theta_i) \).

At the two end points of the \( i \)th interval, the \( \bar{y} \) coordinates are

\[ \bar{y}_i = d_i \quad (20) \]
at \( \theta = \theta_i \) and

\[ \bar{y}_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \quad (21) \]
at \( \theta = \theta_{i+1} \), and the second derivatives are

\[ \bar{y}''_i = 2b_i \text{ or } b_i = \frac{\bar{y}''_i}{2} \quad (22) \]
at \( \theta = \theta_i \) and

\[ \bar{y}''_{i+1} = 6a_i h_i + 2b_i \text{ or } a_i = \frac{\bar{y}''_{i+1} - \bar{y}''_i}{6h_i} \quad (23) \]
at \( \theta = \theta_{i+1} \).

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

\[ c_i = \left( \frac{\bar{y}_{i+1} - \bar{y}_i}{h_i} \right) - \left( \frac{\bar{y}''_{i+1} + 2\bar{y}''_i}{6} \right) h_i \quad (24) \]
At $\theta = \theta_i$, $\ddot{y}_i$ equals $c_i$ and from the previous interval

\[ \ddot{y}_i = 3a_{i-1} h_i^2 + 2b_{i-1} h_{i-1} + c_{i-1} \]  \hspace{1cm} (25)

where from a similar analysis,

\[ a_{i-1} = \frac{\dddot{y}_i - \dddot{y}_{i-1}}{6h_{i-1}} \]  \hspace{1cm} (26)

\[ b_{i-1} = \frac{\dddot{y}_{i-1}}{2} \]  \hspace{1cm} (27)

and

\[ c_{i-1} = \left( \frac{\dddot{y}_i - \dddot{y}_{i-1}}{h_{i-1}} \right) - \left( \frac{\dddot{y}_i + 2\dddot{y}_{i-1}}{6} \right) h_{i-1}. \]  \hspace{1cm} (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:

\[ \left( \frac{1}{h_{i-1}} \right) \ddot{y}_{i-1} - \left( \frac{1}{h_i} + \frac{1}{h_{i-1}} \right) \ddot{y}_i + \left( \frac{1}{h_i} \right) \ddot{y}_{i+1} = \left( \frac{h_{i-1}}{6} \right) \dddot{y}_{i-1} + \left( \frac{h_{i-1} + h_i}{3} \right) \dddot{y}_i + \left( \frac{h_i}{6} \right) \dddot{y}_{i+1} \]  \hspace{1cm} (29)

which represents a set of tridiagonal equations with $i = 2, 3, 4, \ldots, N - 1$. By specifying the desired $y$ 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

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

(30)

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

(31)

and

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

(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 \( \theta \)-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
where $x_C$ and $y_C$ are the coordinates of the camberline, $t$ is the local thickness, and $\phi$ 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

\[
\hat{x} = (x - x_c) \cos (\phi) + (y - y_c) \sin (\phi), \quad (37)
\]
\[
\hat{y} = (y - y_c) \cos (\phi) - (x - x_c) \sin (\phi). \quad (38)
\]

The differentials with respect to \( \bar{x} \) are

\[
d\hat{x}/d\bar{x} = \cos (\phi) + \sin (\phi) \frac{d\hat{y}}{d\bar{x}} \quad (39)
\]
\[
d\hat{y}/d\bar{x} = \cos (\phi) \frac{d\hat{y}}{d\bar{x}} - \sin (\phi) \quad (40)
\]

which combines to obtain the equation for the local slope

\[
\frac{d\hat{y}}{d\hat{x}} = \frac{\frac{d\hat{y}}{d\bar{x}}}{\frac{d\hat{x}}{d\bar{x}}} = \frac{\cos (\phi) \frac{d\hat{y}}{d\bar{x}} - \sin (\phi)}{\sin (\phi) \frac{d\hat{y}}{d\bar{x}} + \cos (\phi)}, \quad (41)
\]

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

\[
\phi = \tan^{-1} \left( \frac{\bar{y}_u - \bar{y}_l}{\bar{x}_u - \bar{x}_l} \right). \quad (42)
\]

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

\[
\left| \frac{d\hat{y}}{d\hat{x}} \right|_u = \left| \frac{d\hat{y}}{d\hat{x}} \right|_l \quad (43)
\]
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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>80-column title for input case</td>
</tr>
<tr>
<td>XINT</td>
<td>array containing ( \bar{x} ) interpolation values</td>
</tr>
<tr>
<td>X, Y</td>
<td>arrays containing reordered ( \bar{x} ) and ( \bar{y} ) coordinates</td>
</tr>
<tr>
<td>W</td>
<td>array containing input weighing factors</td>
</tr>
<tr>
<td>YSMO</td>
<td>array containing smoothed ( \bar{y} ) coordinates</td>
</tr>
<tr>
<td>YPS</td>
<td>array containing smoothed ( \bar{y}' ) values</td>
</tr>
<tr>
<td>YPPS</td>
<td>array containing smoothed ( \bar{y}'' ) values</td>
</tr>
<tr>
<td>THETA</td>
<td>array containing 0-transformation values</td>
</tr>
<tr>
<td>PI</td>
<td>value of ( \pi )</td>
</tr>
<tr>
<td>RAD</td>
<td>value of one radian ( \pi/180 )</td>
</tr>
<tr>
<td>CONS</td>
<td>value of constant ( K ) defined by equation (4)</td>
</tr>
<tr>
<td>JREAD</td>
<td>number of tape or file containing input data</td>
</tr>
<tr>
<td>JWRITE</td>
<td>number of tape or file containing output data</td>
</tr>
<tr>
<td>IPRINT</td>
<td>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</td>
</tr>
<tr>
<td>EPS</td>
<td>convergence criteria used during least-squares polynomial smoothing process in subroutine SMOXY</td>
</tr>
<tr>
<td>DF</td>
<td>standard deviation used during least-squares cubic spline smoothing process in subroutines SMOXY and CSDS</td>
</tr>
</tbody>
</table>
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

\[
\varepsilon_1 = \left| y_1 - y_s \right| \quad \text{and} \quad \varepsilon_2 = \left| y_2 - y_s \right| \quad (44)
\]

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

\[
y = w_1 y_2 + w_2 y_1 \quad , \quad (45)
\]

21
where \( w_1 = \frac{\Delta_1}{\Delta_1 + \Delta_2} \) and \( w_2 = \frac{\Delta_2}{\Delta_1 + \Delta_2} \) (46)

\[ \Delta_1 = \epsilon_1(x - x_{i-1}) \quad \text{and} \quad \Delta_2 = \epsilon_2(x_i - x). \] (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, sub-
routine 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 coordi-
nates are also nondimensionalized by the chord length and, at the
same time, the equivalent transformation $\theta$-values computed using
equations (10) and (11). If, instead of $x$ and $y$ coordinates, the $\tilde{y}$
coordinates, $\tilde{y}'$ values, or $\tilde{y}''$ values as a function of $\theta$ are input,
the equivalent $\tilde{x}$ values are computed using equations (3) and (5).

The following input quantities are defined in this subroutine:

ITER   allowable number of smoothing iterations
IPLT    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 $\tilde{y}$ coordinates at the lower
        surface trailing edge, the nose, and the
        upper surface trailing edge, respectively
NINT
CNEW
NP
NOSE
CHORD
IERR
TITLE
X
Y
W
THETA
YPS
YPPS
The following arrays and constants are used internally in this subroutine:
XL
YL
WL

number of input interpolation \( \bar{x} \) values
desired chord of interpolated \( \bar{y} \) coordinates (all \( \bar{y} \) coordinates computed in subroutines INTP are multiplied by CNEW)
number of elements in output arrays \( x, y, w, \theta, y' \), \( y'' \) and \( y''' \)
array index of nose point after reordering the coordinate
computed longest chord length
if not equal to zero, the last input case has been read or an error occurred during the calls to subroutine IBAD
input 8U-column title
output array containing reordered \( \bar{x} \) coordinates
output array containing reordered \( \bar{y} \) coordinates for IOP=0 or 1
output array containing reordered weighing factors
output array containing equivalent \( \theta \) values
output array containing \( \bar{y}' \) values for IOP=2
output array containing \( \bar{y}'' \) values for IOP=3
array containing input lower surface \( x \) coordinates if IOP=0 and \( \theta \)-values if IOP=0.
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
array containing input lower surface weighing factors
XU, YU, WU are same as XL, YL, and WL except for upper surface
NL is number of elements in XL, YL, and WL arrays
NU is number of elements in XU, YU, and WU arrays
ITRMAX is maximum number of allowable smoothing iterations
TOLR is allowable deviation between input and interpolated \( \tilde{y} \) coordinate in subroutine BADPT
NMAX is 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 \( \phi \) 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 is computed angle of longest chord and input x-axis
XNOSE, YNOSE are computed nose coordinate of longest chord
XTE, YTE are computed coordinates of trailing-edge bisector of longest chord

Subroutine BADPT

The function of subroutine BADPT is to identify and possibly to correct input \( \tilde{y} \) coordinates whose corresponding interpolated values exceed a specified tolerance. The user may execute a call to this subroutine by specifying a nonzero value for the parameter IBAD in...
The 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 equivalent of each input \( x \) coordinate is computed for use during the interpolation process. Then for each input \( y \) coordinate, a corresponding interpolated value is obtained using the weighted quadratic-equation method of subroutine INTER with input arrays loaded with the remaining \( y \) coordinates and \( \theta \) values. (Note that the input \( y \) coordinate itself is not loaded.) If the deviation between the input and interpolated \( y \) coordinate exceeds a specified tolerance, the interpolated \( y \) 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 \( x \) and \( y \) coordinates
- **ISURF**: if equal to 1, indicates upper surface coordinates input, and, if equal to 2, lower surface
- **TI**: work array containing all \( \theta \) values except value at desired interpolation point
- **YI**: work array containing all \( y \) coordinates except value at the desired interpolation point
- **YN**: temporary array containing interpolated \( y \) 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 \( \delta \) 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:

If

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

and

\[
\Delta_i = \left[ \bar{y}_N - \bar{y}_I \right]_i
\]
and if the sign or magnitude of $\Delta_i$ equals $\Delta_{i-1}$, then

$$\bar{y}_{Ni+1} = \frac{1}{2} (\bar{y}_I + \bar{y}_N)_i$$  \hspace{1cm} (49)$$

and, if not, the Newton-Raphson formula

$$\bar{y}_{Ni+1} = (\bar{y}_I)_{i-1} - \left(\frac{\Delta_{i-1}}{\Delta_i - \Delta_{i-1}}\right) \cdot \left[(\bar{y}_I)_i - (\bar{y}_I)_{i-1}\right]$$  \hspace{1cm} (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 $\text{EPS}$. 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 $\bar{y}''$. 

29
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. \(\hat{y}' = 0\)) in the final smoothed airfoil. This step is accomplished by checking each \(\theta\)-interval of the final airfoil for \(\theta\) locations where the first derivative spline equation (31) is equal to zero. This equation can be written as the quadratic equation

\[ a\theta^2 + b\theta + c = 0 \]  

with

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

\[ b = \left( \frac{\bar{y}_{i+1}' \theta_i - \bar{y}_{i}' \theta_{i+1}}{h_i} \right) \]

\[ c = \left( \frac{\bar{y}_{i}' \theta_{i+1}^2 - \bar{y}_{i+1}' \theta_i^2}{2h_i^2} \right) + \frac{h_i}{6} \left( \bar{y}_{i+1}' - \bar{y}_i' \right) - \left( \frac{\bar{y}_{i+1} - \bar{y}_i}{h_i} \right) \]
where \( h_i = \theta_{i+1} - \theta_i \). The real solutions to this equation which lie within the \( \theta \)-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 \((\text{IPT} = 0)\), 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

\[ \Delta x_{i+1} = \Delta x_i - \frac{f(\Delta x_i)}{f'(\Delta x_i)} \]  

(53)

where \( \Delta x \) 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 quantities 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 Ax 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 \( \tilde{y} \) 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 \( \theta \)-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 \( \tilde{y} \) 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

\[
\bar{y}_N = c_N \text{ for } i=N
\]
and \( \ddot{y}_i = \ddot{c}_i - \ddot{f}_i \dddot{y}_{i+1} \) for \( i = N-1, N-2, \ldots, 1 \) \hspace{1cm} (54)

where

\[
\ddot{d}_i = d_i - e_i \ddot{f}_{i-1} \\
\ddot{f}_i = f_i / \ddot{d}_i
\]

and \( \ddot{c}_i = \frac{c_i - e_i \ddot{c}_{i-1}}{\ddot{d}_i} \).

The tridiagonal terms from equation (29) are

\[
e_i = l / h_{i-1} \\
d_i = -1 / h_{i-1} - 1 / h_i \\
f_i = 1 / h_i
\]

and \( c_i = \left( \frac{h_{i-1}}{6} \right) \dddot{y}_{i-1} + \left( \frac{h_{i-1} + h_i}{3} \right) \dddot{y}_i + \left( \frac{h_i}{6} \right) \dddot{y}_{i+1} \)

At the ends the coefficient terms are

\[
d_1 = 1, f_1 = 0, c_1 = \dddot{y}_1
\]

and

\[
e_N = 0, d_N = 1, c_N = \dddot{y}_N
\]
The following is a description of the parameters in argument list for this subroutine:

- **X**: input array containing \( \theta \) values
- **YPP**: input array containing \( \ddot{y} \) values
- **NS**: index of start element
- **NE**: index of end element
- **Y**: output array containing \( \ddot{y} \) coordinates
- **YSTART**: desired \( \ddot{y} \) coordinate at start
- **YEND**: desired \( \ddot{y} \) 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 \( \ddot{y} \) 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 \( \theta \)-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 0 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
allowable deviation between corresponding upper and lower surface $\theta$ and $\bar{y}$ values in the nose region

if equal to zero, input airfoil is symmetric in nose region

does not apply (ISYM = 0)

arrays containing 7 consecutive values of $\theta$, $\bar{y}$, and $w$

array containing elements of symmetric least-squares matrix

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 $\theta$ 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:

<table>
<thead>
<tr>
<th>IPUNCH</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x-coordinate, smoothed y-coordinate, and weighting</td>
</tr>
<tr>
<td>2</td>
<td>θ-value, smoothed y coordinates, and weighting</td>
</tr>
<tr>
<td>3</td>
<td>θ-value, smoothed y', and weighting</td>
</tr>
<tr>
<td>4</td>
<td>θ-value, smoothed y'', and weighting</td>
</tr>
</tbody>
</table>

If IPUNCH equals 3 or 4, the 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 y coordinates, smoothed y', and smoothed y'' versus the θ values (IPLT=1) and to plot the input and smoothed y coordinates versus the input x coordinates (IPLT=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 $\tilde{y}$ and $x$ coordinates and the smoothed $\tilde{y}'$ values, additional values are interpolated at each degree of $\theta$ 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 $\theta$-transformation value (IPLOT=3). Prior to plotting the curvature, additional values are interpolated at each one-half degree of $\theta$ 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 loca-
tion 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 inter-
section 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 (IPL0T = 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

YPFU and YPPL temporary arrays containing input upper and lower surface $\bar{y}$ values

DYXU array containing $\frac{dy}{dx}$ 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 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 \( \frac{dy}{dx} \) and \( \frac{d^2y}{dx^2} \) 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 \( \bar{y} \), \( \frac{dy}{dx} \), \( \frac{d^2y}{dx^2} \), and the curvature at a specified value of \( \theta \) using the cubic-spline equations (30), (31), and (32). The following subroutine constants are used internally:

- **TI**
  - input \( \theta \) value
- **YI**
  - interpolated \( \bar{y} \)-coordinate
- **DYDX**
  - interpolated first derivative \( \frac{dy}{dx} \)
- **DY2DX**
  - interpolated second derivative \( \frac{d^2y}{dx^2} \)
- **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

\[ \sinh(x) = \frac{e^x - e^{-x}}{2} \quad (58) \]

and \[ \cosh(x) = \frac{e^x + e^{-x}}{2} \quad (59) \]
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 \( \theta \) 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 $\phi$
- **THETA** array containing computed $\theta$ 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
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
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

\[ y = ax^2 + bx + c \]  

(60)
to each end point and its two adjacent points and then differentiating to determine the second derivative which is

\[ \frac{d^2 y}{dx^2} = 2a \]  

(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

\[ e_i = h_{i-1}/6 \]

\[ d_i = \frac{h_{i-1} + h_i}{3} \]  

(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 \( \frac{d^2y}{dx^2} \)
- **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 \( \theta \)-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 \( \theta \)-transformation function as

\[ \theta = \pm \pi \bar{x} \]  \hspace{1cm} (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 converged in 300 iterations, the resultant coordinates can be written on the output file TAPE1 in the form of either x and y coordinates or $\theta$ 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 \( \tilde{y} \) coordinates and the corresponding \( \theta \)-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 \( \tilde{y} \) coordinates that are often revealed during the initial run of a new airfoil. The stretching effect of the \( \theta \)-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 \( \tilde{y}' \) (IOP = 3) or second \( \tilde{y}'' \) (IOP = 4) derivatives as a function of the \( \theta \)-transformation value. The corresponding \( \tilde{y} \) 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 \( \tilde{y} \) 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 \( \tilde{y} \) 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 $\gamma$ 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 AFSCCL 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.
LISTING OF DECK: AIRSMO

CARD NO.

1  PROGRAM AIRSMO(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1) AS 1

2  THIS PROGRAM PRESENTS A TECHNIQUE FOR SMOOTHING AIRFOIL

3  COORDINATES USING LEAST SQUARES POLYNOMIAL AND CUBIC SPLINE

4  METHODS

5  CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

6  DIMENSION TITLE(8), XINT(100), X(200), Y(200), W(200), YSMO(200),

7  1YPS(200), YPPS(202), THEATA(202)

8  COMMON /HLM/ DUMMY(2000)

9  COMMON /SMY/ DUMMY(2130)

10 COMMON /BLK1/ PI,PI2,RAD,CONS

11 COMMON /INOUT/ JREAD,WRITE,IPRINT

12 SINH(X) = 0.5*(EXP(X)-EXP(-X))

13 INITIALIZE PROGRAM CONSTANTS

14 PI = ACOS(-1.)

15 PI2 = PI/2.

16 RAD = 100./PI

17 CONS = 0.5*(1.+ATAN(SINH(PI2)))

18 JREAD = 5

19 WRITE = 6

20 IPRINT = 1

21 EPS = 1.E-6

22 DF = 1.E-6

23 REWIND 1

24 INITIALIZE PLOTTING DEVICE

25 CALL PSFUDO

26 CALL LEROY

27 READ INPUT DATA

PAGE 1
LISTING OF DECK: AIFSMD

CARD NO.

41
1
C
CALL INPUT (TITLE,ITER,IPLOT,IPUNCH,IOR,ICAMTK,INTE,YLTE,YNOSE,YUT
1E,NINT,XINT,CNEW,NP,X,Y,W,THERA,YPS,YPPS,NOSE,CHORD,IERR)
AS 41
IF (IERR-1) 2,3
AS 44

45
C
SMOOTH AIRFOIL COORDINATES
AS 46

2
CALL SMOXY (THERA,X,Y,W,YSMD,YPS,YPSS,NP,NOSE,YLTE,YNOSE,YUTE,EPS)
AS 48
IDF,ITER,TITLE,IPUS,IERR)
AS 49
IF (IERR+NE.0) GO TO 1
AS 50

50
C
PUNCH OUTPUT DATA
AS 51
C
IF (IPUNCH,GE.1.AND.IPUNCH,LE.4) CALL PCARD (IPUNCH,X,Y,W,THERA,YS
1MO,YPS,YPPS,NOSE,NP,CHORD,TITLE)
AS 54

55
C
PLOT SMOOTHED AND UNSMOOTHED Y/C, SMOOTHED YPS, AND SMOOTHED
C
YPSS VERSUS THETA. ALSO PLOT SMOOTHED AND UNSMOOTHED Y/C VERSUS
C
Y/C
AS 58

60
C
IF (IPLOT,OE.0.OR.IPL0T,OE.4) GO TO 4
AS 61
IF (IPLOT,OE.3) GO TO 3
AS 62
C
CALL PLOTAF (THERA,Y,YSMD,YPS,YPPS,NP,TITLF,IPLOT)
AS 64

65
C
IF (IPLOT,OE.5.OR.IPL0T,OE.1) GO TO 4
AS 66
IF (IPLOT,OE.6.OR.IPL0T,OE.7) GO TO 3
AS 67
IF (IPLOT,OE.10) GO TO 3
AS 68
GO TO 4
AS 69

70
C
PLOT SMOOTHED CURVATURE VERSUS THETA
AS 70
C
CALL PLOTCK (THERA,YSMD,YPS,YPPS,NP,TITLE)
AS 71

75
4
KPL0T=0
AS 75
IF (IPLOT,OE.4.OR.IPL0T,OE.8) KPL0T=1
AS 76

C
COMPUTE THICKNESS AND CAMBER DISTRIBUTION
AS 77
C
IF (ICAMTK,OE.1) CALL CAMTK (THERA,YSMD,YPS,NOSE,NP,EPS,KPL0T,IPU
AS 78

AS 79

AS 80
LISTING OF DECK: AIRSMO

CARD NO.

81
C
C
C

INTERPOLATE NEW COORDINATES

85
C
C
C

IF (INTR .GT. 0) CALL INTP (THETA, X, YSMO, YPPS, NP, NOSE, CHORD, TITLE, NI
1INT, XINT, CNEW, INTPUNCH)

C
C
C

RETURN AND READ NEXT CASE

90
C
C
C

GO TO 1

C
C
C

FINALIZE PLOTTING DEVICE

95
C
C
C

CALL CALPLT (0, 0, 999)
WRITE (JWRITE, 6)
END FILE 1
REWIND 1
STOP

100
C
C

FORMAT (1H1///48X, 38H -- THE LAST CASE HAS BEEN PROCESSED --)
END

AS 81
AS 82
AS 83
AS 84
AS 85
AS 86
AS 87
AS 88
AS 89
AS 90
AS 91
AS 92
AS 93
AS 94
AS 95
AS 96
AS 97
AS 98
AS 99
AS 100
AS 101-
SUBROUTINE INTER (XINT,YINT,N,X,Y,JSTART,JEND,ICD)

INTERPOLATION ROUTINE

ROUTINE SOURCE -- NORTH AMERICAN ROCKWELL L. A. DIVISION 1973

ICD=0 WEIGHTING METHOD USED
ICD=1 LINEAR INTERPOLATION

DIMENSION X(N), Y(N)
CHECK TO SEE IF XINT IS OUTSIDE BOUNDS OF X-ARRAY

IF (JSTART.EQ.N) GO TO 12
CHECK TO SEE IF X ARRAY IS INCREASING OR DECREASING
SGN=1.
IF (X(N).LT.X(JSTART)) SGN=-1.
DI=SGN*(XINT-X(N))

IF (D1.GE.0.0) GO TO 12
DI=SGN*(XINT-X(JSTART))
IF (D1.LE.0.0) GO TO 13
IF (ICD.EQ.1) GO TO 14
WEIGHTING METHOD REQUIRES AT LEAST 4 VALUES IN X AND Y ARRAYS
IF (N.LE.4) GO TO 14

WEIGHTING METHOD

DETERMINE X-ARRAY INDICES FOR TWO POINTS FORWARD (J,L) AND TWO POINTS AFT (K,M) OF XINT

DO 1 L=JSTART,N
   J=L
   DI=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
LISTING OF DECK: INTER

CARD NO.

41 4  GO TO 6
   JJ=2
   J=N-1
   GO TO 6

45 5  JJ=1
   J=3
   K=J-1
   M=J-2
   L=J+1

50 C  INTERPOLATE A YINT VALUE (YSL) BY FITTING A STRAIGHT LINE 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)+(100-D)*Y(K)

55 C  INTERPOLATE A YINT VALUE (YP1) BY FITTING A QUADRATIC BETWEEN M, K, AND L
   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)

60 C  INTERPOLATE A YINT VALUE (YP2) BY FITTING A QUADRATIC BETWEEN 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)

70 C  IF (JJ-2) 789
   YP2=YP1
   D=(XINT-X(1))/(X(2)-X(1))
   YSL=D*Y(2)+(100-D)*Y(1)
   GO TO 9

75 B  YP1=YP2
   D=(XINT-X(N-1))/(X(N)-X(N-1))
   YSL=D*Y(N)+(100-D)*Y(N-1)

80 C  COMPUTE DEVIATION BETWEEN LINEAR AND QUADRATIC YINT VALUES
   DEV1=ABS(YP1-YSL)
LISTING OF DECK: INTER

CARD NO.

81  DEV2=ABS(YP2-YSL)
    IF (DEV1+DEV2) 10,10,11
10   YINT=YSL
    RETURN

85  C  COMPUTE WEIGHTING FACTORS
11   WT2=(DEV1*D)/(DEV1*D+(1.0-D)*DEV2)
    WT1=1.0-WT2
    C  COMPUTE FINAL YINT
12   YINT=WT2*YP2+WT1*YP1
    RETURN
13   YINT=Y(JSTART)
    RETURN

90  C  LINEAR INTERPOLATION METHOD
95  C
14   DO 15 L=JSTART,N
15   J=L
    DI=SGN*(X(J)-XINT)
    IF (DI) 15,2,16
16   JEND=J
    YINT=Y(J-1)+(Y(J)-Y(J-1))*((XINT-X(J-1))/(Y(J)-X(J-1))
    RETURN
    END
LISTING OF DECK: INPUT

CARD NO.

1

SUBROUTINE INPUT (TITLE,ITER,IPLLOT,IPUNCH,ITP,ICAMTK,INTR,YLTE,YNO IU 1
1SE,YUTE,HINT,XINT,CNEM,NP,X,Y,W,THETA,YPS,YPPS,NOSSE,CHORD,IERE) IU 2
C
C ROUTINE TO READ INPUT DATA FOR AIRFOIL SMOOTHING PROGRAM IU 3
C
C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AA9 1982 IU 4
C
C*************************************************************************
C* DESCRIPTION OF INPUT CARDS FOR SMOOTHING PROGRAM IU 8
C* CARD NUMBER DESCRIPTION
C*
C* 1 FORMAT(8A10) IU 9
C* TITLE CARD IU 10
C* 2 FORMAT(8F10.0) IU 11
C* ITER - MAXIMUM NUMBER OF SMOOTHING ITERATIONS IU 12
C* I PLOT - PLOTTING OPTION IU 13
C* 0 - NO PLOTS IU 14
C* 1 - PLOT SMOOTHED AND UNSMOOTHED Y/C, SMOOTHED IU 15
C* YPS, AND SMOOTHED YPPS VS THETA IU 16
C* 2 - PLOT SMOOTHED AND UNSMOOTHED Y/C VS X/C IU 17
C* 3 - PLOT SMOOTHED CURVATURE VS THETA IU 18
C* 4 - PLOT CAMBER AND THICKNESS DISTRIBUTION IU 19
C* 5 - PLOT OPTIONS 1 AND 2 IU 20
C* 6 - PLOT OPTIONS 1 AND 3 IU 21
C* 7 - PLOT OPTIONS 1, 2, AND 3 IU 22
C* 8 - PLOT OPTIONS 1 AND 4 IU 23
C* 9 - PLOT OPTIONS 1, 2, AND 4 IU 24
C* 10 - PLOT OPTIONS 1, 2, 3 AND 4 IU 25
C* IPUNCH - PUNCH OUTPUT OPTION IU 26
C* 0 - NO PUNCH OUTPUT IU 27
C* 1 - SMOOTHED (X,Y,W) PUNCHED IU 28
C* 2 - SMOOTHED (THETA,YPS,W) PUNCHED IU 29
C* 3 - SMOOTHED (THETA,YPPS,W) PUNCHED (YNLE, IU 30
C* YNDOSE, AND YUTE ALSO PUNCHED) IU 31
C* 4 - SMOOTHED (THETA,YPPS,W) PUNCHED (YNLE, IU 32
C* YNDOSE, AND YUTE ALSO PUNCHED) IU 33

C*

C 62
LISTING OF DECK: INPUT

CARD NO.

41
C* 5 - THICKNESS AND CAMBER DISTRIBUTION (X/C, Y/C, T/C/2, AND SLOPE) PUNCHEO
C* 6 - INTERPOLATED COORDINATES PUNCHEO

45
C* IOP - INPUT DATA OPTION
C* 0 - (X,Y,W) INPUT
C* 1 - (THETA,Y/C,W) INPUT
C* 2 - (THETA,YPS,W) INPUT
C* 3 - (THETA,YPS,W) INPUT

50
C* ICAMTK - THICKNESS AND CAMBER DISTRIBUTION OPTION
C* 0 - DO NOT COMPUTE THICKNESS AND CAMBER
C* 1 - COMPUTE THICKNESS AND CAMBER

55
C* IBAD - BAD COORDINATE CHECK OPTION
C* 0 - DO NOT CHECK FOR BAD COORDINATES
C* 1 - CHECK FOR BAD COORDINATES

60
C* ITPN - INPUT COORDINATE TRANSLATION AND ROTATION OPTION
C* 0 - DO NOT TRANSLATE AND ROTATE
C* 1 - TRANSLATE AND ROTATE SO THAT X-AXIS CORRESPONDS TO THE LONGEST CHORDLINE

65
C* INTR - COORDINATE INTERPOLATION OPTION
C* 0 - NO INTERPOLATION DESIRED
C* 1 - INTERPOLATE NEW COORDINATES USING STANDARD 57 VALUES (0.0 . GE. Y/C .LE. 1.0)

70
C* 3 FORMAT(10,0)
C* NU - NUMBER OF UPPER SURFACE INPUT COORDINATES

75
C* 4 FORMAT(3F10.0)
C* XU,YU,WU - UPPER SURFACE INPUT COORDINATES AND WEIGHTING
C* (NU CARDS ARE INPUT)
C* IF IOP=0, XU=X AND YU=Y COORDINATES
C* IF IOP=1, XU=THETA AND YU=Y/C
C* IF IOP=2, XU=THETA AND YU=YPS
C* IF IOP=3, XU=THETA AND YU=YPS

80
C* 5 FORMAT(10,0)
C* NL - NUMBER OF LOWER SURFACE INPUT COORDINATES

85
C*
LISTING OF DECK: INPUT

CARD NO.

81  C* 6 FORMAT(3F10.0)  
    XL,YL, WL - LOWER SURFACE INPUT COORDINATES AND WEIGHTING  * IU 81 
    (NL CARDS ARE INPUT)  * IU 82 
    IF IOP=0, XL=X AND YL=Y COORDINATES  * IU 83 
    IF IOP=1, XL=THETA AND YL=Y/C  * IU 84 
    IF IOP=2, XL=THETA AND YL=YP  * IU 85 
    IF IOP=3, XL=THETA AND YL=YP  * IU 86 
    FOR ALL IOP, WL=WEIGHTING FACTOR  * IU 87 
    C* 7 FORMAT(3F10.0)  
    XLT,E,NSE,YSE - LOWER SURFACE TRAILING-EDGE, NOSE, 
    AND UPPER SURFACE TRAILING-EDGE  * IU 88 
    X/C COORDINATES  * IU 89 
    C* 8 FORMAT(F10.0)  
    NINT - NUMBER OF INTERPOLATION X/C COORDINATES  * IU 90 
    C* 9 FORMAT(8F10.0)  
    SKIP IF INTR=0 OR 1  * IU 91 
    XINT - INTERPOLATION X/C COORDINATES (NINT VALUES INPUT)  * IU 92 
    C* 10 FORMAT(F10.0)  
    CNEW - DESIRED CHORD LENGTH OF INTERPOLATED COORDINATES  * IU 93 
    C* RESTRICTIONS:  
    * IU 94 
    IT= NOT GREATER THAN 300  * IU 95 
    NU OR NL NOT GREATER THAN 100  * IU 96 
    NINT NOT GREATER THAN 100  * IU 97 
    C* DIMENSION VAR(8), TITLE(8), XINT(1), X(1), Y(1), W(1), THETA(1), Y 
    1PS(1), YPPS(1)  
    C COMMON /SMY/ YU(100), YU(100), WU(100), XL(100), YL(100), WL(100)  
    C COMMON /BLK1/ PI,PI2, PI3, RAD, CONS  
    C COMMON /INOUT/ JREAD, JWRITE, IPRINT  
    C
LISTING OF DECK: INPUT

CARD NO.

121  C  SINH(X)=(EXP(X)-EXP(-X))/2
   C  INITIALIZE ROUTINE CONSTANTS

125  C  ITRMAX=300
     NMAX=100
     TOLP=1.E-2
     IERR=0

130  C  READ AND PRINT INPUT DATA
     C  READ AND WRITE TITLE
     READ (JREAD,27) TITLE
     IF (EOF(JREAD)) 29,1
     WRITE (JWRITE,29) TITLE
     C  READ AND WRITE OPTIONS
     READ (JREAD,29) VAR
     ITER=IFIX(VAR(1))
     IPRINT=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))

145  C  CHECK LIMITS OF OPTIONS
     IF (ITER.GT.ITRMAX) TTF=ITRMAX
     IF (IPRINT.GT.10) IPRINT=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 (INTR.GT.2) INT=0

155  C  WRITE (JWRITE,30) ITER,ITPLOT,IPUNCH,IOP,ICAMTK,IBAD,ITRN,INTR
     READ (JREAD,29) VAR(1)
     NU=IFIX(VAR(1))
     IF (NU.GT.NMAX) GO TO 22

160  C  READ AND WRITE NUMBER OF UPPER SURFACE INPUT POINTS
     WRITE (JWRITE,31) NU
LISTING OF DECK: INPUT

CARD NO.

161 C READ AND WRITE UPPER SURFACE INPUT POINTS AND WEIGHTING IU 161
READ (JREAD,32) (XU(I),YU(I),WU(I),I=1,NU)
DO 2 I=1,NU
IU 162
IF (WU(I).LT.1.0) WU(I)=1.0
IU 163
2 CONTINUE
IU 165
IF (IOP.EQ.0) WRITE (JWRITE,33) (XU(I),I=1,NU)
IU 166
IF (IOP.NE.0) WRITE (JWRITE,34) (XU(I),I=1,NU)
IU 167
IF (IOP.LT.2) WRITE (JWRITE,35) (YU(I),I=1,NU)
IU 168
IF (IOP.EQ.2) WRITE (JWRITE,36) (YU(I),I=1,NU)
IU 169
IF (IOP.EQ.3) WRITE (JWRITE,37) (YU(I),I=1,NU)
IU 170
WRITE (JWRITE,38) (WU(I),I=1,NU)
IU 171
C READ AND WRITE NUMBER OF LOWER SURFACE INPUT POINTS
IU 172
READ (JREAD,29) VAR(1)
IU 173
NL=IFIX(VAR(1))
IU 174
175 IF (NL.GT.NMAX) GO TO 22
IU 175
WRITE (JWRITE,39) NL
IU 176
C READ AND WRITE LOWER SURFACE INPUT POINTS AND WEIGHTING
IU 177
READ (JREAD,32) (XL(I),YL(I),WL(I),I=1,NL)
IU 178
DO 3 I=1,NL
IU 179
3 CONTINUE
IU 180
IF (WL(I).LT.1.0) WL(I)=1.0
IU 181
180 3 CONTINUE
IU 182
IF (IOP.EQ.0) WRITE (JWRITE,40) (XL(I),I=1,NL)
IU 183
IF (IOP.NE.0) WRITE (JWRITE,41) (XL(I),I=1,NL)
IU 184
IF (IOP.LT.2) WRITE (JWRITE,42) (YL(I),I=1,NL)
IU 185
IF (IOP.EQ.2) WRITE (JWRITE,43) (YL(I),I=1,NL)
IU 186
IF (IOP.EQ.3) WRITE (JWRITE,44) (YL(I),I=1,NL)
IU 187
WRITE (JWRITE,45) (WL(I),I=1,NL)
IU 188
C READ AND WRITE TRAILING-EDGE COORDINATES
IU 189
190 IF (IOP.LE.1) GO TO 4
IU 190
READ (JREAD,29) YLTE,YNOSE,YUTE
IU 191
WRITE (JWRITE,46) YLTE,YNOSE,YUTE
IU 192
C READ AND WRITE NUMBER OF INTERPOLATION COORDINATES
IU 193
4 IF (INTR.EQ.0) GO TO 6
IU 194
IF (INTR.NE.2) GO TO 5
IU 195
195 READ (JREAD,29) VAR(1)
IU 196
NINT=IFIX(VAR(1))
IU 197
IF (NINT.GT.NMAX) GO TO 24
IU 198
WRITE (JWRITE,47) NINT
IU 199
C READ AND WRITE INTERPOLATION COORDINATES
IU 200
200 READ (JREAD,29) (XINT(I),I=1,NINT)
LISTING OF DECK1 INPUT

CARD NO.

201 C WRITE (JWRITE,48) (XINT(I),I=1,NINT) READ AND WRITE NEW CHORD OF INTERPOLATED COORDINATES IU 201 5 READ (JREAD,29) CNW WRITE (JWRITE,49) CNW IU 202 205 C CHECK UPPER SURFACE COORDINATES FOR BAD POINTS IU 203 6 IF (IDP.NE.0) GO TO 7 IF (IBAD.EQ.1) CALL BADPT (XU,YU,NU,TOLR,1,IERR) IU 204 IU 205 IU 206 IU 207 IU 208 IU 209 IU 210 IU 211 IU 212 IU 213 IU 214 IU 215 IU 216 IU 217 IU 218 IU 219 IU 220 IU 221 IU 222 IU 223 IU 224 IU 225 IU 226 IU 227 IU 228 IU 229 IU 230 DO 111=1,NL IU 231 NP-NP+1 IU 232 IU 233 IU 234 IU 235 IU 236 IU 237 IU 238 IU 239 IU 240 230 NP=0 J=NL+I-1 W(NP)=WL(J) DELTA=(XL(J)-XL(1))/CHORD IF (DELTA.LE.CONS) GO TO 9 IF (DELTAN(TAN(DELTA/CONS)-1.,1.)) THETA(NP)=PI2-ALOG(DELTA+SQR(Delta*DELTA+1.)) GO TO 10 IU 240 IU 241
LISTING OF DECK: INPUT

CARD NO.

241  10 X(NP)=XL(J)/CHORD

11  Y(NP)=YL(J)/CHORD

C

NOSE=NP

245  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,NP

NP=NP+1

W(NP)=WU(I)

250  DELTA=(XU(I)-XU(1))/CHORD

IF (DELTA.LECONS) GO TO 12

DELTA=TAN(DELTA/CONS)

THETA(NP)=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))

GO TO 13

255  IF (IOP.EQ.1) YU(NP)=YL(J)

12  THETA(NP)=ACOS(1.-DELTA/CONS)

13  X(NP)=XU(I)/CHORD

14  Y(NP)=YU(I)/CHORD

C

GO TO 20

C

260  IF IOP=1, 2, OR 3, COMPUTE X/C FROM INPUT THETA

C

265  COMPUTE X/C FOR LOWER SURFACE

15  CHORD=1.0

NP=0

DO 17 I=1,NL

NP=NP+1

J=NL+I-1

W(NP)=WJ(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)

270  THETA(NP)=XL(J)/RAD

DELTA=ABS(THETA(NP))

IF (DELTA.GT.PI2) GO TO 16

275  XL(J)=CONS*(1.-COS(DELTA))

16  GO TO 17

280  DO 19 I=2,NU

C

17  X(NP)=XL(J)

NOSE=NP

C

COMPUTE X/C FOR UPPER SURFACE

XU(1)=XL(1)

DO 19 I=2,NU
LISTING OF DECK: INPUT

CARD NO.

281 NP=NP+1
W(NP)=WU(I)
IF (IOP, EQ, 1) Y(NP)=YU(I)
IF (IOP, EQ, 2) YPS(NP)=YU(I)
285 IF (IOP, EQ, 3) YPPS(NP)=YU(I)
THETA(NP)=YU(I)/RAD
DELTA=ABS(THETA(NP))
IF (DELTA.GT.PI2) GO TO 18
XU(I)=CONS*(1.-COS(DELTA))
290 GO TO 19
18 XU(I)=CONS*(ATAN(SINH(DELTA-PI2))+1.)
19 X(NP)=XU(I)
C C PRINT SUMMARY OF INPUT DATA
295 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)
300 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
305 C C PRINT ERROR MESSAGES
22 NN=IFIX(VAR(11))
WRITE (JWRITE,55) NN
GO TO 25
310 23 WRITE (JWRITE,56) IOP
GO TO 25
24 WRITE (JWRITE,57) NINT
C C NO ADDITIONAL INPUT DATA
25 IERR=2
C C RETURN TO CALLING PROGRAM
LISTING OF DECK: INPUT

CARD NO.

321  26  RETURN  IU 321
C
27  FORMAT (8A10)  IU 322
29  FORMAT (1H1,5X,14H--INPUT DATA--/5X,7HTITLE--/2X,8A10)  IU 323
29  FORMAT (8F10.5)  IU 324
30  FORMAT (/5X,6HITER =I4,3X,7HPILOT =I3,3X,8HPUNCH =I3,3X,5HIOP
1H,I3,3X,8HICANTK =I3,3X,6HIBAD =I3,3X,6HTRN =I3,3X,6HINT N =I3
2)  IU 325
30  FORMAT (/5X,4HNU =I4)  IU 326
31  FORMAT (/5X,4HNU =I4)  IU 327
32  FORMAT (3F10.5)  IU 331
33  FORMAT (/5X,3HXU =8E15.6/(8X,8E15.6))  IU 332
34  FORMAT (/5X,3HTU =8E15.6/(8X,8E15.6))  IU 333
35  FORMAT (/5X,3HYU =8E15.6/(8X,8E15.6))  IU 334
36  FORMAT (/4X,4HYPU =8E15.6/(8X,8E15.6))  IU 335
37  FORMAT (/3X,5HYPPU =8E15.6/(8X,8E15.6))  IU 336
38  FORMAT (/5X,5HNU =8E15.6/(8X,8E15.6))  IU 337
39  FORMAT (/5X,5HNL =I4)  IU 338
40  FORMAT (/5X,5HNL =I4)  IU 339
41  FORMAT (/5X,5HTL =8E15.6/(8X,8E15.6))  IU 340
42  FORMAT (/5X,5HYL =8E15.6/(8X,8E15.6))  IU 341
43  FORMAT (/4X,4HYPL =8E15.6/(8X,8E15.6))  IU 342
44  FORMAT (/3X,5HYPL =8E15.6/(8X,8E15.6))  IU 343
45  FORMAT (/5X,5HNL =8E15.6/(8X,8E15.6))  IU 344
46  FORMAT (/5X,5HNL =8E15.6/(8X,8E15.6))  IU 345
47  FORMAT (/5X,5HNINT =I4)  IU 346
48  FORMAT (/5X,5HNINT =I4)  IU 347
49  FORMAT (/5X,5HCNEW =F10.3)  IU 348
50  FORMAT (1H1,29X,25H--SUMMARY OF INPUT DATA--/5X,9HTITLE--
1/5X,1HI,10X,3HX/C,12X,3HY/C,12X,5HTHETA,10X,3HYP,12X,4HYPS,14X,1
IU 349
2H)  IU 350
51  FORMAT (110,2F15.6,F15.230X,F15.2)  IU 351
52  FORMAT (110,2F15.6,F15.230X,F15.2)  IU 352
53  FORMAT (110,2F15.6,F15.230X,F15.2)  IU 353
54  FORMAT (110,2F15.6,F15.230X,F15.2)  IU 354
55  FORMAT (/5X,2BINPUT CARD ERROR NU OR WL =I4)  IU 355
56  FORMAT (/5X,2BINPUT CARD ERROR IDP =I4)  IU 356
57  FORMAT (/5X,2BINPUT CARD ERROR NINT =I5)  IU 357
END  IU 358-
SUBROUTINE TRNSRT (XU,YU,WU,NL,XL,YL,NL,TITLE)

ROUTINE TO TRANSLATE AND ROTATE THE INPUT AIRFOIL COORDINATES SO THAT THE X-AXIS CORRESPONDS TO THE LONGEST CHORDLINE

CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

DIMENSION XU(1), YU(1), WU(1), XL(1), YL(1), WL(1), TITLE(8)

COMMON /HM/ X(200), Y(200), W(200)

COMMON /BLK1/ PI, PI2, RAD, CONS

COMMON /INOUT/ JREAD, JWRITE, IPRT

PRINT INPUT COORDINATES

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(I)

1 CONTINUE

WRITE (JWRITE,15) I, XL(I), YL(I)

2 CONTINUE

LOAD LOWER SURFACE COORDINATES

N=O

DO 2 I=1, NL

J=NL+1-I

N=N+1

W(N)=WL(J)

X(N)=XL(J)

Y(N)=YL(J)

J=1

IF (XL(1).EQ.XU(1) .AND. YL(1).EQ.YU(1)) J=2

LOAD UPPER SURFACE COORDINATES
LISTING OF DECK: TRNSRT

CARD NO.

41
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(I)+X(N))
YTE=0.5*(Y(I)+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) .LT.0.5,4
5
CHORD=DIST
NOSE=I
XNOSE=X(I)
YNOSE=Y(I)
5
CONTINUE
C
TRANSLATE AND ROTATE AIRFOIL
C
IF (CHORD.LT.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)
8
X(I)=(DIST-XNOSE)*COSA+(Y(I)-YNOSE)*SINA
Y(I)=(Y(I)-YNOSE)*COSA-(DIST-XNOSE)*SINA
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)
9
10
PAGE 2
LISTING OF DECK: TRNLSRT

CARD NO.

81  9  YL(I)=Y(J)
     NL=NOSE
     DO 10 I=NOSE,N
          J=I+1-NOSE
     10  WU(J)=W(I)
          XU(J)=X(I)
          YU(J)=Y(I)
          NU=J

90  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(I)
    11
    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)
    CONTINUE
    WRITE (JWRITE,12) XNOSE,YNOSE,ANGLE
    RETURN

C

105  12  FORMAT (/5X,7HXNOSE =F15.6,F5X,7HYNOSE =F15.6,F5X,7HANGLE =F8.3)  TR 104
13  FORMAT (1HI,32X,21H--INPUT COORDINATES--/5X,7HTITLE--2X,8A10//9X
     1,1HI,11X,2HXU,13X,2HYU,13X,2HXL,13X,2HYL)
14  FORMAT (5X,15,4F15.6)
15  FORMAT (5X,15,30X,2F15.6)
16  FORMAT (1HI,21X,38H--TRANSLATED AND Rotated COORDINATES--/5X,7HTI
     1,1LE--2X,8A10//9X,1HI,11X,2HXU,13X,2HYU,13X,2HXL,13X,2HYL)
110  END

73
SUBROUTINE BADPT (X,Y,NP,TOLR,ISURF,IERR)  
ROUTINE TO EDIT BAD POINTS FROM X AND Y INPUT COORDINATES  
CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAAB 1982  
DIMENSION X(1), Y(1), SURF(2)  
COMMON /HLM/ TI(100), YI(100), YN(100), THETA(100)  
COMMON /BLK1/ PI, PI2, RAD, CONS  
COMMON /INOUT/ JREAD, JWRITE, IPRINT  
DATA SURF(1)/SHUPPER/, SURF(2)/SHLOWER/  
IF TOLERANCE IS ZERO OR NEGATIVE RETURN  
IERR=0  
IF (TOLR.LE.0.0) RETURN  
COMPUTE LOCAL CHORD  
CHORD=X(NP)-X(1)  
INITIALIZE ITERATION PARAMETERS  
ICD=0  
IPTP=0  
N1=NPI-1  
NMAX=0  
TOLC=TOLR*CHORD  
COMPUTE THETA EQUIVALENT OF X  
DO 2 I=1,NP  
DELTA=(X(I)-X(1))/CHORD  
IF (DELTA.LECONS) GO TO 1  
DELTA=TAN(DELTA/CONS-1.)  
THETA(I)=PI2+ALOG(DELTA+SORT(DELTA*DELTA+1.))
LISTING OF DECK: BADPT

CARD NO.

41
  1  GO TO 2
  2  CONTINUE
  3  LOOP TO SEARCH FOR BAD POINTS
  4  LOAD Ti AND Yi ARRAY - OMIT THE I(TH) INPUT DATA POINT
  5  DO 5 I=2,N1
      K=0
  6  CONTINUE
  7  IF (IPT,EQ.0) RETURN
  8  WRITE (JWRITE,9) SURF(ISURF),TOLC
  9  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
 10  CONTINUE

50
  1  NMAX=NMAX+1
  2  JSTART=I
  3  K=0
  4  CONTINUE
  5  IF (IPT,EQ.0) RETURN
  6  WRITE (JWRITE,9) SURF(ISURF),TOLC
  7  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
  8  CONTINUE

55
  1  IF (I=EQ.J) GO TO 4
  2  TI(K]=T(J)
  3  YI(K]=Y(J)
  4  CONTINUE
  5  IPT=I
  6  ERRHAX=ERRHAX+1
  7  CONTINUE
  8  IF (IPT,EQ.0) RETURN
  9  WRITE (JWRITE,9) SURF(ISURF),TOLC
 10  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
 11  CONTINUE

60
  1  IF (ERR.EQ.0) RETURN
  2  IF (ERR.EQ.0) WRITE (JWRITE,9) SURF(ISURF),TOLC
  3  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
  4  CONTINUE

65
  1  IF (IPT,EQ.0) RETURN
  2  ERRMIN=ERRMIN+1
  3  IF (ERR.EQ.0) WRITE (JWRITE,9) SURF(ISURF),TOLC
  4  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
  5  CONTINUE

70
  1  IF (ERR.EQ.0) WRITE (JWRITE,9) SURF(ISURF),TOLC
  2  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
  3  CONTINUE

75
  1  IF (NMAX.EQ.1) WRITE (JWRITE,9) SURF(ISURF),TOLC
  2  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
  3  CONTINUE

80
  1  IF (ERR.EQ.0) WRITE (JWRITE,9) SURF(ISURF),TOLC
  2  WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)
  3  CONTINUE

END
LISTING OF DECK: BADPT

CARD NO.

81 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

85 C IPTP=IPT
     IF (NMAX.GE.NP) RETURN

C RETURN TO START OF LOOP AND SEARCH FOR NEXT BAD POINT

90 C GO TO 3

C WARNING MESSAGE PRINT STATEMENT

8 C WRITE (JWRITE,11)
     IERR=1
     RETURN

9 C FORMAT (1H1//1X,44HWARNING -- BAD POINTS HAVE BEEN FOUND ON THE,1X
1 A5,1X,37HSURFACE BASED ON AN EDIT TOLERANCE OF,1X
10 C FORMAT (1X,15HBAD POINT AT I=I4,5X,4HX = ,F10.6,5X,4HY = ,F10.6,5
11 C FORMAT (1X,93HADJACENT BAD POINTS HAVE BEEN FOUND -- PLEASE CORRECT
     IT YOUR INPUT DATA AND RESUBMIT THIS CASE.)

END

PAGE 3
LISTING OF DECK: SMOXY

CARD NO.

1

SUBROUTINE SMOXY (THETA,X,Y,W,YSMO,YPS,YPPS,NP,NOSE,YLTE,YNOSE,YUT)
1E,EPS,DF,ITER,TITLE,IOPE,IER)

5

THIS SUBROUTINE PRESENTS A TECHNIQUE FOR SMOOTHING Y INPUT
COORDINATES USING LEAST SQUARES POLYNOMIAL AND CUBIC SPLINE
METHODS

10

IF IOP=0 OR 1, COMPUTE YPPU (UNSMOOTHED SECOND DERIVATIVES) FROM
LEAST SQUARES POLYNOMIAL FITTING OF Y VS THETA. THEN COMPUTE
YPPS (SMOOTHED SECOND DERIVATIVES) FROM LEAST SQUARES CUBIC
SPLINE FITTING OF YPPU VS THETA. FINALLY COMPUTE YSMO (SMOOTHED Y
COORDINATES) USING INVERSE CUBIC SPLINE METHOD.

15

IF IOP=2, COMPUTE SECOND DERIVATIVES FROM INPUT FIRST DERIVATIVES.
THEN COMPUTE UNSMOOTHED Y COORDINATES FROM SECOND DERIVATIVES AND
FOLLOW SAME PROCEDURES AS OUTLINED ABOVE FOR IOP 0 OR 1.

20

IF IOP=3, COMPUTE UNSMOOTHED Y COORDINATES FROM INPUT SECOND
DERIVATIVES. THEN FOLLOW SAME PROCEDURES AS OUTLINED ABOVE FOR
IOP 0 OR 1.

CODFD BY — HARRY MORGAN NASA/LARC/TAD/AAB 1982

25

DIMENSION THETA, X, Y, W, YSMO, YPS, AND YPPS BY NP IN CALLING
PROGRAM

DIMENSION TITLE(1), THETA(1), X(1), Y(1), W(1), YSMO(1), YPS(1), YPPS(1)

30

COMMON /HLM/ WK(200,10)

35

COMMON /SMY/ YPP(200), YSMO(200), DUM(200), A(200,4), YN(200), YPPU(20)
10), SUMY(300), YUTE(30)

40

DATA LMX/200/, WT/100.1/

SINH(X)=EXP(X)-EXP(-X))/2.
LISTING OF DECK: SMXY

CARD NO.

41   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

50   C   IF IOP=2, COMPUTE SECOND DERIVATIVES FROM INPUT FIRST
      C   DERIVATIVES. THEN COMPUTE INITIAL Y/C COORDINATES FROM SECOND
      C   DERIVATIVES.

55   C   DO 2 I=1,NP
      2   DUM(I)=1.0
      T1=0.0
      CALL CSDS (LMX, NP, THETA, YPPS, 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)
      CONTINUE

60   C   COMPUTE SECOND DERIVATIVES USING CSDS

65   C   DO 78 I=1, NP
      CALL LSQSMO (THETA, YPPS, YUSMO, NP, DELTA, EPS, IERR)
      IF (IERR.NE.0) RETURN
      CALL YNEW (THETA, YPPS, YNOSE, NP, YLTE, YNOSE, YUTE, EPS, DUM, WK, JWRITE, 1)
      CALL YNEW (THETA, YPPS, YUSMO, NP, YLTE, YNOSE, YUTE, EPS, DUM, WK, JWRITE, 1)

70   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

PAGE 2
LISTING OF DECK: SMOXY

CARD NO.

81 IF (I.EQ.1) GO TO 5
DELTATHETA(I)=THETA(I-1)
YN(I)=YPPS(I-1)*DELTAT/6.0+YPPS(I)*DELTAT/3.0+(Y(I)-Y(I-1))/DELTAT
DUM(I)=YPP(I-1)*DELTAT/6.0+YPP(I)*DELTAT/3.0+(YUSMO(I)-YUSMO(I-1))/DELTAT
85 5 GO TO 6
DELTATHETA(2)=THETA(1)
YN(I)=YPPS(I)*DELTAT/6.0+(Y(I)-Y(1))/DELTAT
DUM(I)=YPP(I)*DELTAT/6.0+(YUSMO(2)-YUSMO(I))/DELTAT
90 6 T1=YPPS(I)-YN(I)
T2=YPPS(I)-DUM(I)
SUM1=SUM1+T1*T1
SUM2=SUM2+T2*T2
7 WRITE (JWRITE,74) I,YPPS(I),YN(I),T1,DUM(I),T2
95 8 C SELECT OUTPUT FROM EITHER CSDS OR LSQSMO
DO 10 I=1,NP
IF (SUM2.LT.SUM1) GO TO 8
YPPS(I)=YPPS(I)
100 9 Y(I)=YUSMO(I)
YN(I)=DUM(I)
YUSMO(I)=Y(I)
105 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 10 CALL YNEW (THETA,YPPS,YNOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRITE,
110 10) COMPUTE FIRST DERIVATIVES
DO 12 I=1,NP
YUSMO(I)=Y(I)
YUSMO(I)=Y(I)
120 IF (I.EQ.1) GO TO 12

PAGE 3
DELTA=THETA(I)-THETA(I-1)
YN(I)=YPSS(I-1)*DELTA/6+YPSS(I)*DELTA/3+(Y(I)-Y(I-1))/DELTA
YPPI(I)=YPSS(I)
DELTA=THETA(2)-THETA(1)
YN(I)=YPSS(I)*DELTA/3+YPSS(2)*DELTA/6+(Y(2)-Y(1))/DELTA

C INITIALIZE ARRAYS
C
DO 14 I=1,NP
YUSMO(I)=Y(I)
IF (IOP.LT.2) YPP(I)=0.0
YUSMO(I)=THETA(I)*RAD
DUM(I)=1.
IF (ITER.EQ.0) GO TO 17
C
IF IOP=0 OR 1 AND NO SMOOTHING DESIRED (I.E. ITER=0), COMPUTE SECOND DERIVATIVE FROM CUBIC SPLINE SUBROUTINE
C
CALL CSDS (LMX,NP,THETA,Y,DUM,O,O,-1,A,VRK,P,JERR)
IF (JERR.NE.0) GO TO 71
C COMPUTE Y AND SECOND DERIVATIVE
DO 16 I=1,NP
IF (I.EQ.NP) GO TO 15
YUSMO(I)=A(I,1)
YN(I)=A(I,2)
YPPI(I)=A(I,3)
GO TO 16
C
DELTA=THETA(I)-THETA(I-1)
YUSMO(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)
YPPI(I)=6.*A(I-1,4)*DELTA+2.*A(I-1,3)
CONTINUE
GO TO 48
C
FIND MAXIMUM INPUT Y VALUE AND ITS LOCATION FOR UPPER AND LOWER SURFACES
C
YMAX=0.0
JMAXL=1
LISTING OF DECK: SMOXY

CARD NO.

161     DO 19 I=1,NOSE
        J=NOSE+1-I
        IF (ABS(Y(J)) .GT. YMAX) GO TO 18
        GO TO 19
        18     YMAX=ABS(Y(J))
        JMAX=J
        19     CONTINUE
        C      UPPER SURFACE
        YMAX=0.0
        JMAXU=1
        DO 21 I=NOSE,IP
        IF (ABS(Y(I)) .GT. YMAX) GO TO 20
        GO TO 21
        20     YMAX=ABS(Y(I))
        JMAXU=I
        21     CONTINUE
        C      COMPUTE UNSMOOTHED SECOND DERIVATIVE USING LEAST
                SQUARES POLYNOMIAL METHOD
        C
        J1=0
        ICON=0
        MTER=0
        J=ITER
        KTI=0
        IF (IPRINT .NE. 0) WRITE (WRITE,79) TITLE
        DO 23 J=1,30
        KTI=KTI+1
        LTER(I)=10
        J=J-10
        IF (J) 22,24,23
        22     LTER(I)=10+J
        GO TO 24
        23     CONTINUE
        24     DO 39 LL=1,KTI
                N=LTER(LL)
                DO 34 I=1,N1
                CALL LEAST SQUARES POLYNOMIAL SMOOTHING ROUTINE
JR1
        39
        34     CALL LEOSMO (THETA,YUSMC,W,YDN,DUM,YPPL,LP,JMAXL,JMAXU,NOSE,WT,FPS,
                I1ERR)
        30
LISTING OF DECK: SMOXY

CARD NO.

201  C IF (IERR.NE.0) RETURN
     COMPUTE ERROR TERM
     SUMY(I)=0.0
     DO 25 J=1, NP
     SUMY(I)=SUMY(I)+(YPP(J)-YPPU(J))*S
     J1=J+1
     IF (I.LE.3).AND.((LL.EQ.1)) GO TO 26
     IF (J.EQ.1) GO TO 26
     C CHECK FOR OSCILLATIONS IN CONVERGENCE OF ERROR TERM
     IF (SUMY(I)-SUMY(I-1)) 26 26, 32
     C LOAD ARRAYS FOR NEXT ITERATION
     DO 26 J=1, NP
     WK(J,I)=YPPU(J)
     IF ((LL.EQ.1).AND.(I.EQ.1)) YPPS(J)=YPPU(J)
     YPP(J)=YPPU(J)
     CC=YUSMO(J)
     IF (J1-2) 29, 29, 27
     AA=YN(J)-YUSMO(J)
     BB=A(J+1)-A(J+2)
     T1=SIGN(1.,AA)
     T2=SIGN(1.,BB)
     IF (T1.EQ.T2) OR (AA.EQ.BB) GO TO 28
     YUSMO(J)=A(J+2)-BB*(YUSMO(J)-A(J+2))/(AA-BB)
     GO TO 30
     225  28 YUSMO(J)=0.5*(YUSMO(J)+YN(J))
     GO TO 30
     229  29 YUSMO(J)=YN(J)
     30  A(J+1)=YN(J)
     31  A(J+2)=CC
     GO TO 33
     32  NTER=I-1
     ICON=2
     GO TO 36
     33  NTER=I
     235  C CHECK FOR CONVERGENCE BASED ON INPUT EPS
     IF (SUMY(I).LE.EPS) GO TO 35
     CONTINUE
     GO TO 36
     35  ICON=1
     240  C
LISTING OF DECK: SMOKY

CARD NO.

241 C PRINT SECOND DERIVATIVES GENERATED DURING SMOOTHING PROCESS

245 36 IF (IPRINT,NE,0) GO TO 38
WRITE (JWRITE,80) TITLE
DO 37 J=1,MP
WRITE (JWRITE,68) J,YSMD(J),(1/(H(0.5,J-1))-7+1#INTER)
WRITE (JWRITE,82) (SUMY(I),I=1,INTER)
IF (IPRINT,NE,0) WRITE (JWRITE,79) LL,(SUMY(I),I=1,INTER)
MTER=MTER+INTER
IF (ICON,NE,0) GO TO 40
CONTINUE
IF (ICON,NE,0) WRITE (JWRITE,83) MTER
IF (ICON,NE,1) WRITE (JWRITE,84) MTER
IF (ICON,NE,2) WRITE (JWRITE,85) MTER

255 C COMPUTE SMOOTHED SECOND DERIVATIVE USING LEAST SQUARES
C CUBIC SPLINE
C DO 41 I=1,MP
DUM(I)=DF
C CALL LEAST SQUARES CUBIC SPLINE ROUTINE
CALL CSDS (LMX,MP,THETA,YPPU,DUM,FLOAT(NP),-1,A,WK,IERR)
IF (IERR,NE,0) GO TO 71
C COMPUTE SECOND DERIVATIVE
C SUM=0.0
DO 44 I=1,MP
YPP(I)=A(I-1)
GO TO 43
C
260 41 CALL LEAST SQUARES CUBIC SPLINE ROUTINE
C
265 SUM=0.0
DO 44 I=1,MP
IF (I,NE,NP) GO TO 42
YPP(I)=A(I-1)
GO TO 43
C
270 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)
SUM=SUM+(YPPU(I)-YPP(I))**2
YPPU(I)=YPPS(I)
WRITE (JWRITE,88) SUM
C
275 C COMPUTE NEW Y COORDINATES FROM SMOOTHED SECOND DERIVATIVES
C CALL YNEW (THETA,YPP,YSMO,NOSE,NP,YUSMD(NOSE),YUSMD(NOSE),YUSMD(NP),E
1PS,DUM,WK,JWRITE,1)
C
280
LISTING OF DECK: SMOXY

CARD NO.

281  C  CHECK NEW Y COORDINATES FOR SMOOTHNESS  SQ 281
285  C  CALL LEAST SQUARES POLYNOMIAL ROUTINE  SQ 283
  DO 45 I=1,NP  SQ 284
  A(I,1)=0.0  SQ 285
  CALL LSQSMO (THETA,YSMO,YN,DUM,YPPS,NP,1,NP,NOSE,WT,EPS,IERR)  SQ 286
  IF (IERR.NE.0) RETURN  SQ 287
  C  COMPUTE ERROR TERMS  SQ 288
    SUM1=0.0  SQ 289
    SUM2=0.0  SQ 290
    DO 46 I=1,NP  SQ 291
      A(I,1)=YSMO(I)-YN(I)  SQ 292
      A(I,2)=YPP(I)-YPPS(I)  SQ 293
      SUM1=SUM1+A(I,1)**2  SQ 294
      SUM2=SUM2+A(I,2)**2  SQ 295
  295  C  COMPUTE FIRST DERIVATIVE FROM SMOOTHED SECOND DERIVATIVE  SQ 296
    N1=NP-1  SQ 297
    DO 47 I=1,N1  SQ 298
      DELTA=THETA(I+1)-THETA(I)  SQ 299
      YN(I)=YPP(I)*DELTA/3.-YPP(I+1)*DELTA/6.+(YSMO(I+1)-YSMO(I))/DELTA  SQ 300
      DELTA=THETA(NP)-THETA(N1)  SQ 301
      YN(NP)=YPP(N1)*DELTA/6.+YPP(NP)*DELTA/3.+(YSMO(NP)-YSMO(N1))/DELTA  SQ 302
  305  C  PRINT SUMMARY OF SMOOTHED AND UNSMOOTHED DATA  SQ 303
    WRITE (JWRITE,86) TITLE  SQ 304
    YPS(I)=YN(I)  SQ 305
    IF (THETA(I).LE.0.) YN(I)=YN(I)  SQ 306
    T1=APS(THETA(I))  SQ 307
    IF (T1.GT.PI2) GO TO 49  SQ 308
    GP=CONS*SIN(T1)  SQ 309
    GPP=CONS*COS(T1)  SQ 310
    GO TO 50  SQ 311
    49  DIF=COSH(T1-PI2)  SQ 312
    DELTA=SINH(T1-PI2)  SQ 313
    GP=CONS/DIF  SQ 314
    GPP=-CONS*DELTA/(DIF*DIF)  SQ 315
  320

84
LISTING OF DECK: SMDXY

CARD NO.

321 50 IF (I.EQ.NOSE) GO TO 51
DYDX=YN(I)/GP
DY2DX=(YP(I)*GP-YN(I)*GPP)/(GPP*3)
CURV=ABS(DY2DX)/(SQRT(1+DYDX**2)*3)
GO TO 52

325 51 DYDX=0.1E99
DY2DX=0.1E99
CURV=CONS/(YN(I)**2)
RLE=1./CURV

330 52 DELTA+THETA(I)*RAD
DIF=Y(I)-YSMD(I)
YP(S(I)+YPP(I))

335 53 WRITE (JWRITE,87) X,DELTA,X(I),Y(I),YSMO(I),YSHO(I),DIF,YPS(I),YP, I P(I),DYDX,DY2DX,CURV
WRITE (JWRITE,89) RLE

C CHECK FOR INTERSECTION OF UPPER AND LOWER SURFACES

C OFFINE ITERATION INTERVAL

340 KRT=1001
N1=2*KRT
TE=THETA(NP)
TN=-THETA(I)
IF (TN.LT.TE) TE=TN
DIF=TE/FLT(KRT-1)
BB=0.5*DIF
AA=0.85*TE
YL1=YU1+YSMO(NOSE)
TP=TN=0.0

350 J1=NOSE
J2=2

C DO-LOOP TO SEARCH FOR INTERSECTION

DO 50 I=2,N1
IF (TP.LE.AA) TN=TN+DIF
IF (TP.GT.AA) TN=TN+BB
IF (TN.GT.TE) GO TO 61

355 TI=TN

C FIND UPPER SURFACE Y-COORDINATE AT THETA = TN

DO 54 K=J1,NP
J=K-1

360
LISTING OF DECK: SMOXY

CARD NO.

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

PAGE 11

CARD NO.  

401  
T1=T1*RAD  
WRITE (JWRITE,72) T1  
IERR=1  
RETURN  

405  
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  
415  
GP=BB*BB-4.*AA*CC  
IF (GP) 66,62,62  
62 GP=SQR(T(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  
420  
63 KRT=KRT+1  
WK(KRT,1)=T1  
425  
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  
430  
C IF (KRT.EQ.0) GO TO 70  
FIND X/C AND Y/C WHERE DY/DX=0.  
C  
DO 69 I=1,KRT  
CALL INTER (WK(I,1),WK(I,2),NP,THETA,X,1,KTI,O)  
DO 67 J=1,N1  
435  
J1=J  
J2=J+1  
IF (WK(I,1).GE.THETA(J).AND.WK(I,1).LE.THETA(J+1)) GO TO 68  
CONTINUE  
67  
AA=THETA(J2)-WK(I,1)  
BB=WK(I,1)-THETA(J1)  
440
LISTING OF DECK: SMDXY  PAGE 12

CARD NO.

441    WK(I,J)*WK(I,J)*RAD  
        DELTA=THETA(J)-THETA(J1)  

442    WK(I,3)*YPP(J1)*(AA**3/(6.*DELTA)-AA*DELTA/6.)*YPP(J2)*BB**3/(6.*  

443    DELTA)-BB*DELTA/6.)*YSHO(J1)*(AA*YSHO(J2)+BB)/DELTA  

444    CONTINUE  

445    IF (KRT.GT.O) WRITE (JWRITE,90) (WK(I,2),WK(I,3),WK(I,1),I=1,KRT)  

446    C    PRINT RESULTS OF SMOOTHNESS CHECK  

447    C    IF (ITER.EQ.O) RETURN  

448    WRITE (JWRITE,91) TITLE,DF  

449    WRITE (JWRITE,92) (I,A(I,1),A(I,2),I=1,NP)  

450    WRITE (JWRITE,93) SUM1,SUM2  

451    RETURN  

452    C    PRINT WARNING MESSAGE IF ERROR OCCURRED IN CALL TO CSDS  

453    WRITE (JWRITE,94) IERR  

454    RETURN  

455    C    FORMAT (/*5X,10BERROR MESSAGE --- SMOOTHING PROCESS RESULTED IN  

456    1AN INTERSECTION OF THE UPPER AND LOWER SURFACES AT THETA = F10.3)  

457    FORMAT (1H1,1X,7HTITLE--,2X,8A10//12X,62H--CHECK OF FIRST DERIVATI  

458    IVES GENERATED FROM IQP=2 INPUT DATA--/*9X,1HI,5X,12HDY/DT(INPUT),  

459    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

460    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

461    FORMAT (5X,15,F10.6))  

462    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

463    FORMAT (5X,15,F10.6))  

464    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

465    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

466    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

467    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

468    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

469    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

470    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

471    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

472    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

473    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

474    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

475    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

476    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

477    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

478    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

479    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

480    24X,12HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMD),8X,3HDIF/)  

88
LISTING OF DECK: SMXY

CARD NO.

481

1NS)

85 FORMAT (/3X,41HSMOOTHING PROCESS BEGAN OSCILLATING AFTER,I4,1X,10H)

1ITERATIONS)

86 FORMAT (1H1,1X,7HTITLE--2X,8A10//48X,28H--SMOOTHING OUTPUT SUMMAR)

485

2Y--//4X,1HI,5X,5HTHETA,5X,3HX/C,7X,3HY/C,7X,4HYT/C,5X,6HYSM/C/4X,

25HDELTA,7X,3HIPS,6X,4HYPPS,8X,5HDY/DX,7X,11HD(Y/DX)/DX,6X,

19HCURVATURE/

87 FORMAT (15+F10.2,7F10.6,3E15.6)

490

88 FORMAT (/3X,58HSUM OF SQUARES FROM LEAST SQUARES CUBIC SPLINE SMOO

495

1THING =E12.4)

89 FORMAT (/3X,22HLEADING-EDGE RADIUS/C=F10.6)

497

90 FORMAT (/3X,16HDY/DX=0. AT X/C=F10.6,5X,4HY/C=,F10.6,5X,6HTHETA=,

500

1F10.3)

91 FORMAT (1H1,1X,7HTITLE--2X,8A10//12X,29HCHECK OF SMOOTHED COORDIN

500

ATES,3X,3HDF=,F10.6//9X,1HI,5X,20H(YSMO/C-CHECK VALUE),7X,

500

21B(YPPS-CHECK VALUE)/)

92 FORMAT (5X,I5,10X,F10.6,15X,F10.6)

93 FORMAT (/5X,15HSUM OF SQUARES=F10.6,15X,F10.6)

94 FORMAT (/3X,21HINPUT ERROR -- POINT, I3,18H IS NOT INCREASING/)
Subroutine YNEW (THETA,YPP,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JW)

1) Routine to compute new y/c coordinates using an iteration

5) Procedure that insures a desired y/c coordinate at the nose

(C = or D) or that insures continuity of the first derivative w/r to

C = theta at the nose (IPT = 1)

C = coded by -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

10) Dimension theta, ypp, y, and dum by np and wk by 2*np in

C = calling program

C = dimension theta(1), ypp(1), y(1), dum(1), wk(1)

15) Initialize iteration parameters

NMAX = 20

N1 = -1

DELTA = 0.

T1 = theta(noise) - theta(noise - 1)

T2 = theta(noise + 1) - theta(noise)

DO 1 I = 1, NP

DUM(I) = YPP(I)

1) Iteration loop to compute incremental adjustment to second

C = derivative to insure that the desired convergence option at

C = the nose is obtained

2) N1 = N1 + 1

30) IF (N1 .GT. NMAX) GO TO 11

31) IF (IPT .EQ. 1) GO TO 3

C = compute upper and lower surface y/c coordinates

C = concurrently

32) CALL INVY (THETA,DUM,1,NP,Y,YLTE,YUTE,WK)

33) Compute difference between output and desired y/c coordinate

C = at the nose

34) DIF = Y(NOSE) - YNOSE

35) GO TO 4

C = compute upper and lower surface y/c coordinates

C = consecutively

40)
CALL INVY (THETA,DUM,NOSE,MP,Y,YNSE,YUTE,WK)  YW 41
CALL INVY (THETA,DUM,1,NOSE,Y,YLTE,YNSE,WK)  YW 42
C
COMPUTE DIFFERENCE BETWEEN UPPER AND LOWER SURFACE FIRST  YW 43
C
AA=-DUM(NOSE)*T2/3.-DUM(NOSE+1)*T2/6.+(Y(NOSE+1)-Y(NOSE))/T2  YW 44
BB=DUM(NOSE-1)*T1/6.+DUM(NOSE)*T1/3.+(Y(NOSE)-Y(NOSE-1))/T1  YW 45
DIF=AA-BB  YW 46
C
CHECK FOR CONVERGENCE  YW 47
45 IF (ABS(DIF),LE,EPS) GO TO 9  YW 48
C
COMPUTE ADJUSTMENT VALUE TO SECOND DERIVATIVE  YW 49
C
IF (N1,EQ,0) GO TO 6  YW 50
IF (DIF,EQ,DIFP) GO TO 5  YW 51
SP=(DELTA-DELAP)/(DIF-DIFP)  YW 52
DELAP=DELTA  YW 53
DIFP=DIF  YW 54
DELTA=DELTA-DIF*SP  YW 55
GO TO 7  YW 56
DELTA=0.5*(DELTA+DELAP)  YW 57
GO TO 7  YW 58
DELAP=DELTA  YW 59
DIFP=DIF  YW 60
DELTA=DELTA+DIF  YW 61
C
ADD ADJUSTMENT VALUE TO SECOND DERIVATIVE  YW 62
7 DO 8 I=1,NP  YW 63
8 DUM(I)=YPP(I)+DELTA  YW 64
C
CONTINUE TO ITERATE  YW 65
GO TO 2  YW 66
C
PRINT CONVERGENCE MESSAGE  YW 67
C
WRITE (JWRITE,14) N1,DELTA  YW 68
C
REDEFINE THE SECOND DERIVATIVE  YW 69
C
DO 10 I=1,NP  YW 70
10 YPP(I)=DUM(I)  YW 71
C
IF (IPT.EQ.1) GO TO 12  YW 72
GO To 13  YW 73
C
PRINT NON-CONVERGENCE MESSAGE  YW 74
C
N1=N1-1  YW 75
C
LISTING OF DECK: YNEW

CARD NO.

81  WRITE (WRITE,15) N1
82  WRITE (WRITE,15) N1
83  COMPUTE NEW UPPER AND LOWER SURFACE Y/C COORDINATES CONCURRENTLY
84  CALL INVY (THETA,YPP,1,NP,Y,YLTE,YUTE,WK)
85  RETURN TO CALLING PROGRAM
86  RETURN
87  RETURN
88  FOR ATTACHMENT (/*X,88HITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND DERIVATIVE CONVERGED IN */I3,23H ITERATIONS AND DELTA Y/*E12.4)*/2
89  FORMAT (*3X,88HITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND DERIVATIVE DID NOT CONVERGE IN */I3,11H ITERATIONS)*/2
90  END
91  END
92  END
LISTING OF DECK: INVY

CARD NO.

1  SUBROUTINE INVY (X, YPP, NS, NE, YSTART, YEND, A)
   C THIS ROUTINE COMPUTES Y VALUES FROM KNOWN SECOND DERIVATIVES AND
   C END CONDITIONS
   C Coded by -- Harry Morgan NASA/LARC/TAD/AAB 1982
   C IN CALLING PROGRAM DIMENSION X, YPP, AND Y BY NE AND A BY 2*NE

5  DIMENSION X(1), YPP(1), Y(1), A(NE,2)
   C SET END CONDITIONS
   Y(NS)=YSTART
   Y(NE)=YEND

10  C PERFORM FORWARD ELIMINATION
   A(1,1)=YSTART
   A(1,2)=0.0
   N=NE-NS+1
   NI=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)+H1+H2)*YPP(J)/6.+H2*YPP(J+1)/6.+H1*H2
      D=-H2*(A(I-1,2)+1.)*H1
      A(I,2)=H1/D

15  C PERFORM BACK SUBSTITUTION
      A(I,1)=(C-H2*A(I-1,1))/D

20  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)

25  C RETURN TO CALLING PROGRAM
SUBROUTINE LSQSMO (X,Y,W,N,YP,YPP,NMAX,JMAX,NOSE,WT,EPS,IERR) 

THIS SUBROUTINE IS USED TO SMOOTH X AND Y BY CONSECUTIVELY FITTING 
A LEAST SQUARES POLYNOMIAL OF DEGREE 4 THRU 7 POINTS AT A TIME 

CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982 

DIMENSION X(1), Y(1), W(1), YN(1), YP(1), YPP(1) 

DIMENSION XI(7), YI(7), WW(7), A(5,6), B(5) 

COMMON /INOUT/ JREAD,JWRITE,IPRINT 

CHECK NOSE REGION FOR SYMMETRY 

ISYM=1 
DO 1 I=1,3 
IF (ABS(X(NOSE-I)+X(NOSE+I)).GT.EPS) TSYM=0 
IF (ABS(Y(NOSE-I)+Y(NOSE+I)).GT.EPS) ISYM=0 
CONTINUE 
IERR=0 
FIT A LEAST SQUARES POLYNOMIAL OF DEGREE 4 THRU 7 POINTS 

DO 14 I=1,N 
LOAD 7 POINTS FOR LEAST SQUARES POLYNOMIAL FIT 
IF (I.LT.4) GO TO 2 
IF (I.GT.N-3) GO TO 3 
J1=I-3 
J2=I+3 
GO TO 4 
2 J1=1 
J2=7 
GO TO 4 
3 J1=N-6 
J2=N 
4 KK=0 
IF (ISYM.EQ.0) GO TO 7 
IF (I.GT.NOSE-3.AND.I.LE.NOSE) GO TO 5 
IF (I.LT.NOSE+3.AND.I.GT.NOSE) GO TO 6
GO TO 7
J1=NOSE-6
J2=NOSE
GO TO 7
J1=NOSE
J2=NOSE+6
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)
X(I(KK))=X(J)
DO 11 K=1,7
T1=1.
DO 11 J=1,5
8 Y=K(1,0)
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*X(K)
A(J,L)=A(J,L)-Y(K)*T1*WW(K)
11 T1=T1*X(K)
C SOLVE FOR COEFFICIENTS OF LEAST SQUARES POLYNOMIAL
DO 12 K=1,4
T1=A(J+1,K)/A(K,K)
12 A(J+1,K)=A(J+1,K)-A(K,L)*T1
B(5)=-A(5,5)/A(5,5)
DO 13 L=2,5
K=6-L
B(K)=-A(K,6)/A(K,K)
LISTING OF DECK: LSOSMD

CARD NO.

81   K1 = K + 1

82   DO 13 J = K1, 5

83   13   B(K) = B(K) - B(J) * A(K, J) / A(K, K)

84   C   COMPUTE NEW Y, FIRST, AND SECOND DERIVATIVE

85   YN(I) = (((B(5) * X(I) + B(4)) * X(I) + B(3)) * X(I) + B(2)) * X(I) + B(1)

86   YP(I) = (((4 * B(5) * X(I) + 3 * B(4)) * X(I) + 2 * B(3)) * X(I) + B(2)

87   YPP(I) = (12 * B(5) * X(I) + 6 * B(4)) * X(I) + 2 * B(3)

88   CONTINUE

89   IF (I < YH, EQ, 0) RETURN

90   YN(NOSE) = 0.0

91   YPP(NOSE) = 0.0

92   RETURN

93   END
SUBROUTINE CSDS (MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)

C*** List of Card Numbers ***
CARD NO.                  PAGE 1
1 SUBROUTINE CSDS (MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)     CS 1
C*** List of Card Numbers ***
CARD NO.                  PAGE 1
5  C* PURPOSE:
C* SUBROUTINE CSDS FITS A SMOOTH CUBIC SPLINE TO A
C* UNIVARIATE FUNCTION. DATA MAY BE UNEQUALLY SPACED.
C* USE:
C* CALL CSDS(MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)
C* MAX INPUT INTEGER SPECIFYING THE MAXIMUM NUMBER OF DATA
C* POINTS FOR THE INDEPENDENT VARIABLE.
C* IX INPUT INTEGER SPECIFYING THE ACTUAL NUMBER OF DATA
C* POINTS FOR THE INDEPENDENT VARIABLE. IX<=MAX.
C* X ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST
C* IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS,
C* X(I) MUST CONTAIN THE VALUE OF THE INDEPENDENT
C* VARIABLE AT POINT I.
C* F ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST
C* IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS,
C* F(I) MUST CONTAIN THE VALUE OF THE FUNCTION AT
C* POINT X(I).
C* DF ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST
C* IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS,
C* DF(I) MUST CONTAIN AN ESTIMATE OF THE STANDARD
C* DEVIATION OF F(I).
C* S A NON-NEGATIVE INPUT PARAMETER WHICH CONTROLS THE
C* EXTENT OF SMOOTHING. S SHOULD BE IN THE RANGE
C* (IX-(2*IX)**.5)<S<(IX+(2*IX)**.5).
C* IPT INPUT INITIALIZATION PARAMETER. THE USER MUST
C* SPECIFY IPT=-1 WHENEVER A NEW X ARRAY IS
C* INPUT. THE ROUTINE WILL ALSO CHECK TO INSURE THAT
C* THE X ARRAY IS IN STRICTLY INCREASING ORDER.
C*
**LISTING OF DECK: CSDS**

**CPS NO.**

<table>
<thead>
<tr>
<th>Page</th>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>C*</td>
<td>COEF</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>A TWO-DIMENSIONAL OUTPUT ARRAY DIMENSIONED (MAX,4)</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>IN THE CALLING PROGRAM. UPON RETURN, COEF(I,J)</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>CONTAINS THE J-TH COEFFICIENT OF THE SPLINE FOR</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>THE INTERVAL BEGINNING AT POINT X(I). THE</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>FUNCTIONAL VALUE OF THE SPLINE AT ABSCISSA XI,</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>WHERE XI&lt;X1&lt;XI+1, IS GIVEN BY:</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>F(X1) = ((COEF(I,4)*H+COEF(I,3))*H+COEF(I,2))*H</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>+COEF(I,1)</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>WHERE H= X1-XI</td>
</tr>
<tr>
<td>50</td>
<td>C*</td>
<td>WK</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>A ONE-DIMENSIONAL WORK AREA ARRAY DIMENSIONED AT</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>LEAST (7*IX+9) IN THE CALLING PROGRAM.</td>
</tr>
<tr>
<td>55</td>
<td>C*</td>
<td>IERR</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>OUTPUT ERROR PARAMETER:</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>=0 NORMAL RETURN. NO ERROR DETECTED.</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>=J THE J-TH ELEMENT OF THE Y ARRAY IS NOT IN</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>STRICTLY INCREASING ORDER.</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>=-1 THERE ARE LESS THAN FOUR VALUES IN THE X ARRAY.</td>
</tr>
<tr>
<td>60</td>
<td>C*</td>
<td>UPON RETURN FROM CSOS, THIS PARAMETER SHOULD BE</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>TESTED IN THE CALLING PROGRAM.</td>
</tr>
<tr>
<td>65</td>
<td>C*</td>
<td>REQUIRED ROUTINES</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>-NONE</td>
</tr>
<tr>
<td>70</td>
<td>C*</td>
<td>LANGUAGE</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>-FORTAN</td>
</tr>
<tr>
<td>75</td>
<td>C*</td>
<td>DATE RELEASED</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>SEPTEMBER 5, 1973</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>LATEST REVISION</td>
</tr>
<tr>
<td></td>
<td>C*</td>
<td>MARCH 1975</td>
</tr>
</tbody>
</table>

**SOURCE**

IMSL ROUTINE ICSSMU MODIFIED BY COMPUTER SCIENCES CORPORATION

**DATE RELEASED**

SEPTEMBER 5, 1973

**LATEST REVISION**

MARCH 1975

**DIMENSION**

X(I), F(I), DF(I), COEF(MAX,4), WK(1)

**SET UP WORKING AREAS**
LISTING OF DECK: CSDS

CARD NO.

81  C
    IERR=0
    IF (IPT.NE.-1) GO TO 4
    IPT=0

85  CARD NO.
    IF (IX.LT.4) GO TO 1
    GO TO 2
    IERR=-1
    RETURN
    IX=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.
    IJK5=IB5+NP1
    CONTINUE
    P=0.
    H=X(2)-X(1)
    F2=-S
    FF=(F(2)-F(1))/H

120  IF (IX.LT.3) GO TO 10
    CS 81
    CS 82
    CS 83
    CS 84
    CS 85
    CS 86
    CS 87
    CS 88
    CS 89
    CS 90
    CS 91
    CS 92
    CS 93
    CS 94
    CS 95
    CS 96
    CS 97
    CS 98
    CS 99
    CS 100
    CS 101
    CS 102
    CS 103
    CS 104
    CS 105
    CS 106
    CS 107
    CS 108
    CS 109
    CS 110
    CS 111
    CS 112
    CS 113
    CS 114
    CS 115
    CS 116
    CS 117
    CS 118
    CS 119
    CS 120
LISTING OF DECK: CSDS

CARD NO.

121  DO 5 I=3,IX
  G=H
  H=X(I)-X(I-1)
  E=FF

125  FF=(F(I)-F(I-1))/H
  CDEF(I-1,1)=FF-E
  IJK3=IB3+I
  WK(IJK3)=(G+H)*666666666666
  IJK4=IB4+I

130  WK(IJK4)=H/3.
  IJK2=IB2+I
  WK(IJK2)=DF(I-2)/G
  WK(I)=DF(I)/H
  IJK1=IB1+I

135  WK(IJK1)=DF(I-1)/G-DF(I-1)/H

140  CONTINUE
  DO 6 I=3,IX
  IJK1=IB1+I
  IJK2=IB2+I
  CDEF(I-1,2)=WK(I)*WK(I)+WK(IJK1)*WK(IJK1)*WK(IJK2)*WK(IJK2)
  CDEF(I-1,3)=WK(I)*WK(IJK1+1)+WK(IJK1)*WK(IJK2+1)
  CDEF(I-1,4)=WK(I)*WK(IJK2+2)

145  NEXT ITERATION

150  IF (IX.LT.3) GO TO 10
  DO 8 I=3,IX
  IJK1=IB1+I-1
  IJK0=I-1
  WK(IJK1)=FF*WK(IJK0)
  IJK2=IB2+I-2
  IJK0=I-2
  WK(IJK2)=G*WK(IJK0)

155  IJK=I
  IJK3=IB3+I
  WK(IJK0)=1./(P*CDEF(I-1,2)+WK(IJK3)-FF*WK(IJK1)-G*WK(IJK2))
  IJK5=IR5+I
  IJKN=IJK5-1

160  IJK0=IJKN-1

CS 121
CS 122
CS 123
CS 124
CS 125
CS 126
CS 127
CS 128
CS 129
CS 130
CS 131
CS 132
CS 133
CS 134
CS 135
CS 136
CS 137
CS 138
CS 139
CS 140
CS 141
CS 142
CS 143
CS 144
CS 145
CS 146
CS 147
CS 148
CS 149
CS 150
CS 151
CS 152
CS 153
CS 154
CS 155
CS 156
CS 157
CS 158
CS 159
CS 160
LISTING OF DECK: CSDS  

CARD NO.  

161  
WK(IJK5)=COEF(I-1+1)-WK(IJK1)*WK(IJK0)-WK(IJK2)*WK(IJK0)  
IJK5=IB5+I  
FF=P*COEF(I-1,3)+WK(IJK4)-H*WK(IJK1)  
G=H  

165  
8 CONTINUE  
DO 9 I=3,IX  
J=IX-I+3  
IJK5=IB5+J  
IJK6=IJK5+1  
IJK7=IJK6+1  
IJK1=IB1+J  
IJK2=IB2+J  
WK(IJK5)=WK(J)*WK(IJK5)-WK(IJK1)*WK(IJK6)-WK(IJK2)*WK(IJK7)  

170  
9 CONTINUE  
10 E=0  
H=0  
C  
C  

COMPUTE U AND ACCUMULATE E  

180  
C  
DO 11 I=2,IX  
G=H  
IJK5=IB5+I  
H=(WK(IJK5+1)-WK(IJK5))/X(I)-X(I-1))  
IJK6=IB6+I  
WK(IJK6)=(H-G)*DF(I-1)*DF(I-1)  
E=E+WK(IJK6)*H  
11 CONTINUE  
G=H*DF(IX)*DF(IX)  
IJK6=IB6+NP1  
WK(IJK6)=G  
E=E+G*H  
G=F2  
F2=E*F*P  
195  
IF (F2.GE.S0.OR.F2.LE.G) GO TO 14  
FF=0.  
IJK6=IB6+2  
H=(WK(IJK6+1)-WK(IJK6))/X(2)-X(1))  
190  
IF (IX.LT.3) GO TO 13  
194  
199  
DO 12 I=3,IX  
200
LISTING OF DECK: CSDS

CARD NO.

201
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
IF (H.LE.0) GO TO 14

C
C
C
P=P+(S-F2)/((SQRT(S/E)+P)*H)
GO TO 7

C
C
C
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,NX
H=X(I)-X(I-1)
230
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

CS 201
CS 202
CS 203
CS 204
CS 205
CS 206
CS 207
CS 208
CS 209
CS 210
CS 211
CS 212
CS 213
CS 214
CS 215
CS 216
CS 217
CS 218
CS 219
CS 220
CS 221
CS 222
CS 223
CS 224
CS 225
CS 226
CS 227
CS 228
CS 229
CS 230
CS 231
CS 232
CS 233
CS 234
CS 235

PAGE 6
LISTING OF DECK: PCARD  PAGE 1

CARD NO.

1  SUBROUTINE PCARD (IPUNCH, X, Y, W, THETA, YSMD, YPS, YPPS, NOSE, NP, CHORD, T TITLE)

5  ROUTINE TO PUNCH OUTPUT DATA (TAPE 1 IS PUNCH FILE)

C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C C C
C

PH 1
PH 2
PH 3
PH 4
PH 5
PH 6
PH 7
PH 8
PH 9
PH 10
PH 11
PH 12
PH 13
PH 14
PH 15
PH 16
PH 17
PH 18
PH 19
PH 20
PH 21
PH 22
PH 23
PH 24
PH 25
PH 26
PH 27
PH 28
PH 29
PH 30
PH 31
PH 32
PH 33
PH 34
PH 35
PH 36
PH 37
PH 38
PH 39
PH 40

10 COMMON /HLH/ DX(200),DY(200),DW(200)

C COMMON /BLK1/ PI,PI2,RAD,CONS

15 COMMON /INOUT/ JREAD,JWRITE,IPRINT

C IF (IPUNCH.LE.0.OR.IPUNCH.GE.5) RETURN

C PUNCH TITLE CARD

20 WRITE (JWRITE,10) IPUNCH,TITLE

C WRITE (1,11) TITLE

C DETERMINE OUTPUT PUNCH OPTION

25 INP=0

C IF (IPUNCH.EQ.2) IDP=1

C IF (IPUNCH.EQ.3) IDP=2

C IF (IPUNCH.EQ.4) IDP=3

30 WRITE (JWRITE,12) IDP

C XI=FLOAT(IDP)

C WRITE (1,13) XI

C PUNCH UPPER SURFACE QUANTITIES

35 J=KP=0

C DO 1 I=NOSE,NP

C J=J+1

C DW(J)=W(I)

40 IF (W(I).GT.1.0) KP=1
LISTING OF DECK: PCARD

CARD NO.

41 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)
45 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)
50 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)
55 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
60 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 PUNCH LOWER SURFACE QUANTITIES
65 C
J=KP=0
DO 4 I=1,NOS-E
J=J+1
K=NOS*1-I
70 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)=YSMD(K)*CHORD
IF (IOP.EQ.1) DY(J)=YSMD(K)
IF (IOP.EQ.2) DY(J)=YPS(K)
IF (IOP.EQ.3) DY(J)=YPPS(K)
4 CONTINUE
WRITE (JWRITE,19) J
80 XI=FLOAT(J)
LISTING OF DECK: PCARD

CARD NO.

81 WRITE (1,15) XI
82 IF (IOP.EQ.O) WRITE (JWRITE,16) (DX(I),I=1,J)
83 IF (IOP.NE.O) WRITE (JWRITE,7) (DX(I),I=1,J)
84 WRITE (JWRITE,17) (DY(I),I=1,J)
85 IF (KP.EQ.1) WRITE (JWRITE,21) (DW(I),I=1,J)
86 DO 6 I=1,J
87 IF (IOP.NE.O) GO TO 5
88 IF (DW(I).GT.1.0) WRITE (1,22) DX(I),DY(I),DW(I)
89 IF (DW(I).LE.1.0) WRITE (1,18) DX(I),DY(I)
90 GO TO 6
91 IF (DW(I).GT.1.0) WRITE (1,8) DX(I),DY(I),DW(I)
92 IF (DW(I).LE.1.0) WRITE (1,9) DX(I),DY(I)
93 CONTINUE
94 C
95 C PUNCH YLTE AND YUTE
96 IF (IOP.LE.1) RETURN
97 YLTE=YSM(Q(I))
98 YNose=YSMO(NOSE)
99 YUTE=YSMO(NP)
100 WRITE (JWRITE,20) YLTE,YNose,YUTE
101 WRITE (1,18) YLTE,YNose,YUTE
C
102 C RETURN TO CALLING PROGRAM
103 C
104 RETURN
C
105 FORMAT (/3X,4HTH=8F10.5/(7X,8F10.5))
106 FORMAT (F10.5,F10.6,F10.2)
107 FORMAT (F10.5,F10.6)
108 FORMAT (1H*,I4/3X,8A10)
109 FORMAT (8A10)
110 FORMAT (/3X,5HIOP=I4)
111 FORMAT (30X,F10.2)
112 FORMAT (/3X,7HNU=I4)
113 FORMAT (F10.2)
114 FORMAT (/3X,6HDX=8F10.6/(7X,8F10.6))
115 FORMAT (/3X,6HDY=8F10.6/(7X,8F10.6))
116 FORMAT (3F10.6)
LISTING OF DECK: PCARD

CARD NO.

121  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,6F10.2))  PH 123
     22  FORMAT (2F10.6,F10.2)  PH 124

125  END  PH 125-
SUBROUTINE PLOTAF (THETA, Y, YSMO, YPS, YPPS, NP, TITLE, IPRINT)

THIS ROUTINE PLOTS INPUT AND SMOOTHED Y/C, SMOOTHED YPS, AND SMOOTHED YPPS VERSUS THETA. ALSO PLOTS INPUT AND SMOOTHED Y/C VERSUS X/C.

CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

DIMENSION TITLE(8), THETA(1), Y(1), YSMO(1), YPS(1), YPPS(1)

COMMON /HLM/ XI(363), YI(363), TI(363)
COMMON /SMY/ YPSI(363)
COMMON /BLKI/ PI, PI2, RAD, CONS
COMMON /INPUT/ JREAD, JWRITE, IPRINT
DATA NM/361/, SIZ/.40/, ISIZ/3/

SINH(X) = (EXP(X) - EXP(-X)) / 2

INTERPOLATE NM SMOOTHED COORDINATES Y/C AND YPS VALUES

YMAX = 0.0
DP = (THETA(NP) - THETA(1)) / FLOAT(NM-1)
M = 2
DO 5 I = 1, NM
  YP = THETA(1) - 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
3  DO 20
41  J=K-1
   IF (YP .GE. THETA(J) .AND. YP .LE. THETA(K)) GO TO 4        PF 41
   CONTINUE
   IF (YP .GE. 0.06) GO TO 6
   WRITE (JWRITE,15) TITLE
   WRITE (JWRITE,16) (I,THETA(J),THETA(J),I=1,NM)

55  PRINT INTERPOLATED YC-COORDINATES
   IF (IPRINT .NE. 0) GO TO 6
   WRITE (JWRITE,15) TITLE
   WRITE (JWRITE,16) (I,THETA(J),THETA(J),I=1,NM)

60  DETERMINE SCALING FACTOR FOR Y/C AXIS
   YSCALE = 0.1
   IF (YP .LE. 0.06) YSCALE = 0.01
   IF (YP .LE. 0.12) YSCALE = 0.02
   IF (YP .LE. 0.24) YSCALE = 0.04
   IF (YP .LE. 0.30) YSCALE = 0.05
   YSCALE = 0.5

70  DRAW AND LABEL Y/C AND THETA AXIS
   IF (IPLOT .EQ. 2) GO TO 11
   CALL CALPLT (2,1,-3)
   CALL NOTATE (0.0,0.0,SIZ,TITLE,0.80)
   CALL AXES (0.0,2.0,3.0,-180,10,10,2,1,10,THETA,DEG,SIZ,-10,0)
   CALL AXES (0.0,2.0,3.0,90,12,90,THETA,YSCALE,-1.0,0.0,3HY/C,SIZ,3,2)
   CALL NOTATE (1.0,13.0,3,0,2,-1)
   CALL NOTATE (1.0,12.9,SIZ,BM SMOOTHED,0.8,0.80)
LISTING OF DECK: PLOTAF

CARD NO.

81
CALL NOTATE (1.5,13.5,SIZ,5,INPUT,0.,5)
CALL CALPLT (0.,8.,-3)
C
PLOT INPUT Y/C-COORDINATES VS THETA

85
DO 7 I=1,NP
DEC 7
TP=THETA(I)*RAD/10.+18.0
YP=Y(I)/YScale
CALL PNTPLT (TP,YP,22.,ISIZ)
CONTINUE
C
PLOT SMOOTHED Y/C-COORDINATES VS THETA

90
TI(NM+1)=-180.0
TI(NM+Z)=10.0
YI(NM+1)=0.
YI(NM+2)=YScale
CALL LINE (TI,YI,NM,1.,0.,0.,0.)
C
DETERMINE SCALING FACTOR FOR FIRST DERIVATIVE AXIS (YP AXIS)

95
YMAX=0.0
DO 8 I=1,NM
IF (ABS(YPSI(I)) .GT. YMAX) YMAX=ABS(YPSI(I))
CONTINUE
C
SCALe=-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
C
DETERMINE SCALING FACTOR FOR SECOND DERIVATIVE AXIS (YPP AXIS)

100
C
CMIN=-.6*CScale
C
DETERMINE SCALING FACTOR FOR SECOND DERIVATIVE AXIS (YPP AXIS)

105
YMAX=0.0
DO 9 I=1,NP
IF (ABS(YPPS(I)) .GT. YMAX) YMAX=ABS(YPPS(I))
CONTINUE
YSCALE=1.
IF ((YMAX.LE.3.00).AND.(YMAX.GT.2.40)) YSCALE=.5
LISTING OF DECK: PLOTAF

CARD NO.

121 IF ((YMAX.LE.2.40).AND.(YMAX.GT.1.20)) YSCALE=.4    PF 121
121 IF ((YMAX.LE.1.20).AND.(YMAX.GT.0.60)) YSCALE=.2    PF 122
121 IF ((YMAX.LE.0.60).AND.(YMAX.GT.0.30)) YSCALE=.1    PF 123
121 IF ((YMAX.LE.0.30).AND.(YMAX.GT.0.24)) YSCALE=.05   PF 124
125 IF ((YMAX.LE.0.24).AND.(YMAX.GT.0.12)) YSCALE=.04   PF 125
125 IF ((YMAX.LE.0.12).AND.(YMAX.GT.0.06)) YSCALE=.02   PF 126
125 IF ((YMAX.LE.0.06).AND.(YMAX.GE.0.00)) YSCALE=.01   PF 127
125 YMIN=-6.*YSCALE                                       PF 128
130 C DRAW AND LABEL YP, YPP, AND THETA AXIS              PF 129
130 C CALL CALPLT (0..8..-3)                               PF 130
130 C CALL AXES (0..0..0..90..12..0..3HYP..SIZ,3,2)        PF 131
130 C CALL AXES (36..0..90..12..0..4HYP..SIZ,-4,2)         PF 132
130 C CALL NOTATE (1.0,11,13,0..3,0..-1)                   PF 133
130 C CALL NOTATE (1.5,10,9,SIZ,4HYP..0..4)                PF 134
130 C CALL NOTATE (1.0,11,7,.4,.2,.0,.-1)                  PF 135
130 C CALL NOTATE (1.5,11,5,SIZ,3HYP..0..3)                 PF 136
130 C CALL CALPLT (0..6..-3)                               PF 137
135 C PLOT SMOOTHED FIRST DERIVATIVES YP VS THETA          PF 138
135 C YPSI(NM+1)=0.0                                        PF 139
135 C YPSI(NM+2)=CSSCALE                                    PF 140
135 C CALL LINE (Ti,YPSI,NM,1,0,0,0,)                      PF 141
150 C PLOT SMOOTHED SECOND DERIVATIVES YPP VS THETA        PF 151
150 C THETA(NP+1)=--PI                                      PF 152
150 C THETA(NP+2)=10.*RAD                                   PF 153
150 C YPPS(NP+1)=0.0                                       PF 154
150 C YPPS(NP+2)=YSCALE                                    PF 155
150 C CALL LINE (THETA,YPPS,NP,1,0,0,0,)                   PF 156
155 DO 10 I=1,NP                                          PF 157
155 TP=THETA(I)*RAD/10.+18.0                              PF 158
155 YP=YPPS(I)/YSCALE                                     PF 159
160 CALL PNTPLT (TP,YP,22,ISIZ)                            PF 160
LISTING OF DECK: PLOTAF

CARD NO.

161 10 CONTINUE
CALL NFRAME
C CHECK PLOT OPTION
IF (IPL6T.EQ.1.OR.IPL6T.EQ.6) RETURN
IF (IPL6T.EQ.8) RETURN
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 PLOT INPUT AND SMOOTHED Y/C-COORDINATES VS X/C
C DRAW AND LABEL Y/C AND X/C AXIS
CALL CALP L (2.0,2.0,-3)
CALL NOTATE (0.0,0.0,SIZ,TITLE,0.0,80)
CALL CALP L (0.0,2.0,-3)
CALL AXES (0.0,0.0,40,0.0,0.025,-2..1..3H/C,SIZ,-3,2)
CALL AXES (0.0,0.0,90,0.0,YMAX,YMIN,0.0,025,-2..1..3H/Y/C,SIZ,3,2)
YP=YMAX-0.0
CALL NOTATE (1.0,YP,SIZ,2.0,-1)
YP=YMAX-1.1
CALL NOTATE (1.5,YP,SIZ,8HSMOOTHED,0.0,8)
YP=YMAX-3
CALL NOTATE (1.0,YP,SIZ,3.0,-1)
YP=YMAX-5
CALL NOTATE (1.5,YP,SIZ,5HINPUT,0.0,5)
YP=0.5*YMAX
CALL CALP L (0.0,YP,-3)
C PLOT INPUT Y/C-COORDINATES
C
DO 14 I=1,NP
TP=ABS(THETA(I))
IF (TP.LE.PI2) GO TO 12
XP=COS*(ATAN(SINH(TP-PI2))+1.)/0.025
14 CONTINUE
LISTING OF DECK: PLOTAF

CARD NO.

201       GO TO 13
 12       XP=CONV*(1.-COS(TP))/0.025
 13       YP=Y(I)/0.025
       CALL PNTPLT (XP,YP,22,ISIZ)

205       CONTINUE

       PLOT SMOOTHED Y/C-COORDINATES

       XI(NM+1)=YI(NM+1)=0.0
       XI(NM+2)=YI(NM+2)=0.025
       CALL LINE (XI,YI,NM,0,0,0)

       RETURN TO CALLING PROGRAM

215       CALL NFRAME

       RETURN

15       FORMAT (1H1,1X,7HTITLE--2X,8A10//49X,28H--INTERPOLATED COORDINATE PF 218
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,14,F8.2,2F10.6))
       END

PF 201
PF 202
PF 203
PF 204
PF 205
PF 206
PF 207
PF 208
PF 209
PF 210
PF 211
PF 212
PF 213
PF 214
PF 215
PF 216
PF 217
PF 218
PF 219
PF 220
PF 221
PF 222-
LISTING OF DECK: PLOTCF

CARD NO.

1

SUBROUTINE PLOTCF (THETA,YSMO,YP,YPPS,NP,TITLE)
ROUTINE TO PLOT SQUARE ROOT OF SMOOTHED CURVATURE VERSUS THETA
CODED BY -- HARRY MORGAN NASA/LARC/TAD/AA 1982

DIMENSION THETA(1), YSMO(1), YPS(1), YPPS(1), TITLE(8)

COMMON /HLM/ TI(723)
COMMON /SMY/ CURV(723)
COMMON /RLK1/ PI, PI2, RAD, CONS
COMMON /INOUT/ JREAD, JWRITE, IPRINT

DATA NM/721/, SI/40/, SI2/3/

INTERPOLATE NM CURVATURE POINTS

IF (IPRINT.NE.0) GO TO 1
WRITE (JWRITE,15) TITLE
DP=(THETA(NP)-THETA(1))/FLOAT(NM-1)
TDEL=THETA(1)-DP
M=2
DO 1 I=1,NM
TDEL=TDEL+DP
IF (TDEL.LT.THETA(1)) TDEL=THETA(1)
IF (TDEL.GT.THETA(NP)) TDEL=THETA(NP)
1 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
M=J
DELTA=THETA(J+1)-THETA(J)
T2=THETA(J+1)-TP
40 TI=TP-THETA(J)

PAGE 1
LISTING OF DECK: PLOTCK

CARD NO.

41  YI=YPPS(J) *(T2**3/(6.*DELTA)-T2*DELTA/6.) *YPPS(J+1) *(T1**3/(6.*DELTA) PC 41
1A-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
YPI=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

50 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-PZ2) PC 54
T2=SINH(TP-PZ2) 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)/(SQR(T+DY2DX**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, T(I), XI, YI, YPI, YPPS, DYDX, DY2DX, CURV(I) PC 68
CURV(I)=SORT(CURV(I)) PC 69

8 C DETERMINE SCALING FACTOR FOR CURVATURE AXES PC 70
C CMAX=0.0 PC 71
75 IF (CURV(I).GT.CMAX) CMAX=CURV(I) PC 72
CONTINUE PC 73
M=IFIX(CMAX)+1 PC 74
C CMAX=FLOAT(M)/16. PC 75
C DRA AND LABEL CURVATURE AND THETA AXES PC 76
80 C
LISTING OF DECK: PLOTCK

CARD NO.

81 C
CALL GRIDCK
CALL CALPLT (2, 2, 2) PC 81
CALL NOTATE (0, 0, SZ, TITLE, 0, 0, 80) PC 82
CALL CALPLT (0, 2, 2) PC 83
CALL AXES (O, 0, 0, 0, 36, -180, 10, -2, 1, 10) PC 84
CALL AXES (O, 0, 90, 0, 0, CSAX, -2, 1, 15) PC 85
CALL NOTATE (O, 0, 0, 0, 0, 15) PC 86

85 C
PAGE 3

90 C
PLOT INTERPOLATED CURVATURE POINTS

90 C
TI(NH+1)=-180.0 PC 91
CURV(NH+1)=O.0 PC 92
TI(NHM+2)=10.0 PC 93
CURV(NHM+2)=CSAX PC 94
CALL LINE (TI, CURV, NH, 1, 0, 0, 0.0) PC 95

95 C
COMPUTE AND PLOT CURVATURE AT INPUT THETA POINTS

100 DO 1 I=1, NP
DELT=YS(I) PC 100
IF (THETA(I) .LE. 0.0) DELT=-DELTA PC 101
TP=ABS(THETA(I)) PC 102
IF (TP .GT. PI2) GO TO 10
GP=CONS*SIN(TP) PC 103
GPP=CONS*COS(TP) PC 104
GO TO 11 PC 105

10 C
T1=COSH(TP-PI2) PC 106
T2=SINH(TP-PI2) PC 107
GP=CONS/T1 PC 108
GPP=-CONS*T2/(T1*T1) PC 109

11 DO 12 I=1, NP
IF (TP .LE. 0.0 OR GP .EQ. 0.0) GO TO 12
DYDX=DELTA/GP PC 110
DY2DX=IT(YS(I)*GP-DELTA*GPP)/(GP**3) PC 111
T1=ARS(DY2DX)/(SQRT(1+DY2DX**2)**3) PC 112
GO TO 13 PC 113

12 T1=CONS/(DELTA*DELTA) PC 114
T2=THETA(I)*RAD/10.+18.0 PC 115
T1=SRT(T1)*CSAX PC 116
CALL PNTPLT (T2, T1, 22, ISIZ) PC 117

120 PC 120

116
LISTING OF DECK: PLOTCK

CARD NO.

121 14 CONTINUE
C ADVANCE TO NEXT FRAME AND RETURN
C
125 CALL NFRAME
C RETURN
C
130 15 FORMAT (14I,1X,7HTITLE--2X,8A10//36X,26H--INTERPOLATED CURVATURE--
1-3X,1HI,6X,5HTHETA,5X,3HX/C,7X,3HY/C,6X,5HDY/DT,5X,6HDY2/DT,7X,5H
2DY/DX,7X,11HD(DY/DX)/DX,5X,9HCURVATURE/)
16 FORMAT (15,F10.2,4F10.6,3E15.6)
END

PAGE 4
SUBROUTINE CAMTK (THETA, YSMO, YPPS, NOSE, NP, EPS, KPLOT, IPUNCH, TITLE)

THIS SUBROUTINE COMPUTES THE THICKNESS AND CAMBER DISTRIBUTIONS
OF THE SMOOTHED AIRFOIL

CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

DIMENSION TITLE(8), THETA(1), YSMO(1), YPPS(1)

COMMON /SHYI TU(100), YPPU(100), TL(100), YPPPL(100), DUXU(100), LX(101)
1,XLS(101), YLS(101), TH(101), XU(102), YU(102), XL(102), YL(102), XC(103)
2, YC(103), TK(103)

COMMON /BLK1/ PI, PI2, RAD, CONS

COMMON /INOUT/ JREAD, JWRITE, IPRINT

DATA NH/200/, SS/0.4/, ISIZ/3/

COSH(X)=0.5*(EXP(X)+EXP(-X))
SINH(X)=0.5*(EXP(X)-EXP(-X))

F(X1,X2,X3,X4,X5,X6,X7,X8,X9)=X1*(X5*X9-X6*X8)+X2*(X6*X7-X4*X9)+X3
1*(X4*X8-X5*X7)

LOAD THETA, X/C, VIC, AND SECOND DERIVATIVES INTO SEPARATE
ARRAYS FOR UPPER AND LOWER SURFACES

J=0
NU=NP-NOSE+1
DO 2 I=NOSE, NP
J=J+1
TU(J)=THETA(I)
YU(J)=YSMO(I)
TP=ABS(THETA(I))
IF (TP.GT.PI2) GO TO 1
XU(J)=CONS*(1.-COS(TP))
GO TO 2
1 XU(J)=CONS*ATAN(SINH(TP-PI2))+1.
2 YPPU(J)=YPPS(I)
LISTING OF DECK: CAMTK

CARD NO.

41
NL=NOSE
J=NOSE+1
DO 4 I=1,NOSE
J=J-1
45
TL(J)=THETA(I)
YL(J)=YSM0(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)))
5
IF (TU(I).GT.PI2) DYXU(I)=DYXU(I)*COSH(TU(I)-PI2)/CONS
CONTINUE
60
DYXU(1)=0.1E99
C
C
COMPUTE THICKNESS AND CAMBER DISTRIBUTIONS BY FINDING LOWER SURFACE COORDINATE (XLS,YLS) CORRESPONDING TO INPUT UPPER SURFACE COORDINATE (XU,YU)
65
C
NT=0
KSAVE=1
NS=1
70
NL=NL-1
NM1=NM-1
A1=PI/FLOAT(NM1)
DEL=1./(FLOAT(NM1)**2)
DO 12 I=1,NU
75
LOAD XU AND YU
C
IJ=NU+1-I
XXU=XU(IJ)
YYU=YU(IJ)
DYU=DYXU(IJ)
NN=1
80
C
FIND XLS
LISTING OF DECK: CAMTK

CARD NO.

81  DD 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))
     CK 81
85  IF (TP.LE.P2) XXL=CONS*(1.-COS(TP))
    IF (TP.GT.P2) XXL=CONS*(ATAN(SINH(TP-P2))+1.)
    IF (NN.EQ.NL) NN=NL1
     CK 85
80  DO 6 J=NN,NL1
     CK 80
72  J2=NL-J
     CK 72
60  IF (TP.GE.ABS(TL(J2)) AND TP.LE.ABS(TL(J1))) GO TO 7
     CK 60
70  CONTINUE
     CK 70
60  DELTA=TL(J2)-TL(J1)
     CK 60
90  T1=TP-GL(J1)
     CK 90
95  T2=TL(J2)+TP
     CK 95
100  YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DELTA)
     CK 100
100  -T1*DELTA/6.)+YPPL(J1)*TZ+YPPL(J2)*T1)/DELTA
     CK 100
115  YYL=YPPL(J1)*(DELTA/6.*T2*DELTA/2.*DELTA)+YPPL(J2)*(T1*DELTA/2.*DELTA)
     CK 115
120  )/DELTA
     CK 120
120  D=DSORT((XXL-XXU)**2+(YYL-YYU)**2)
     CK 120
125  IF (I.EQ.1 AND D.LE.DEL) GO TO 10
     CK 125
120  IF (D.LE.DEL) GO TO 9
     CK 120
120  COST=(YYU-YYL)/D
     CK 120
120  SINT=(XXL-XXU)/D
     CK 120
120  IF (DYU.NE.0.1E99) DU=(COST*DYU-SINT)/(SINT*DYU+COST)
     CK 120
130  IF (DYU.EQ.0.1E99 AND SINT.NE.0.0) DU=COST/SINT
     CK 130
120  IF (DYL.NE.0.1E99) DL=(COST*DYL-SINT)/(SINT*DYL+COST)
     CK 120
120  IF (DYL.EQ.0.1E99 AND SINT.EQ.0.0) DL=-0.1E99
     CK 120
120  IF (K.EQ.NS) GO TO 8
     CK 120
120  DKL=(DL-DL1)/((XXL-XP)
     CK 120
120  DU=(DU-DUP)/((XXL-XP)
     CK 120
120  IF (DU.EQ.DKL) GO TO 8
     CK 120
120  XXL=DP+DUP)/((DKL-DKL)
     CK 120
LISTING OF DECK: CAMTK

CARD NO.

121 IF (XK.LE.XP+DEL.AND.XK.GE.XXL-DEL) GO TO 11
     KSAVE=K
     XP=XXL
     DP=DU
     DLP=DL
     CONTINUE
     IF (I.GT.1) GO TO 12
     XK=XL(NL)
     KSAVE=NS
     NT=NT+1
     LX(NL)=IJ
     XLS(NT)=XK
     NS=KSAVE
     CONTINUE

135 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.))
          IF (TP.GE.ABS(TL(JZ))AND.TP.LE.ABS(TL(J1))) GO TO 16
          CONTINUE
          T1=TP-TL(J1)
          T2=TL(J2)-TP
          YYL=YPPL(J1)*((TI+T2)/2)*((TI-T2)*DELTA/6.+YPPL(J2)*((TI+T2)/2)/6.*DELTA/6.)
          YLS(I)=YYL
          XL(I)=(XU(I)+XLS(I))/2.
          YC(I)=(YU(I)+YYL)/2.
          TK(I)=0.5*SORT((XU(IJ)-XLS(I))**2+(YU(IJ)-YYL)**2)
     CONTINUE
     IF (YU(IJ).EQ.YYL) TH(I)=0.0

PAGE 4
LISTING OF DECK: CAMTK

CARD NO.

161
IF (YU(IJ)+HE.YYL) TH(I)=ATAN((XLS(I)-XU(IJ))/(YU(IJ)-YYL))
IF (TK(I).LE.0.0) GO TO 17
DYL=YPPL(J1)+(DELA/6.-T2/T2/(2.*DELA))*YPPL(J2)*T1*T1/(2.*DELA)
1=DELA/6.1*(YI(J2)-YL(J1))/DELA
165
IF (TP. LE.PI2) DELTA=CONS*Sin(TP)
IF (TP. GT.PI2) DELTA=CONS/CosH(TP-PI2)
170
IF (TP. GT.0.0) DYL=DYL/DELA
cost=(yu(ij)-yyl)/(2.*tk(i))
s.INT=(xhs(i)-xu(ij))/(2.*tk(i))
du=(cos*dyxu(ij)-sint)/(sINT*dyxu(ij)+cost)
175
TL=ABS(DYL-SINT)/(SINT*DYL+cost)
176
if (dyu(j2)+yl(j2)-6.*yl(de)) GT 1
177
IT2 GO TO 18
178
TH(I)*RAD
179
WRITE (JWRITE,45) I,XU(IJ),YU(IJ),XLS(I),YYL,XC(I),YC(I),TK(I),T1
180
181
C CONTINUE
182
C COMPUTE STARTING LOCATION OF CAMBER DISTRIBUTION (I. E.
183
C THICKNESS = 0) BY FITTING SECOND ORDER CURVE TO LAST THREE
184
C COMPUTED CAMBER LINE COORDINATES AND THEN DETERMINING
185
C INTERSECTION OF THAT CURVE WITH AIRFOIL SURFACE
186
C
187
ISYM=1
188
DO 20 I=1,5
189
IF (ABS(XU(I)-XL(I)) GT EPS) ISYM=0
190
IF (ABS(YU(I)+YL(I)) GT EPS) ISYM=0
191
20
CONTINUE
192
IF (ISYM.EQ.1) GO TO 30
193
IF (XC(NT)..LE.DEL) GO TO 31
194
X1=XC(NT)**2
195
X2=XC(NT-1)**2
196
X3=XC(NT-2)**2
197
d1=f(x1,xc(nt),1,s,x2,xc(nt-1),1,s,x3,xc(nt-2),1,s)
198
A1=f(yc(nt),xc(nt),1,s,yc(nt-1),1,s,yc(nt-2),1,s)/d
199
A2=f(x1,yc(nt),1,s,x2,yc(nt-1),1,s,x3,yc(nt-2),1,s)/d
200
A3=yc(nt)-A1*x1-A2*xc(nt)
201
NM1=NM/4
202

122
LISTING OF DECK: CAMTK

CARD NO.

201
D=XC(NT)/FLOAT(NM1)
X=O
XP=XR
YYUP=YY(1)
YYCP=(A1*X+A2)*X+A3
NM1=NM1+1
DO 27 I=2,NM1
X=X+D
IF (X.GT.CONS) GO TO 27
TP=ACOS(1.-X/CONS)
DO 21 K=2,NU
K1=K-1
K2=K
215
IF (TP.GE.TU(K1).AND.TP.LE.TU(K2)) GO TO 22
CONTINUE
22
DELTA=TU(K2)-TU(K1)
T1=TP-TU(K1)
T2=TU(K2)-TP
YYU=YPPL(K1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)*YPPL(K2)*T1**3/(6.*DE
1LTU=1L*DELTA/6.)*(YYU(K2)*T1+YYU(K1)*T2)/DELTA
DO 23 J=2,NU
J2=J-1
J1=J
225
IF (TP.GE.ABS(TL(J2))).AND.TP.LE.ABS(TL(J1))) GO TO 24
CONTINUE
24
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
1LTLU=1L*DELTA/6.)*(YYU(J2)*T1+YYL(J1)*T2)/DELTA
YYC=(A1*X+A2)*X+A3
DKC=(YYC-YYCP)/(X-XP)
DKU=(YYU-YYUP)/(X-XP)
235
IF (DKU.EQ.DKC) GO TO 25
XXU=XP+(YYCP-YYUP)/(DKU-DKC)
IF (XXU.GE.XP).AND.XKU.LE.X) GO TO 28
25
DKL=(YYL-YYLP)/(X-XP)
IF (DKL.GE.DKC) GO TO 26
XXL=XP+(YYCP-YYLP)/(DKL-DKC)
240
IF (XKLGE*XP.AND.XKLLE*X) GO TO 29

26 XP=X
YYLP=YYL
YYUP=YYU
YYCP=YYC

27 CONTINUE
GO TO 31

28 NT=NT+1
LX(NT)=0
XLS(NT)*XKU
YC(NT)=XKU
DU=(A1*XKU+A2)*XKU+A3
TK(NT)=0*
TH(NT)=ATAN(2.*A1*XKU+A2)

29 NT=NT+1
LX(NT)=0
XLS(NT)*XKU
YC(NT)=XKU
DL=(A1*XKL+A2)*XKL+A3
TK(NT)=0*
TH(NT)=ATAN(2.*A1*XKL+A2)

30 IF (XKLGE*XP.AND.*XKLLE*X) GO TO 29

XLS(NT)*XKU
YYU=YYP(U(K1))*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPUP(K2) *(T1**3/(6.*DE)

31 YLS(NT)=YYU
YC(NT)=YLS(NT)
D=ABS(ABS(DU)-ABS(YC(NT)))
TI=TH(NT)*RAD

32 WRITE (JWRITE) NT,XLS(NT),YLS(NT),XLS(NT),YLS(NT),XC(NT),YC(NT) CK 266
1,TI,TK(NT),TI,D
GO TO 31
LISTING OF DECK: CAMTK

CARD NO.

281
YLS(NT)=YWL
YC(NT)=YLS(NT)
D=ABS(ABS(DL)-ABS(YC(NT)))
T1=TH(NT)*RAD

285
WRITE (JWRITE,45) NT,XLS(NT),YLS(NT),XLS(NT),YLS(NT),XC(NT),YC(NT)

1,TK(NT),T1,D

GO TO 31

30
IF (LX(NT) .EQ. 1) GO TO 31

NT=NT+1
LX(NT)=1
XC(NT)=0,0
YC(NT)=YU(1)
XLS(NT)=0,0
YLS(NT)=YL(1)

295
TK(NT)=0,0
TH(NT)=0,0
D=0,0
WRITE (JWRITE,45) NT,XC(NT),YC(NT),XLS(NT),YLS(NT),XC(NT),YC(NT),T

1K(NT),TH(NT),D

300
C
PUNCH CAMBER AND THICKNESS DISTRIBUTIONS
C

31
IF (IPUNCH .NE. 5) GO TO 33

WRITE (1,46) TITLE
WRITE (JWRITE,41) IPUNCH,TITLE,NT

305
C
D=FLOAT(NT)
WRITE (1,42) D
C

310
DO 32 I=1,NT
J=NT+1-I
WRITE (JWRITE,43) XC(J),YC(J),TK(J),TH(J)
WRITE (1,47) XC(J),YC(J),TK(J),TH(J)
CONTINUE

315
C
PLOT CAMBER AND THICKNESS DISTRIBUTIONS
C

33
IF (KLOT .EQ. 0) RETURN

C
PLOT CAMBER

320
CALL CALPLT (4,2,2,3)
LISTING OF DECK: CANTK

CARD NO.

321 CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)
CALL CALPLT (0.,0.,-3)
CALL AXES (0.,0.,0.,20.,0.,0.05,-2.,1.,3HX/C,SIZ,-3,1)
DU=0.0

325 DO 34 I=1,NT
34 IF (ABS(YC(I)).GT.DU) DU=ABS(YC(I))
CONTINUE
D=.1
IF (DU.LE.0.02.AND.DU.GT.0.08) D=.05
IF (DU.LE.0.08.AND.DU.GT.0.04) D=.02
IF (DU.LE.0.04) D=.01
DL=-4.*D
CALL AXES (0.,0.,90.,8.,DL,-1.,0.,3HY/C,SIZ,3,2)
CALL CALPLT (0.,4.,-3)
335 XC(NT+1)=YC(NT+1)=0.0
XC(NT+2)=.05
YC(NT+2)=D
DO 35 I=1,NT
35 XU1=XC(I)/.05
36 YU1=YC(I)/D
CALL PNTPLT (XU1,YU1,22,ISIZ)
CONTINUE
34 CALL LINE (XC,YC,NT,1,0,0,4)
C PLOT THICKNESS
34 CALL CALPLT (0.,6.,-3)
CALL AXES (0.,0.,0.,20.,0.,0.05,-2.,1.,3HX/C,SIZ,-3,1)
DU=0.0

350 DO 36 I=1,NT
36 IF (ABS(TK(I)).GT.DU) DU=ABS(TK(I))
CONTINUE
D=.1
IF (DU.LE.0.06) D=.01
IF (DU.GT.0.06.AND.DU.LE.0.12) D=.02
IF (DU.GT.0.12.AND.DU.LE.0.24) D=.04
IF (DU.GT.0.24.AND.DU.LE.0.30) D=.05
CALL AXES (0.,0.,90.,6.,0.,D,-1.,0.,5HT/C,2,SIZ,5,2)
TK(NT+1)=0.0
TK(NT+2)=D
DO 37 I=1,NT
37 XU1=XC(I)/.05

126
LISTING OF DECK: CAMTK

CARD NO.
361 YU=TK(I)/D
CALL PNTPLT (XU,YU,22,ISIZ)
CONTINUE
37 CALL LINE (XC,TK,NT,100,0,0)
365 C PLOT INPUT AIRFOIL AND AIRFOIL GENERATED BY COMBINING
C THICKNESS AND CAMBER DISTRIBUTIONS
CALL CALPLT (0,8,3)
CALL AXES (0,0,0,20,0,20,0,05,-2,1,3HY/C,ISIZ,3,1)
CALL AXES (0,0,90,8,90,-3,2,05,-2,1,3HY/C,ISIZ,3,1)
370 CALL CALPLT (0,4,9,3)
XU(NU+1)=YU(NU+1)=0.0
XU(NU+2)=YU(NU+2)=1.05
CALL LINE (XU,YU,NU,100,0,0)
XU(NL+1)=YL(NL+1)=0.0
XL(NL+2)=YL(NL+2)=1.05
375 CALL LINE (XL,YL,NL,100,0,0)
DO 40 I=1,NT
IJ=LX(I)
IF (IJ.EQ.0) GO TO 38
380 XU=XR(I)/.05
YU=YL(I)/.05
XL=XS(I)/.05
YL=YIS(I)/.05
GO TO 39
385 XU=XI=XS(I)/.05
YU=YL=YS(I)/.05
390 CONTINUE
CALL PNTPLT (XU,YU,22,ISIZ)
CALL PNTPLT (XL,YL,22,ISIZ)
CONTINUE
CALL NFRAME
RETURN
41 FORMAT (1H1,5X,47HTHE FOLLOWING CAMBERLINE DATA HAVE BEEN PUNCHED,
15X,7HPIUNC++,14//5X,8A10//5X,4HNT=,14//9X,3H/C,7X,3HY/C,5X,5HT/C
2/2,5X,5HSLOPE)
42 FORMAT (10,2)
43 FORMAT (5x,4F10.6)
44 FORMAT (1H1,1X,7HTITLE--2X,8A10//32X,37HT--THICKNESS AND CAMBER DI
1STIBUTION--//4X,1H1,5X,4HUX/C,6X,4HY/C,6X,4HXL/C,6X,4HYL/C,6X,3H
2X/C,7X,3HY/C,6X,5HT/C,2/5X,5HSLOPE,10X,5HFPNR/)
CONTINUE
ISTING OF DECK: CANTK

ARD NO.

401 45 FORMAT (I5,7F10.6,F10.4,5X,F10.6)  CK 401
    46 FORMAT (RA10)                    CK 402
    47 FORMAT (4F10.6)                   CK 403
    END                                   CK 404
SUBROUTINE INTP (THETA, X, YSMO, YPPS, NP, NOSE, CHORD, TITLE, NINT, XINT, C)
1
NEW, INTR, IPUNCH)

ROUTINE TO INTERPOLATE ADDITIONAL UPPER AND LOWER SURFACE
COORDINATES

CODED BY — HARRY MORGAN NASA/LARC/TAD/AAB 1982

DIMENSION TITLE(8), THETA(1), X(1), YSMO(1), YPPS(1), XINT(1)

DIMENSION XSAV(57)

COMMON /INOUT/ JREAD, JWRITE, IPRINT

COMMON /BLK1/ PI, PI2, RAD, CONS

COMMON /HLM/ XU(100), YU(100), XL(100), YL(100), TLS(100)

STANDARD X/C COORDINATE INTERPOLATION VALUES

DATA (XSAV(I), I = 1, 57) / 0.0, 0.0025, 0.005, 0.0075, 0.01, 0.015, 0.02, 0.0225
15, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05, 0.055, 0.06, 0.065, 0.07, 0.075, 0.08, 0.085
25, 0.09, 0.095, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375
35, 0.4, 0.425, 0.45, 0.475, 0.5, 0.525, 0.55, 0.575, 0.6
45, 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
55, 1.0

IF INTR EQUAL 1, LOAD STANDARD X/C COORDINATE VALUES

IF (INTR.EQ.0) RETURN
IF (INTR.EQ.2) GO TO 2

NINT = 57
DO 1 I = 1, NINT
XINT(I) = XSAV(I)

INTERPOLATE UPPER SURFACE COORDINATES

WRITE (JWRITE, 7) TITLE
XUP = X(NP) * CHORD
XNOSE = X(NOSE) * CHORD
XLO = X(1) * CHORD
RATIO = CNEW / CHORD
LISTING OF DECK: INTP  
CARD NO.

41  DO 5 I=1,NINT
  XU(I)=XINT(I)*CHORD*XNOSE
  XL(I)=XL(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.0.0) GO TO 3
  DELTA=TAN(DELTA/CONS-1.)
  TU=PI2*ALOG(DELTA+SORT(DELTA*DELTA+1.))
  GO TO 4
50  3  TU=ACOS(1.-DELTA/CONS)
      TL=-TU
55  4  IF (TL.LT.THETA(I)) TL=THETA(I)
      IF (TU.GT.THETA(NP)) TU=THETA(NP)
      TLI=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
      CONTINUE
      WRITE (JWRITE,9) CNEW
      WRITE (JWRITE,10) TITLE
      DO 6 I=1,NINT
      TL=TL(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
      CONTINUE
5   6  C  INTERPOLATE LOWER SURFACE COORDINATES
      WRITE (JWRITE,10) TITLE
      DO 6 I=1,NINT
      TL=TL(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
      CONTINUE
      C  PUNCH COORDINATES
      IF (IPUNCH.NE.6) RETURN
      WRITE (JWRITE,11) CNEW,TITLE
      WRITE (1,12) TITLE
      WRITE (JWRITE,13) NINT
      XINT=FLOAT(NINT)  

130
CARD NO.

81
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)

85
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)

90
C
RETURN TO CALLING PROGRAM
C
RETURN
C

95
7 FORMAT (1H1,5X,9HTITLE-- ,8A10//26X,42H--UPPER SURFACE INTERPOLAT IT 95
1ED COORDINATES--//9X,1H1,10X,2HXU,13X,2HYU,11X,5HDY/DX,6X,11HD(DY/ IT 96
2DX)/DX,6X,9HCURVATURE)
8 FORMAT (I1O,2F15.6,3E15.6)
9 FORMAT (/10X,7HCORD =,F10.6)

100
10 FORMAT (1H1,5X,9HTITLE-- ,8A10//26X,42H--LOWER SURFACE INTERPOLAT IT 100
1ED COORDINATES--//9X,1H1,10X,2HXL,13X,2HYL,11X,5HDY/DX,6X,11HD(DY/ IT 101
2DX)/DX,6X,9HCURVATURE)

11 FORMAT (1H1,10X,50H THE FOLLOWING DATA HAVE BEEN PUNCHED FOR A CHORD IT 103
1D =,F10.6//3X,9HTITLE-- ,8A10)

105
12 FORMAT (8A10)
13 FORMAT (9X,4HNU =,I4)
14 FORMAT (F10.2)
15 FORMAT (9X,4HXL =,8F10.6/(9X,8F10.6))
16 FORMAT (9X,4HYU =,8F10.6/(9X,8F10.6))

110
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
SUBROUTINE COORD (THETA, YPPS, YSMO, NP, TI, YI, DYDX, DY2DX, CURV)

ROUTINE TO COMPUTE THE Y COORDINATE, DY/DX, D(DY/DX)/DX, AND
CURVATURE AT A GIVEN VALUE OF THETA

CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

DIMENSION THETA(1), YPPS(1), YSMO(1)

COMMON /BLK1/ PI, PIZ, RAD, CONS

COSH(X) = (EXP(X) + EXP(-X)) / 2.
SINH(X) = (EXP(X) - EXP(-X)) / 2.

DO 1 K=2, NP
J=K-1
IF (TI.GE.THETA(J) .AND. TI.LE.THETA(K)) GO TO 2
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.*DELTA)
1A=THETA(J)*T2+YSMO(J+1)*T1)/DELTA
YP1=YPPS(J)*(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

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)

132
<table>
<thead>
<tr>
<th>CARD NO.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>CURV=ABS(DY2DX)/{SQRT(1+DYDX**2)**3}</td>
</tr>
<tr>
<td></td>
<td>RETURN</td>
</tr>
<tr>
<td>5</td>
<td>DYDX=0.1E99</td>
</tr>
<tr>
<td></td>
<td>DY2DX=0.1E99</td>
</tr>
<tr>
<td>45</td>
<td>CURV=CONS/(DELTA*DELTA)</td>
</tr>
<tr>
<td></td>
<td>RETURN</td>
</tr>
<tr>
<td></td>
<td>END</td>
</tr>
</tbody>
</table>
LISTING OF DECK: SINH
CARD NO.

1  C   FUNCTION SINH(X)
    HYPERBOLIC SINE
    SINH=0.5*(EXP(X)-EXP(-X))
    RETURN

5  END

SH  1
SH  2
SH  3
SH  4
SH  5
LISTING OF DECK: COSH

CARD NO.

1

FUNCTION COSH(X)
C
HYPERBOLIC COSINE
COSH=0.5*(EXP(X)+EXP(-X))
RETURN

5

END
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.
LISTING OF DECK: SCALE

CARD NO.

<table>
<thead>
<tr>
<th>LINE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROGRAM SCALE(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE1)</td>
</tr>
<tr>
<td>2</td>
<td>C ITEM DESCRIPTION</td>
</tr>
<tr>
<td>3</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>4</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>5</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>6</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
<tr>
<td>7</td>
<td>C SC 1</td>
</tr>
<tr>
<td>8</td>
<td>C SC 2</td>
</tr>
<tr>
<td>9</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>10</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>11</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>12</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
<tr>
<td>13</td>
<td>C SC 1</td>
</tr>
<tr>
<td>14</td>
<td>C SC 2</td>
</tr>
<tr>
<td>15</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>16</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>17</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>18</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
<tr>
<td>19</td>
<td>C SC 1</td>
</tr>
<tr>
<td>20</td>
<td>C SC 2</td>
</tr>
<tr>
<td>21</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>22</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>23</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>24</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
<tr>
<td>25</td>
<td>C SC 1</td>
</tr>
<tr>
<td>26</td>
<td>C SC 2</td>
</tr>
<tr>
<td>27</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>28</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>29</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>30</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
<tr>
<td>31</td>
<td>C SC 1</td>
</tr>
<tr>
<td>32</td>
<td>C SC 2</td>
</tr>
<tr>
<td>33</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>34</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>35</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>36</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
<tr>
<td>37</td>
<td>C SC 1</td>
</tr>
<tr>
<td>38</td>
<td>C SC 2</td>
</tr>
<tr>
<td>39</td>
<td>C THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF</td>
</tr>
<tr>
<td>40</td>
<td>C AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED</td>
</tr>
<tr>
<td>41</td>
<td>C OUTPUT MAXIMUM THICKNESS RATIO</td>
</tr>
<tr>
<td>42</td>
<td>C CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982</td>
</tr>
</tbody>
</table>

Note: Card 3 is read N times.
LISTING OF DECK SCALE

CARD NO.

41  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
45  C*  TKEW - 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:
C*  MT NOT GREATER THAN 101  * SC 49
C*  LT NOT GREATER THAN 10  * SC 50
C*  XC MUST BE MONOTONICALLY INCREASING  * SC 51
C*  * SC 52
C*  *******************************************  SC 53
55  C  DIMENSION XC(101), YC(101), TK(101), TH(101), THETA(101), YPP(101)
1, TKEW(10), TITLE(8), VAR(4)
C  1, TKEW(10), TITLE(8), VAR(4)
C  COMMON /HLM/ WK(404,3)  SC 54
C  COMMON /BLK/ PI, PI2, RAD, CONS  SC 55
C  COMMON /INOUT/ JREAD, JWRITE, IPRINT  SC 56
C  SINH(X)=0.5*EXP(X)-EXP(-X))  SC 57
C  INITIALIZE PROGRAM CONSTANTS  SC 58
C  JWRITE=6  SC 59
C  JREAD=5  SC 60
C  IPRINT=0  SC 61
C  NTMAX=101  SC 62
C  PI=ACOS(-1.)  SC 63
C  PI2=PI/2.  SC 64
C  RAD=180./PI  SC 65
C  CONS=1./(1.+ATAN(SINH(P12)))  SC 66
C  READ AND PRINT INPUT DATA  SC 67
C  READ (JREAD,26) TITLE  SC 68
C  I  READ (JREAD,26) TITLE  SC 69
LISTING OF DECK: SCALE

CARD NO.

81 IF (EOF(JREAD)) 25,2
READ (JREAD,27) VAR
NT=IFIX(VAR(1))
IF (NT.GT.NTMAX) GO TO 24
IPLLOT=IFIX(VAR(2))
IF (IPLLOT.NE.0) IPLLOT=1
IPUNCH=IFIX(VAR(3))
IF (IPUNCH.NE.0) IPUNCH=1
IOP=IFIX(VAR(4))
WRITE (JWRITE,28) TITLE,NT,IPLLOT,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)
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 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

WRITE (JWRITE,36)
GO TO 1

C FIND MAXIMUM THICKNESS RATIO OF INPUT AIRFOIL

C

C COMPUTE THETA EQUIVALENT OF XC

C

PAGE 3
LISTING OF DECK: SCALE

CARD NO.

121 5  
CHORD*XC(NI)=XC(I)
DO 7 I=1,NI
DELTA=(XC(I)-XC(1))/CHORD
IF (DELTAE.LE.CONS) GO TO 6
DELTA=TAN(DELTAE/CONS-1.)
THETA(I)=PI2+ALOG(DELTAE*SQRT(DELTAE*DELTAE+1.))
GO TO 7
6  
THETA(I)=ACOS(1._DELTAE/CONS)
7  CONTINUE

130  
C  
FIT CUBIC SPLINE THRU TK VS THETA
CALL CUBSPL (THETA,TK,YPP,NT,WK)
C  
FIND LOCATIONS WHERE D(TK)/D(THETA) = 0.0
KRT=0
N1=NI-1
DO 12 I=1,N1
DELTA=THETA(I+1)-THETA(I)
AA=(YPP(I)-YPP(I+1))/(2.*DELTAE)
BB=(YPP(I+1)-THETA(I)-YPP(I))/DELTAE
CC=(YPP(I)-THETA(I+1))*2.-YPP(I+1)*THETA(I)**2)/(2.*DELTAE)+(YPP(I+1)
1)YPP(I)**DELTA/6.-(TK(I+1)-TK(I))/DELTAE
GP=BB*BB-4.*AA*CC
IF (GP) 12,B,8
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 9
GO TO 10
9  KRT=KRT+1
WK(KRT,1)=T1
10  KRT=KRT+1
WK(KRT,1)=T2
11  CONTINUE
12  IF (KRT.EQ.0) GO TO 16
C  
COMPUTE XC LOCATIONS WHERE D(TK)/D(THETA) = 0.0
DO 15 I=1,KRT
WK=ABS(WK(I))
IF (T1.LE.P12) WK(I,2)=CONS*(1._COS(T1))
IF (T1.GT.P12) WK(I,2)=CONS*(ATAN(SINH(T1-P12))+1.)
15  CONTINUE
160
LISTING OF DECK: SCALE

CARD NO.

161  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

165  13 CONTINUE

14    AA=THETA(J2)-WK(I,1)
    BB=WK(I,1)-THETA(J1)
    DELTA=THETA(J2)-THETA(J1)
    WK(I,3)=YPP(J1)*(AA**3/(6.*DELTA)-AA*DELTA/6.)*YPP(J2)* (BB**3/(6.)*DELTA)

170  16 CONTINUE

171  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
     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

185  C
     IF (IOP.EQ.1) COMPUTE SLOPES OF CAMBERLINE
     IF (IOP.NE.1) GO TO 21
     CALL CUBSPL (XC,YC,YPP,NT,WK)
  190  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
  195  19 DELTA=XC(NT)-XC(NT-1)
     TH(NT)=YPP(NT-1)*DELTA/6.+YPP(NT)*DELTA/3.+YC(NT-1)-YC(NT-1))/DELTA
     TH(NT)=ATAN(TH(NT))

200  C COMPUTE AND PRINT COORDINATES OF INPUT AIRFOIL
LISTING OF DECK: SCALE

CARD NO.

201  21  CALL SCTK (XC, YC, TK, TH, NT, TITLE, TKMAX, IPUNCH, IPRINT, IERR)
      IF (ERR .NE. 0) GO TO 1
      COMPUTE AND PRINT COORDINATES OF SCALED AIRFOILS

205  22  DO 22 I = 1, LT
      CALL SCTK (XC, YC, TK, TH, NT, TITLE, TKNEW(I), TKMAX, IPUNCH, IPRINT, IERR)
      IF (ERR .NE. 0) GO TO 1
      CONTINUE

210  22  GO TO 1
      READ NEXT CASE

215  24  WRITE (JWRITE, 38)
      GO TO 1
      STOP

220  25  PRINT ERROR MESSAGE
      CALL CALPLT (0, 0, 999)
      FINALIZE PLOTTING DEVICE

225  26  FORMAT (8A10)
      FORMAT (4F10.6)
      FORMAT (1M1, 57X, 14H--INPUT DATA--/5X, 7HTITLE --, 2X, 8A10/5X, 3HNT=)

230  27  1I3, 5X, 6HIMPLT = I3, 5X, 7HIPUNCH = I3, 5X, 4HIOP = I3)
      FORMAT (4F10.6)
      FORMAT (/4X, 4HX/C=, 8E15.6/8X, 8E15.6)
      FORMAT (/4X, 4HY/C=, 8E15.6/8X, 8E15.6)
      FORMAT (/2X, 6HT/C=2X, 8E15.6/8X, 8E15.6)
      FORMAT (/2X, 6HSLOPE = 8E15.6/8X, 8E15.6)
      FORMAT (F10.2)
      FORMAT (F10.2)
      FORMAT (2X, 3HLT = I3, 5X, 9HNEW T/C =, 10F12.6)
      FORMAT (/5X, 40HXC ARRAY IS NOT MONOTONICALLY INCREASING)
      FORMAT (/5X, 28H(C)MAX FOR INPUT AIRFOIL =, F10.6, 2X, 8HAT X/C =, F10.6)
LISTING OF DECK: SCALE

CARD NO.

241  38  FORMAT ('/5X,64H(T/C)MAX OF INPUT AIRFOIL WAS NOT FOUND -- CHECK YOUR INPUT DATA)

241  39  FORMAT ('/5X,35HINPUT CARD ERROR -- NOT GREATER THAN ,I4)

END
SUBROUTINE SCTK (XC, YC, TK, TH, NT, TITLE, TKNEW, TKMAX, IPUNCH, IPLLOT, IER)

THIS SUBROUTINE SCALES THE COORDINATES OF AN AIRFOIL FROM A BASIC MAXIMUM THICKNESS RATIO (TKMAX) TO A NEW MAXIMUM THICKNESS RATIO (TKNEW).

coded BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982

DIMENSION XCC(I), YCC(I), TK(I), TH(I), TITLE(8)

COMMON /HLMN/X(220), Y(220), XU(110), YU(110), XL(110), YL(110), XPRT(11), YPRT(110), TPRT(110)

COMMON /BLK1/ PI, PI2, RAD, CONS

COMMON /INOUT/ JREAD, JWRITE, IPRINT

SCALE THICKNESS AND COMPUTE UPPER AND LOWER SURFACE COORDINATES OF NEW AIRFOIL

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
  YL(I) = YC(I) - TK(I) * DELT2 * DELT1
1 CONTINUE

LOAD SURFACE COORDINATES INTO X AND Y ARRAYS

DO 2 I = 1, NT
  J = NT + 1 - I
  X(I) = XL(J)
  Y(I) = YL(J)
2 CONTINUE

N = NT
M = 1
IF (XU(1) .EQ. XL(1) .AND. YU(1) .EQ. YL(1)) M = 2
LISTING OF DECK: SCTK

CARD NO.

41
DO 3 I=M+1
  N=N+1
  X(N)=XU(I)
  Y(N)=YU(I)
3
INTERPOLATE OR EXTRAPOLATE TRAILING EDGE COORDINATES
4
IF (X(1)-X(N)) .GT. 6
  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))*(DELTA3*DELTA2-DELTA4*DELTA1)*(X(N)-X(1))+(DELTA3*DELTA2-DELTA4*DELTA1)*(DELTA2*DELTA1-DELTA1*DELTA2)*D
  2*DELTA2)*D
5
X(1)=X(N)
  GO TO 6
5
DELTA1=X(N-1)-X(N-2)
  DELTA2=X(N)-X(N-2)
  DELTA3=Y(N-1)-Y(N-2)
  DELTA4=Y(N)-Y(N-2)
  Y(N)=Y(N-2)+(X(1)-X(N-2))*(DELTA3*DELTA2-DELTA4*DELTA1)*(X(1)-X(N-2))+(DELTA3*DELTA2-DELTA4*DELTA1)*(DELTA2*DELTA1-DELTA1*DELTA2)*D
  2*DELTA2)*D
6
X(N)=X(I)
6
COMPUTE LONGEST CHORD
7
CHORD=0.0
  DO 8 I=2,N
    DELT=X(I-1)-X(I)
    IF (DELT.GT.CHORD) GO TO 7
  GO TO 8
7
CHORD=DELT
  NOSE=I
  CONTINUE
8
ADJUST COORDINATES FOR LONGEST CHORD
9
DELT=X(NOSE)
81 DO 9 I=1,N
   X(I)=(X(I)-DELT)/CHORD
9   Y(I)=Y(I)/CHORD
C  CHECK UPPER AND LOWER SURFACE X VALUES TO DETECT CROSSOVER OF PERPENDICULARS TO CAMBERLINE AND TO FIND NOSE POINT
C  DO 10 I=2,NOSE
   IF (X(I)-X(I-1)) .LT.0, CONTINUE
   J=NOSE+I
90   DO 11 I=J,N
   IF (X(I)-X(I-1)) .LT.0, CONTINUE
11   CONTINUE
C  LOAD COORDINATES INTO UPPER AND LOWER SURFACE ARRAYS
C  DO 12 I=1,NOSE
   J=NOSE+I-1
100  XL(I)=X(J)
12   YL(I)=Y(J)
   DO 13 I=NOSE,N
   J=I+1-NOSE
105  XU(J)=X(I)
13   YU(J)=Y(I)
   IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,22) I,XU(I),YU(I),XL(I),YL(I)
   IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,23) I,XL(I),YL(I)
   IF I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,24) I,XU(I),YU(I)
C  PRINT SCALED SURFACE COORDINATES
110 WRITE (JWRITE,21) TITLE,TKNEW
C  I=NU
   J=NU
   IF (NL.GT.NU) J=NL
115   DO 14 I=1,J
   IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,22) I,XU(I),YU(I),XL(I),YL(I)
   IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,23) I,XL(I),YL(I)
   IF I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,24) I,XU(I),YU(I)
14   CONTINUE
LISTING OF DECK: SCTK

CARD NO.

121 C PRINT CAMBER AND THICKNESS DISTRIBUTIONS
     C
     PRINT (JWRITE, 24) TITLE, TKNEW
     DELT = TKNEW / TKMAX
     DO 15 I = 1, NT
     XPRT(I) = XC(I) - DELT / CHORD
     YPRT(I) = YC(I) / CHORD
     TPRT(I) = TK(I) * DELT
     DELT3 = 2.0 * TPRT(I)
     DELT = TH(I) * RAD
     WRITE (JWRITE, 25) I, XPRT(I), YPRT(I), DELT, DELT3

135 C PUNCH DESIRED OUTPUT DATA
C
C
135 IF (IPUNCH.EQ.0) GO TO 16
     WRITE (JWRITE, 26) (TITLE(I), I = 1, 6), TKNEW
     WRITE (1, 27) (TITLE(I), I = 1, 6), TKNEW
     WRITE (JWRITE, 28) NU
     DELT1 = FLOAT(NU)
     WRITE (1, 29) DELT1
     WRITE (JWRITE, 30) (XU(I), I = 1, NU)
     WRITE (JWRITE, 31) (YU(I), I = 1, NU)
     WRITE (1, 32) (XU(I), YU(I), I = 1, NU)
     WRITE (JWRITE, 33) NL
     DELT1 = FLOAT(NL)
     WRITE (1, 29) DELT1
     WRITE (JWRITE, 34) (XL(I), I = 1, NL)
     WRITE (JWRITE, 35) (YL(I), I = 1, NL)
     WRITE (1, 32) (XL(I), YL(I), I = 1, NL)
     IF (TITLE.EQ.0) RETURN

150 C PLOT AIRFOIL SHAPE AND CAMBER AND THICKNESS DISTRIBUTIONS
C
C
155 C LABEL PLOT
     CALL CALPLT (2, 0, 3)
     CALL NOTATE (0, 0, 40, 44) PLOT OF AIRFOIL GENERATED BY SCALING PRO
     160 GMAP = 0, 44
     CALL NOTATE (16, 0, 40, 10H(T/C) = 0, 10)
     CALL NOTATE (20, 0, 40, TKNEW = 0, 3)
     CALL NOTATE (0, 1, 40, TITLE = 0, 80)
LISTING OF DECK: SCTK

CARD NO.

161  C
      PLOT AIRFOIL
      CALL AXES (0.,0.,0.,20.,0.,0.,0.05,-2.,1.,3HY/C.,40.,-3.,1)
      CALL AXES (0.,0.,90.,0.,0.,0.,0.05,-2.,1.,3HY/C.,40.,3.,1)
      CALL CALPLT (0.,0.,-3)
      X(N+1)=Y(N+1)=0.0
      X(N+2)=Y(N+2)=.05
      CALL LINE (X,Y,N+1,0.,0.,0.)
      CALL CALPLT (0.,0.,-3)
      CALL LINE (X,Y,N+2,0.,0.,0.)

165  C
      PLOT CAMBER DISTRIBUTION
      CALL CALPLT (0.,0.,-3)
      X(N+1)-Y(N+1)-O.0
      X(N+2)-Y(N+2)-.05
      CALL LINE (X,Y,N,1,0.,0.,0.0)

170  C
      PLOT THICKNESS DISTRIBUTION
      CALL CALPLT (0.,0.,-3)
      CALL AXES (0.,0.,90.,0.,0.,0.,0.05,-2.,1.,3HY/C.,40.,3.,1)
      DELT1=0.0
      DO 17 I=1,NT
         IF (ABS(YPR)(I).GT.DELT1) DELT1=ABS(YPR(I))
      CONTINUE
      DELT2=.1
      IF (DELT1.LE.0.2.AND.DELT1.GT.0.08) DELT2=.05
      IF (DELT1.LE.0.08.AND.DELT1.GT.0.04) DELT2=.02
      IF (DELT1.LE.0.04) DELT2=.01
      DELT1=-4.*DELT2
      CALL AXES (0.,0.,90.,0.,0.,0.,0.05,-2.,1.,3HY/C.,40.,3.,2)
      CALL CALPLT (0.,0.,-3)
      XPRT(N+1)=YPR(N+1)=0.0
      XPRT(N+2)=.05
      YPR(N+2)=DELT2
      DO 18 I=1,NT
         DELT3=XPRT(I)/.05
         DELT4=YPRT(I)/DELT2
      CONTINUE
      CALL PNTPLT (DELT3,DELT4,22,3)
      CALL LINE (XPRT,YPR,NT,1,0.,0.)
      CALL CALPLT (0.,0.,-3)
      CALL AXES (0.,0.,90.,0.,0.,0.,0.05,-2.,1.,3HY/C.,40.,3.,1)
      TPRT(N+1)=0.0
      TPRT(N+2)=.02
      CALL LINE (XPRT,YPR,NT,1,0.,0.)
      CALL PNTPLT (DELT3,DELT4,22,3)
LISTING OF DECK: SCTK

CARD NO.
201 19 CONTINUE
   CALL LINE (XPRT,TPRT,NT,1,0,0,0,0)
   CALL NFRAME
   RETURN
205 C
   PRINT ERROR MESSAGE
   C
20 WRITE (JWRITE,36) TITLE,TKNOW
   WRITE (JWRITE,37) (I,X(I),Y(I),I=1,N)
210 IERR=1
   RETURN
   C
21 FORMAT (1HI,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)
215 FORMAT (5X,I5,5X,2F10.6,5X,2F10.6)
22 FORMAT (5X,I5,30X,2F10.6)
24 FORMAT (1HI,3X,7HTITLE--2X,8A10//12X,49HCAMBER AND THICKNESS DIST
   1RIBUTIONS FOR (T/C)MAX =F7.4//34X,6HCAMBER,22X,9HTHICKNESS//9X,1H
   210X,3HX/C,11X,5HSLOPE,10X,5HT/C/2)
220 FORMAT (5X,I5,2(5X,F10.6),5X,F10.4,5X,F10.6)
226 FORMAT (1HI,10X,36HTHE FOLLOWING DATA HAVE BEEN PUNCHED///5X,9HTITL
   1E--6A10,10H(T/C)MAX =F10.6)
   SK 223
227 FORMAT (6A10,10H(T/C)MAX =F10.6)
228 FORMAT (/5X,4HNU =I4)
229 FORMAT (/5X,4HNU =I4)
30 FORMAT (/5X,4HNU =8F10.6/(9X,8F10.6))
31 FORMAT (/5X,4HNU =8F10.6/(9X,8F10.6))
32 FORMAT (2F10.6)
33 FORMAT (/5X,4HNL =I4)
34 FORMAT (/5X,4HNL =8F10.6/(9X,8F10.6))
35 FORMAT (/5X,4HNL =8F10.6/(9X,8F10.6))
36 FORMAT (1HI,3X,7HTITLE--2X,8A10//3X,38HATTEMPT TO SCALE AIRFOIL T
   10 (T/C)MAX =F7.4//2X,59HFAILED DUE TO CROSSOVER OF PERPENDICULARS
   SK 223
235 2D CAMBERLINE//9X,1HI,9X,3HX/C,13X,3HY/C)
37 FORMAT (5X,I5,5X,F10.6,5X,F10.6)
   END
SUBROUTINE CUBSPL (X, Y, YPP, N, A)
THIS SUBROUTINE FITS A CUBIC SPLINE TO A SET OF Y VS X INPUT
POINTS
CODED BY -- HARRY MORGAN NASA/LARC/TAD/AAB 1982
IN CALLING PROGRAM DIMENSION X, Y, AND YPP BY N AND A BY 2*N

DIMENSION X(1), Y(1), YPP(1), A(N, 2)
COMPUTE SECOND DERIVATIVE AT END POINTS BY FITTING
Y = A*X**2 + B*X + C TO THE LAST THREE POINTS AND SOLVE FOR A.
SECOND DERIVATIVE AT END POINT IS THEN EQUAL TO 2*A

H1 = X(2) - X(3)
H2 = X(3) - X(1)
H3 = X(1) - X(2)
YPP1 = 2.*((Y(1)*H1 + Y(2)*H2 + Y(3)*H3) / (H1*X(1)**2 + H2*X(2)**2 + H3*X(3))

1**2)

H1 = X(N-1) - X(N)
H2 = X(N) - X(N-2)
H3 = X(N-2) - X(N-1)
YPP(N) = 2.*((Y(N-2)*H1 + Y(N-1)*H2 + Y(N)*H3) / (H1*X(N-2)**2 + H2*X(N-1)**2)

1 + H3*X(N)**2)

PERFORM FORWARD ELIMINATION

A(1, 1) = 0.0
A(1, 2) = YPP(1)
N1 = N - 1
DO 1 I = 2, N1
H1 = X(I) - X(I-1)
H2 = X(I+1) - X(I)
H3 = (Y(I+1) - Y(I)) / H2 - (Y(I) - Y(I-1)) / H1
D = H1*(2.*A(I-1, 1) + 2.*H2
A(I, 1) = H2/D
1 A(I, 2) = (6.*H3 - H1*A(I-1, 2)) / D

PERFORM BACK SUBSTITUTION
LISTING OF DECK: CUBSPL

CARD NO.

41  C  J=N
    DO 2 I=2*N1
    J=J-1
45  2  YPP(J)=A(J*2)-A(J*1)*YPP(J+1)
    C  C  RETURN TO CALLING PROGRAM
    C  C  RETURN
    50  END

CB 41
CB 42
CB 43
CB 44
CB 45
CB 46
CB 47
CB 48
CB 49
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.

<table>
<thead>
<tr>
<th>CARD</th>
<th>VARIABLE</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TITLE</td>
<td>-</td>
<td>80-column title</td>
</tr>
<tr>
<td>2</td>
<td>ITER</td>
<td>-</td>
<td>Maximum number of smoothing iteratives</td>
</tr>
<tr>
<td>2</td>
<td>IPLOT</td>
<td>0</td>
<td>No plots desired</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Plot smoothed and unsmoothed ( \tilde{y} ) and smoothed ( \tilde{y}' ) and ( \tilde{y}'' ) versus ( \theta )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Plot smoothed and unsmoothed ( \tilde{y} ) versus ( \tilde{x} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Plot smoothed curvature versus ( \theta )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Plot camber and thickness distribution versus ( \tilde{x} ) (ICAMTK must equal 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>Plot combined options 1 and 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Plot combined options 1 and 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>Plot combined options 1, 2, and 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Plot combined options 1 and 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>Plot combined options 1, 2, and 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>Plot combined options 1, 2, 3, and 4</td>
</tr>
<tr>
<td>2</td>
<td>IPUNCH</td>
<td>0</td>
<td>No punched output desired</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Punch smoothed ( x, y, ) and ( w )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Punch smoothed ( \theta, \tilde{y}, ) and ( w )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Punch smoothed ( \theta, \tilde{y}', ) and ( w ) (YLTE, YNOSE, YUTE also punched)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Punch smoothed ( \theta, \tilde{y}'', ) and ( w ) (YLTE, YNOSE, YUTE also punched)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>Punch ( x_c, y_c, t/c/2, ) and ( \phi ) of camber and thickness distribution (ICAMTK must equal 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Punch interpolated ( x ) and ( y ) coordinates (INTR must equal 1 or 2)</td>
</tr>
<tr>
<td>2</td>
<td>IOP</td>
<td>0</td>
<td>Upper and lower surface ( x, y, ) and ( w ) input</td>
</tr>
<tr>
<td>CARD</td>
<td>VARIABLE</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>Upper and lower surface $\theta$, $\bar{y}$, and $w$ input</td>
<td>Upper and lower surface $\theta$, $\bar{y}'$, and $w$ input</td>
<td>Upper and lower surface $\theta$, $\bar{y}''$, and $w$ input</td>
</tr>
<tr>
<td>2</td>
<td>ICAMTK</td>
<td>0</td>
<td>Do not compute camber and thickness distribution</td>
</tr>
<tr>
<td>2</td>
<td>IBAD</td>
<td>0</td>
<td>Do not check for bad input coordinates</td>
</tr>
<tr>
<td>2</td>
<td>ITRN</td>
<td>0</td>
<td>Do not translate and rotate input coordinates</td>
</tr>
<tr>
<td>2</td>
<td>INTR</td>
<td>0</td>
<td>No coordinate interpolation desired</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Interpolate smoothed $\bar{y}$ coordinates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Interpolate smoothed $\bar{y}$ coordinates at input $x$ coordinates (must specify NINT, XINT, and CNEW quantities)</td>
</tr>
<tr>
<td>3</td>
<td>NU</td>
<td>-</td>
<td>Number of input upper surface points</td>
</tr>
<tr>
<td>4</td>
<td>XU, YU, WU</td>
<td>0</td>
<td>Upper surface $x$, $y$, and $w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Upper surface $\theta$, $\bar{y}'$, and $w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Upper surface $\theta$, $\bar{y}''$, and $w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>(card 4 must be input NU times and $x$ or $\theta$ runs from nose to trailing edge)</td>
</tr>
<tr>
<td>5</td>
<td>NL</td>
<td>-</td>
<td>Number of input lower surface points</td>
</tr>
<tr>
<td>6</td>
<td>XL, YL, WL</td>
<td>0</td>
<td>Lower surface $x$, $y$, and $w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Lower surface $\theta$, $\bar{y}'$, and $w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Lower surface $\theta$, $\bar{y}''$, and $w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>(card 6 must be input NL times and $x$ or $\theta$ runs from nose to trailing edge)</td>
</tr>
<tr>
<td>CARD</td>
<td>VARIABLE</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>7</td>
<td>YLTE, YNOSE, YUTE</td>
<td>-</td>
<td>Lower surface trailing edge, nose, and upper surface trailing-edge $\bar{y}$ coordinates (Skip this card if IOP=0 or 1)</td>
</tr>
<tr>
<td>8</td>
<td>NINT</td>
<td>-</td>
<td>Number of desired interpolation $\bar{x}$ coordinates (Skip this card if INTR = 0 or 1)</td>
</tr>
<tr>
<td>9</td>
<td>XINT</td>
<td>-</td>
<td>Interpolation $\bar{x}$ coordinates (must be input NINT times with 8 values per card, but skip if INTR = 0 or 1)</td>
</tr>
<tr>
<td>10</td>
<td>CNEW</td>
<td>-</td>
<td>Desired chord length of interpolated $\bar{x}$ and $\bar{y}$ coordinates. (must be greater than zero, but skip if INTR = 0 or 1)</td>
</tr>
</tbody>
</table>

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 \( Y/C \)), \( \theta \)-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 $\theta$-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 \( \theta \)-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 \( \phi, \) 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 \( \frac{dy}{dx} \), \( \frac{d^2y}{dx^2} \), 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.

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

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 quantities listed adjacent to the X/C, Y/C, and SLOPE headings are the \( x_C \) and \( y_C \) coordinates and local slopes \( \phi \) (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 \( x \) 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


TABLE I. - TRANSFORMATION FUNCTION AND FIRST AND SECOND DERIVATIVES

<table>
<thead>
<tr>
<th>$\theta$, deg (+/−)</th>
<th>$x/c$</th>
<th>$d(x/c)/d\theta$</th>
<th>$d^2(x/c)/d\theta^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000</td>
<td>0.00000</td>
<td>4.6278</td>
</tr>
<tr>
<td>1</td>
<td>0.00007</td>
<td>0.00808</td>
<td>4.6270</td>
</tr>
<tr>
<td>2</td>
<td>0.0028</td>
<td>0.01615</td>
<td>4.6249</td>
</tr>
<tr>
<td>3</td>
<td>0.0063</td>
<td>0.02422</td>
<td>4.6214</td>
</tr>
<tr>
<td>4</td>
<td>0.0113</td>
<td>0.03228</td>
<td>4.6165</td>
</tr>
<tr>
<td>5</td>
<td>0.0176</td>
<td>0.04033</td>
<td>4.6101</td>
</tr>
<tr>
<td>6</td>
<td>0.0254</td>
<td>0.04837</td>
<td>4.6024</td>
</tr>
<tr>
<td>7</td>
<td>0.0345</td>
<td>0.05640</td>
<td>4.5933</td>
</tr>
<tr>
<td>8</td>
<td>0.0450</td>
<td>0.06441</td>
<td>4.5827</td>
</tr>
<tr>
<td>9</td>
<td>0.0570</td>
<td>0.07239</td>
<td>4.5708</td>
</tr>
<tr>
<td>10</td>
<td>0.0703</td>
<td>0.08036</td>
<td>4.5574</td>
</tr>
<tr>
<td>11</td>
<td>0.0850</td>
<td>0.08830</td>
<td>4.5427</td>
</tr>
<tr>
<td>12</td>
<td>0.1011</td>
<td>0.09622</td>
<td>4.5266</td>
</tr>
<tr>
<td>13</td>
<td>0.1186</td>
<td>0.10410</td>
<td>4.5091</td>
</tr>
<tr>
<td>14</td>
<td>0.1375</td>
<td>0.11196</td>
<td>4.4903</td>
</tr>
<tr>
<td>15</td>
<td>0.1577</td>
<td>0.11978</td>
<td>4.4701</td>
</tr>
<tr>
<td>16</td>
<td>0.1793</td>
<td>0.12756</td>
<td>4.4445</td>
</tr>
<tr>
<td>17</td>
<td>0.2022</td>
<td>0.13530</td>
<td>4.4255</td>
</tr>
<tr>
<td>18</td>
<td>0.2265</td>
<td>0.14301</td>
<td>4.4013</td>
</tr>
<tr>
<td>19</td>
<td>0.2521</td>
<td>0.15066</td>
<td>4.3756</td>
</tr>
<tr>
<td>20</td>
<td>0.2791</td>
<td>0.15828</td>
<td>4.3487</td>
</tr>
<tr>
<td>21</td>
<td>0.3074</td>
<td>0.16584</td>
<td>4.3204</td>
</tr>
<tr>
<td>22</td>
<td>0.3370</td>
<td>0.17336</td>
<td>4.2908</td>
</tr>
<tr>
<td>23</td>
<td>0.3679</td>
<td>0.18082</td>
<td>4.2599</td>
</tr>
<tr>
<td>24</td>
<td>0.4001</td>
<td>0.18823</td>
<td>4.2277</td>
</tr>
<tr>
<td>25</td>
<td>0.4336</td>
<td>0.19558</td>
<td>4.1942</td>
</tr>
<tr>
<td>26</td>
<td>0.4684</td>
<td>0.20287</td>
<td>4.1594</td>
</tr>
<tr>
<td>27</td>
<td>0.5044</td>
<td>0.21010</td>
<td>4.1234</td>
</tr>
<tr>
<td>28</td>
<td>0.5417</td>
<td>0.21726</td>
<td>4.0861</td>
</tr>
<tr>
<td>29</td>
<td>0.5802</td>
<td>0.22436</td>
<td>4.0475</td>
</tr>
<tr>
<td>30</td>
<td>0.6200</td>
<td>0.23139</td>
<td>4.0078</td>
</tr>
<tr>
<td>31</td>
<td>0.6610</td>
<td>0.23835</td>
<td>3.9668</td>
</tr>
<tr>
<td>32</td>
<td>0.7032</td>
<td>0.24523</td>
<td>3.9246</td>
</tr>
<tr>
<td>33</td>
<td>0.7466</td>
<td>0.25205</td>
<td>3.8812</td>
</tr>
<tr>
<td>34</td>
<td>0.7912</td>
<td>0.25878</td>
<td>3.8366</td>
</tr>
<tr>
<td>35</td>
<td>0.8369</td>
<td>0.26544</td>
<td>3.7908</td>
</tr>
<tr>
<td>36</td>
<td>0.8838</td>
<td>0.27201</td>
<td>3.7439</td>
</tr>
<tr>
<td>37</td>
<td>0.9319</td>
<td>0.27851</td>
<td>3.6959</td>
</tr>
<tr>
<td>38</td>
<td>0.9810</td>
<td>0.28491</td>
<td>3.6667</td>
</tr>
<tr>
<td>39</td>
<td>1.0313</td>
<td>0.29123</td>
<td>3.5964</td>
</tr>
<tr>
<td>40</td>
<td>1.0827</td>
<td>0.29747</td>
<td>3.5451</td>
</tr>
<tr>
<td>41</td>
<td>1.1351</td>
<td>0.30361</td>
<td>3.4926</td>
</tr>
<tr>
<td>42</td>
<td>1.1887</td>
<td>0.30966</td>
<td>3.4391</td>
</tr>
<tr>
<td>43</td>
<td>1.2432</td>
<td>0.31561</td>
<td>3.3845</td>
</tr>
<tr>
<td>44</td>
<td>1.2988</td>
<td>0.32147</td>
<td>3.3289</td>
</tr>
<tr>
<td>45</td>
<td>1.3554</td>
<td>0.32723</td>
<td>3.2723</td>
</tr>
</tbody>
</table>
TABLE I. - CONTINUED

<table>
<thead>
<tr>
<th>$\theta$, deg (+/-)</th>
<th>x/c</th>
<th>$d(x/c)/d\theta$</th>
<th>$d^2(x/c)/d\theta^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>.14130</td>
<td>.33289</td>
<td>.32147</td>
</tr>
<tr>
<td>47</td>
<td>.14716</td>
<td>.33845</td>
<td>.31561</td>
</tr>
<tr>
<td>48</td>
<td>.15312</td>
<td>.34391</td>
<td>.30966</td>
</tr>
<tr>
<td>49</td>
<td>.15917</td>
<td>.34926</td>
<td>.30361</td>
</tr>
<tr>
<td>50</td>
<td>.16531</td>
<td>.35451</td>
<td>.29747</td>
</tr>
<tr>
<td>51</td>
<td>.17154</td>
<td>.35964</td>
<td>.29123</td>
</tr>
<tr>
<td>52</td>
<td>.17786</td>
<td>.36467</td>
<td>.28491</td>
</tr>
<tr>
<td>53</td>
<td>.18427</td>
<td>.36959</td>
<td>.27851</td>
</tr>
<tr>
<td>54</td>
<td>.19076</td>
<td>.37439</td>
<td>.27201</td>
</tr>
<tr>
<td>55</td>
<td>.19734</td>
<td>.37908</td>
<td>.26544</td>
</tr>
<tr>
<td>56</td>
<td>.20399</td>
<td>.38366</td>
<td>.25878</td>
</tr>
<tr>
<td>57</td>
<td>.21073</td>
<td>.38812</td>
<td>.25205</td>
</tr>
<tr>
<td>58</td>
<td>.21754</td>
<td>.39246</td>
<td>.24523</td>
</tr>
<tr>
<td>59</td>
<td>.22443</td>
<td>.39668</td>
<td>.23835</td>
</tr>
<tr>
<td>60</td>
<td>.23139</td>
<td>.40078</td>
<td>.23139</td>
</tr>
<tr>
<td>61</td>
<td>.23842</td>
<td>.40475</td>
<td>.22436</td>
</tr>
<tr>
<td>62</td>
<td>.24552</td>
<td>.40861</td>
<td>.21726</td>
</tr>
<tr>
<td>63</td>
<td>.2526</td>
<td>.41234</td>
<td>.21010</td>
</tr>
<tr>
<td>64</td>
<td>.25991</td>
<td>.41594</td>
<td>.20287</td>
</tr>
<tr>
<td>65</td>
<td>.26720</td>
<td>.41942</td>
<td>.19558</td>
</tr>
<tr>
<td>66</td>
<td>.27455</td>
<td>.42277</td>
<td>.18823</td>
</tr>
<tr>
<td>67</td>
<td>.28195</td>
<td>.42599</td>
<td>.18082</td>
</tr>
<tr>
<td>68</td>
<td>.28942</td>
<td>.42908</td>
<td>.17336</td>
</tr>
<tr>
<td>69</td>
<td>.29693</td>
<td>.43204</td>
<td>.16584</td>
</tr>
<tr>
<td>70</td>
<td>.30450</td>
<td>.43487</td>
<td>.15828</td>
</tr>
<tr>
<td>71</td>
<td>.31211</td>
<td>.43756</td>
<td>.15066</td>
</tr>
<tr>
<td>72</td>
<td>.31977</td>
<td>.44013</td>
<td>.14301</td>
</tr>
<tr>
<td>73</td>
<td>.32747</td>
<td>.44255</td>
<td>.13530</td>
</tr>
<tr>
<td>74</td>
<td>.33522</td>
<td>.44485</td>
<td>.12756</td>
</tr>
<tr>
<td>75</td>
<td>.34300</td>
<td>.44701</td>
<td>.11978</td>
</tr>
<tr>
<td>76</td>
<td>.35082</td>
<td>.44903</td>
<td>.11196</td>
</tr>
<tr>
<td>77</td>
<td>.35867</td>
<td>.45091</td>
<td>.10410</td>
</tr>
<tr>
<td>78</td>
<td>.36656</td>
<td>.45266</td>
<td>.09622</td>
</tr>
<tr>
<td>79</td>
<td>.37447</td>
<td>.45427</td>
<td>.08830</td>
</tr>
<tr>
<td>80</td>
<td>.38242</td>
<td>.45574</td>
<td>.08036</td>
</tr>
<tr>
<td>81</td>
<td>.39038</td>
<td>.45708</td>
<td>.07239</td>
</tr>
<tr>
<td>82</td>
<td>.39837</td>
<td>.45827</td>
<td>.06441</td>
</tr>
<tr>
<td>83</td>
<td>.40638</td>
<td>.45933</td>
<td>.05640</td>
</tr>
<tr>
<td>84</td>
<td>.41440</td>
<td>.46024</td>
<td>.04837</td>
</tr>
<tr>
<td>85</td>
<td>.42244</td>
<td>.46101</td>
<td>.04033</td>
</tr>
<tr>
<td>86</td>
<td>.43049</td>
<td>.46165</td>
<td>.03228</td>
</tr>
<tr>
<td>87</td>
<td>.43856</td>
<td>.46214</td>
<td>.02422</td>
</tr>
<tr>
<td>88</td>
<td>.44662</td>
<td>.46249</td>
<td>.01615</td>
</tr>
<tr>
<td>89</td>
<td>.45470</td>
<td>.46270</td>
<td>.00080</td>
</tr>
<tr>
<td>90</td>
<td>.46278</td>
<td>.46278</td>
<td>.00000</td>
</tr>
</tbody>
</table>
TABLE I. - CONTINUED

<table>
<thead>
<tr>
<th>θ, deg (+/-)</th>
<th>x/c</th>
<th>d(x/c)/dθ</th>
<th>d²(x/c)/dθ²</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>470.85</td>
<td>46270</td>
<td>-0.00807</td>
</tr>
<tr>
<td>92</td>
<td>478.93</td>
<td>46249</td>
<td>-0.01614</td>
</tr>
<tr>
<td>93</td>
<td>487.00</td>
<td>46214</td>
<td>-0.02418</td>
</tr>
<tr>
<td>94</td>
<td>495.06</td>
<td>46165</td>
<td>-0.03218</td>
</tr>
<tr>
<td>95</td>
<td>503.11</td>
<td>46102</td>
<td>-0.04013</td>
</tr>
<tr>
<td>96</td>
<td>511.15</td>
<td>46025</td>
<td>-0.04802</td>
</tr>
<tr>
<td>97</td>
<td>519.17</td>
<td>45934</td>
<td>-0.05584</td>
</tr>
<tr>
<td>98</td>
<td>527.18</td>
<td>45830</td>
<td>-0.06358</td>
</tr>
<tr>
<td>99</td>
<td>535.17</td>
<td>45712</td>
<td>-0.07122</td>
</tr>
<tr>
<td>100</td>
<td>543.14</td>
<td>45582</td>
<td>-0.07876</td>
</tr>
<tr>
<td>101</td>
<td>551.08</td>
<td>45438</td>
<td>-0.08618</td>
</tr>
<tr>
<td>102</td>
<td>559.00</td>
<td>45281</td>
<td>-0.09347</td>
</tr>
<tr>
<td>103</td>
<td>566.89</td>
<td>45111</td>
<td>-0.10063</td>
</tr>
<tr>
<td>104</td>
<td>574.74</td>
<td>44930</td>
<td>-0.10765</td>
</tr>
<tr>
<td>105</td>
<td>582.57</td>
<td>44736</td>
<td>-0.11451</td>
</tr>
<tr>
<td>106</td>
<td>590.36</td>
<td>44530</td>
<td>-0.12122</td>
</tr>
<tr>
<td>107</td>
<td>598.11</td>
<td>44313</td>
<td>-0.12775</td>
</tr>
<tr>
<td>108</td>
<td>605.83</td>
<td>44084</td>
<td>-0.13411</td>
</tr>
<tr>
<td>109</td>
<td>613.50</td>
<td>43845</td>
<td>-0.14029</td>
</tr>
<tr>
<td>110</td>
<td>621.13</td>
<td>43595</td>
<td>-0.14628</td>
</tr>
<tr>
<td>111</td>
<td>628.72</td>
<td>43334</td>
<td>-0.15208</td>
</tr>
<tr>
<td>112</td>
<td>636.26</td>
<td>43064</td>
<td>-0.15768</td>
</tr>
<tr>
<td>113</td>
<td>643.75</td>
<td>42784</td>
<td>-0.16308</td>
</tr>
<tr>
<td>114</td>
<td>651.19</td>
<td>42495</td>
<td>-0.16827</td>
</tr>
<tr>
<td>115</td>
<td>658.58</td>
<td>42197</td>
<td>-0.17326</td>
</tr>
<tr>
<td>116</td>
<td>665.92</td>
<td>41890</td>
<td>-0.17803</td>
</tr>
<tr>
<td>117</td>
<td>673.20</td>
<td>41575</td>
<td>-0.18260</td>
</tr>
<tr>
<td>118</td>
<td>680.43</td>
<td>41253</td>
<td>-0.18695</td>
</tr>
<tr>
<td>119</td>
<td>687.60</td>
<td>40923</td>
<td>-0.19108</td>
</tr>
<tr>
<td>120</td>
<td>694.72</td>
<td>40586</td>
<td>-0.19500</td>
</tr>
<tr>
<td>121</td>
<td>701.77</td>
<td>40242</td>
<td>-0.19871</td>
</tr>
<tr>
<td>122</td>
<td>708.76</td>
<td>39892</td>
<td>-0.20220</td>
</tr>
<tr>
<td>123</td>
<td>715.69</td>
<td>39537</td>
<td>-0.20548</td>
</tr>
<tr>
<td>124</td>
<td>722.56</td>
<td>39175</td>
<td>-0.20855</td>
</tr>
<tr>
<td>125</td>
<td>729.37</td>
<td>38809</td>
<td>-0.21140</td>
</tr>
<tr>
<td>126</td>
<td>736.11</td>
<td>38437</td>
<td>-0.21406</td>
</tr>
<tr>
<td>127</td>
<td>742.79</td>
<td>38062</td>
<td>-0.21650</td>
</tr>
<tr>
<td>128</td>
<td>749.40</td>
<td>37682</td>
<td>-0.21874</td>
</tr>
<tr>
<td>129</td>
<td>755.94</td>
<td>37298</td>
<td>-0.22079</td>
</tr>
<tr>
<td>130</td>
<td>762.41</td>
<td>36911</td>
<td>-0.22264</td>
</tr>
<tr>
<td>131</td>
<td>768.82</td>
<td>36521</td>
<td>-0.22430</td>
</tr>
<tr>
<td>132</td>
<td>775.16</td>
<td>36128</td>
<td>-0.22577</td>
</tr>
<tr>
<td>133</td>
<td>781.43</td>
<td>35733</td>
<td>-0.22706</td>
</tr>
<tr>
<td>134</td>
<td>787.64</td>
<td>35336</td>
<td>-0.22818</td>
</tr>
<tr>
<td>135</td>
<td>793.77</td>
<td>34937</td>
<td>-0.22911</td>
</tr>
<tr>
<td>θ, deg (+/-)</td>
<td>x/c</td>
<td>d(x/c)/dθ</td>
<td>d²(x/c)/dθ²</td>
</tr>
<tr>
<td>-------------</td>
<td>-----</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>136</td>
<td>.79983</td>
<td>.34536</td>
<td>-.22988</td>
</tr>
<tr>
<td>137</td>
<td>.80582</td>
<td>.34134</td>
<td>-.23049</td>
</tr>
<tr>
<td>138</td>
<td>.81175</td>
<td>.33732</td>
<td>-.23093</td>
</tr>
<tr>
<td>139</td>
<td>.81760</td>
<td>.33328</td>
<td>-.23123</td>
</tr>
<tr>
<td>140</td>
<td>.82338</td>
<td>.32925</td>
<td>-.23137</td>
</tr>
<tr>
<td>141</td>
<td>.82909</td>
<td>.32521</td>
<td>-.23137</td>
</tr>
<tr>
<td>142</td>
<td>.83473</td>
<td>.32117</td>
<td>-.23123</td>
</tr>
<tr>
<td>143</td>
<td>.84030</td>
<td>.31714</td>
<td>-.23096</td>
</tr>
<tr>
<td>144</td>
<td>.84580</td>
<td>.31311</td>
<td>-.23056</td>
</tr>
<tr>
<td>145</td>
<td>.85123</td>
<td>.30909</td>
<td>-.23004</td>
</tr>
<tr>
<td>146</td>
<td>.85659</td>
<td>.30508</td>
<td>-.22940</td>
</tr>
<tr>
<td>147</td>
<td>.86188</td>
<td>.30108</td>
<td>-.22865</td>
</tr>
<tr>
<td>148</td>
<td>.86710</td>
<td>.29710</td>
<td>-.22779</td>
</tr>
<tr>
<td>149</td>
<td>.87225</td>
<td>.29313</td>
<td>-.22683</td>
</tr>
<tr>
<td>150</td>
<td>.87733</td>
<td>.28918</td>
<td>-.22577</td>
</tr>
<tr>
<td>151</td>
<td>.88235</td>
<td>.28525</td>
<td>-.22462</td>
</tr>
<tr>
<td>152</td>
<td>.88729</td>
<td>.28134</td>
<td>-.22338</td>
</tr>
<tr>
<td>153</td>
<td>.89217</td>
<td>.27746</td>
<td>-.22206</td>
</tr>
<tr>
<td>154</td>
<td>.89698</td>
<td>.27359</td>
<td>-.22066</td>
</tr>
<tr>
<td>155</td>
<td>.90172</td>
<td>.26975</td>
<td>-.21919</td>
</tr>
<tr>
<td>156</td>
<td>.90639</td>
<td>.26594</td>
<td>-.21764</td>
</tr>
<tr>
<td>157</td>
<td>.91100</td>
<td>.26216</td>
<td>-.21604</td>
</tr>
<tr>
<td>158</td>
<td>.91554</td>
<td>.25840</td>
<td>-.21437</td>
</tr>
<tr>
<td>159</td>
<td>.92002</td>
<td>.25467</td>
<td>-.21264</td>
</tr>
<tr>
<td>160</td>
<td>.92443</td>
<td>.25098</td>
<td>-.21086</td>
</tr>
<tr>
<td>161</td>
<td>.92878</td>
<td>.24731</td>
<td>-.20904</td>
</tr>
<tr>
<td>162</td>
<td>.93307</td>
<td>.24368</td>
<td>-.20716</td>
</tr>
<tr>
<td>163</td>
<td>.93729</td>
<td>.24008</td>
<td>-.20525</td>
</tr>
<tr>
<td>164</td>
<td>.94145</td>
<td>.23652</td>
<td>-.20329</td>
</tr>
<tr>
<td>165</td>
<td>.94554</td>
<td>.23299</td>
<td>-.20131</td>
</tr>
<tr>
<td>166</td>
<td>.94958</td>
<td>.22949</td>
<td>-.19928</td>
</tr>
<tr>
<td>167</td>
<td>.95356</td>
<td>.22603</td>
<td>-.19724</td>
</tr>
<tr>
<td>168</td>
<td>.95747</td>
<td>.22261</td>
<td>-.19516</td>
</tr>
<tr>
<td>169</td>
<td>.96133</td>
<td>.21922</td>
<td>-.19306</td>
</tr>
<tr>
<td>170</td>
<td>.96512</td>
<td>.21587</td>
<td>-.19094</td>
</tr>
<tr>
<td>171</td>
<td>.96886</td>
<td>.21255</td>
<td>-.18881</td>
</tr>
<tr>
<td>172</td>
<td>.97254</td>
<td>.20928</td>
<td>-.18666</td>
</tr>
<tr>
<td>173</td>
<td>.97617</td>
<td>.20604</td>
<td>-.18449</td>
</tr>
<tr>
<td>174</td>
<td>.97973</td>
<td>.20284</td>
<td>-.18231</td>
</tr>
<tr>
<td>175</td>
<td>.98325</td>
<td>.19967</td>
<td>-.18013</td>
</tr>
<tr>
<td>176</td>
<td>.98670</td>
<td>.19655</td>
<td>-.17794</td>
</tr>
<tr>
<td>177</td>
<td>.99011</td>
<td>.19346</td>
<td>-.17575</td>
</tr>
<tr>
<td>178</td>
<td>.99346</td>
<td>.19041</td>
<td>-.17355</td>
</tr>
<tr>
<td>179</td>
<td>.99676</td>
<td>.18740</td>
<td>-.17135</td>
</tr>
<tr>
<td>180</td>
<td>1.00000</td>
<td>.18443</td>
<td>-.16915</td>
</tr>
</tbody>
</table>
## Table II - Sample Output for Airfoil Smoothing Program

### Page 1 Output

**--Input Data--**

<table>
<thead>
<tr>
<th>TITLE</th>
<th>GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITR</td>
<td>100</td>
</tr>
<tr>
<td>IPRINT</td>
<td>10</td>
</tr>
<tr>
<td>IPUNCH</td>
<td>1</td>
</tr>
<tr>
<td>IOP</td>
<td>0</td>
</tr>
<tr>
<td>ICAMMK</td>
<td>1</td>
</tr>
<tr>
<td>IFAD</td>
<td>1</td>
</tr>
<tr>
<td>ITYN</td>
<td>1</td>
</tr>
<tr>
<td>INT</td>
<td>2</td>
</tr>
</tbody>
</table>

### Input Data

| X    | 0.0 |
| Y    | 0.0 |
| W    | 0.0 |

| X    | 0.1 |
| Y    | 0.1 |
| W    | 0.1 |

| X    | 0.2 |
| Y    | 0.2 |
| W    | 0.2 |

| X    | 0.3 |
| Y    | 0.3 |
| W    | 0.3 |

| X    | 0.4 |
| Y    | 0.4 |
| W    | 0.4 |

| X    | 0.5 |
| Y    | 0.5 |
| W    | 0.5 |

| X    | 0.6 |
| Y    | 0.6 |
| W    | 0.6 |

| X    | 0.7 |
| Y    | 0.7 |
| W    | 0.7 |

| X    | 0.8 |
| Y    | 0.8 |
| W    | 0.8 |

| X    | 0.9 |
| Y    | 0.9 |
| W    | 0.9 |

| X    | 1.0 |
| Y    | 1.0 |
| W    | 1.0 |

### Page 2 Output

**Warning** -- Bad points have been found on the upper surface based on an edit tolerance of 0.010000

Bad point at I = 12  
X = 0.175000  
Y = 0.884000  
Replaced with Y = 0.08310

Bad point at I = 5  
X = 0.025000  
Y = 0.61700  
Replaced with Y = 0.04170
<table>
<thead>
<tr>
<th>PAGE 3 OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT COORDINATES</strong></td>
</tr>
<tr>
<td><strong>TITLE</strong> - GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>31</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>33</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>35</td>
</tr>
<tr>
<td>36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAGE 4 OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRANSLATED AND ROTATED COORDINATES</strong></td>
</tr>
<tr>
<td><strong>TITLE</strong> - GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
</tbody>
</table>

\[XNO = 0.000000, \ YNO = 0.000000, \ ANGLE = -2.49\]
<table>
<thead>
<tr>
<th>X/C</th>
<th>Y/C</th>
<th>THETA</th>
<th>YPS</th>
<th>YPPS</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>-.003650</td>
<td>-180.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.994995</td>
<td>-.002372</td>
<td>-178.46</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.989990</td>
<td>-.001393</td>
<td>-176.97</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.974983</td>
<td>.000241</td>
<td>-172.67</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.949978</td>
<td>.001532</td>
<td>-166.10</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.924979</td>
<td>.003150</td>
<td>-160.12</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.899974</td>
<td>.004902</td>
<td>-154.54</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.874969</td>
<td>.006744</td>
<td>-149.77</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.849963</td>
<td>.008686</td>
<td>-145.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.824957</td>
<td>.010628</td>
<td>-140.10</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.799951</td>
<td>.012570</td>
<td>-135.20</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.774945</td>
<td>.014512</td>
<td>-130.30</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.749939</td>
<td>.016454</td>
<td>-125.40</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.724933</td>
<td>.018406</td>
<td>-120.50</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.699927</td>
<td>.020358</td>
<td>-115.60</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.674921</td>
<td>.022310</td>
<td>-110.70</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.649915</td>
<td>.024262</td>
<td>-105.80</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.624909</td>
<td>.026214</td>
<td>-100.90</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.599903</td>
<td>.028166</td>
<td>-96.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.574897</td>
<td>.030118</td>
<td>-91.10</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.549891</td>
<td>.032070</td>
<td>-86.20</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.524885</td>
<td>.034022</td>
<td>-81.30</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.499879</td>
<td>.035974</td>
<td>-76.40</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.474873</td>
<td>.037926</td>
<td>-71.50</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.449867</td>
<td>.039878</td>
<td>-66.60</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.424861</td>
<td>.041830</td>
<td>-61.70</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.399855</td>
<td>.043782</td>
<td>-56.80</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.374849</td>
<td>.045734</td>
<td>-51.90</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.349843</td>
<td>.047686</td>
<td>-47.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.324837</td>
<td>.049638</td>
<td>-42.10</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.299831</td>
<td>.051590</td>
<td>-37.20</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.274825</td>
<td>.053542</td>
<td>-32.30</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.249819</td>
<td>.055494</td>
<td>-27.40</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.224813</td>
<td>.057446</td>
<td>-22.50</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.199807</td>
<td>.059398</td>
<td>-17.60</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.174801</td>
<td>.061350</td>
<td>-12.70</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.149795</td>
<td>.063302</td>
<td>-7.80</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.124789</td>
<td>.065254</td>
<td>-2.90</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.099783</td>
<td>.067206</td>
<td>7.10</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.074777</td>
<td>.069158</td>
<td>12.20</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.049771</td>
<td>.071110</td>
<td>17.30</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.024765</td>
<td>.073062</td>
<td>22.40</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>.000000</td>
<td>.075014</td>
<td>27.50</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

CHORD = 1.000025
### TABLE II.- CONTINUED

**PAGE 6 OUTPUT**

<table>
<thead>
<tr>
<th>TITLE-- // GA(W)-1 AIRFOIL WITH BAD COORDINATE POINTS //</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>--SUM OF SQUARES GENERATED DURING SMOOTHING PROCESS--</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

SMOOTHING PROCESS CONVERGED AFTER 84 ITERATIONS

SUM OF SQUARES FROM LEAST SQUARES CUBIC SPLINE SMOOTHING = \(7900\times10^{-6}\)

ITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND DERIVATIVE CONVERGED IN 2 ITERATIONS AND DELTA = \(1554\times10^{-3}\)
<table>
<thead>
<tr>
<th>I</th>
<th>X/C</th>
<th>Y/C</th>
<th>XSMO</th>
<th>DELTA</th>
<th>YPS</th>
<th>YPPS</th>
<th>DX/DY</th>
<th>D DY/DX/DY</th>
<th>CURVATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000000</td>
<td>0.000000</td>
<td>0.003640</td>
<td>0.003634</td>
<td>0.000000</td>
<td>0.046662</td>
<td>-0.251722</td>
<td>-0.253040</td>
<td>-0.000000</td>
</tr>
<tr>
<td>2</td>
<td>1.094060</td>
<td>0.023722</td>
<td>0.042823</td>
<td>0.042741</td>
<td>0.000000</td>
<td>0.041833</td>
<td>-0.231665</td>
<td>-0.231994</td>
<td>-0.000000</td>
</tr>
<tr>
<td>3</td>
<td>1.188120</td>
<td>0.047403</td>
<td>0.085724</td>
<td>0.085502</td>
<td>0.000000</td>
<td>0.060984</td>
<td>-0.168921</td>
<td>-0.170065</td>
<td>-0.000000</td>
</tr>
<tr>
<td>4</td>
<td>1.188120</td>
<td>0.061527</td>
<td>0.116159</td>
<td>0.116013</td>
<td>0.000000</td>
<td>0.127391</td>
<td>-0.270270</td>
<td>-0.271040</td>
<td>-0.000000</td>
</tr>
<tr>
<td>5</td>
<td>1.192490</td>
<td>0.075360</td>
<td>0.145961</td>
<td>0.145870</td>
<td>0.000000</td>
<td>0.249574</td>
<td>-0.456540</td>
<td>-0.457315</td>
<td>-0.000000</td>
</tr>
<tr>
<td>6</td>
<td>1.215630</td>
<td>0.090070</td>
<td>0.161647</td>
<td>0.161488</td>
<td>0.000000</td>
<td>-0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

**Note:** The table above is a sample of the data provided in the image, showing the coordinates and other related values. The full table contains additional rows with similar data points. The image also includes a table of airfoil points with coordinates and additional measurements.
<table>
<thead>
<tr>
<th>1</th>
<th>( \text{YSMS-C CHECK VALUE} )</th>
<th>( \text{YPPS-C CHECK VALUE} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.00000</td>
<td>-0.00969</td>
</tr>
<tr>
<td>2</td>
<td>-0.00002</td>
<td>-0.00343</td>
</tr>
<tr>
<td>3</td>
<td>-0.00000</td>
<td>-0.00447</td>
</tr>
<tr>
<td>4</td>
<td>-0.00001</td>
<td>-0.01280</td>
</tr>
<tr>
<td>5</td>
<td>-0.00001</td>
<td>-0.00263</td>
</tr>
<tr>
<td>6</td>
<td>-0.00000</td>
<td>-0.01720</td>
</tr>
<tr>
<td>7</td>
<td>-0.00002</td>
<td>-0.00338</td>
</tr>
<tr>
<td>8</td>
<td>-0.00003</td>
<td>-0.01214</td>
</tr>
<tr>
<td>9</td>
<td>-0.00002</td>
<td>-0.00235</td>
</tr>
<tr>
<td>10</td>
<td>-0.00001</td>
<td>-0.00633</td>
</tr>
<tr>
<td>11</td>
<td>-0.00001</td>
<td>-0.00664</td>
</tr>
<tr>
<td>12</td>
<td>-0.00001</td>
<td>-0.01267</td>
</tr>
<tr>
<td>13</td>
<td>-0.00001</td>
<td>-0.00543</td>
</tr>
<tr>
<td>14</td>
<td>-0.00001</td>
<td>-0.01177</td>
</tr>
<tr>
<td>15</td>
<td>-0.00001</td>
<td>-0.02626</td>
</tr>
<tr>
<td>16</td>
<td>-0.00001</td>
<td>-0.01807</td>
</tr>
<tr>
<td>17</td>
<td>-0.00001</td>
<td>-0.02977</td>
</tr>
<tr>
<td>18</td>
<td>-0.00002</td>
<td>-0.04562</td>
</tr>
<tr>
<td>19</td>
<td>-0.00004</td>
<td>-0.03987</td>
</tr>
<tr>
<td>20</td>
<td>-0.00004</td>
<td>-0.02889</td>
</tr>
<tr>
<td>21</td>
<td>-0.00003</td>
<td>-0.01241</td>
</tr>
<tr>
<td>22</td>
<td>-0.00000</td>
<td>-0.00125</td>
</tr>
<tr>
<td>23</td>
<td>-0.00000</td>
<td>-0.00594</td>
</tr>
<tr>
<td>24</td>
<td>-0.00001</td>
<td>-0.00636</td>
</tr>
<tr>
<td>25</td>
<td>-0.00000</td>
<td>-0.00322</td>
</tr>
<tr>
<td>26</td>
<td>-0.00001</td>
<td>-0.00155</td>
</tr>
<tr>
<td>27</td>
<td>-0.00002</td>
<td>-0.00774</td>
</tr>
<tr>
<td>28</td>
<td>-0.00001</td>
<td>-0.00666</td>
</tr>
<tr>
<td>29</td>
<td>-0.00000</td>
<td>-0.00442</td>
</tr>
<tr>
<td>30</td>
<td>-0.00000</td>
<td>-0.00258</td>
</tr>
<tr>
<td>31</td>
<td>-0.00001</td>
<td>-0.00161</td>
</tr>
<tr>
<td>32</td>
<td>-0.00001</td>
<td>-0.00756</td>
</tr>
<tr>
<td>33</td>
<td>-0.00001</td>
<td>-0.00461</td>
</tr>
<tr>
<td>34</td>
<td>-0.00002</td>
<td>-0.00293</td>
</tr>
<tr>
<td>35</td>
<td>-0.00003</td>
<td>-0.00523</td>
</tr>
<tr>
<td>36</td>
<td>-0.00001</td>
<td>-0.00470</td>
</tr>
<tr>
<td>37</td>
<td>-0.00000</td>
<td>-0.00479</td>
</tr>
<tr>
<td>38</td>
<td>-0.00005</td>
<td>-0.01167</td>
</tr>
<tr>
<td>39</td>
<td>-0.00005</td>
<td>-0.01470</td>
</tr>
<tr>
<td>40</td>
<td>-0.00002</td>
<td>-0.00565</td>
</tr>
<tr>
<td>41</td>
<td>-0.00001</td>
<td>-0.00735</td>
</tr>
<tr>
<td>42</td>
<td>-0.00005</td>
<td>-0.01198</td>
</tr>
<tr>
<td>43</td>
<td>-0.00001</td>
<td>-0.00394</td>
</tr>
<tr>
<td>44</td>
<td>-0.00001</td>
<td>-0.00172</td>
</tr>
<tr>
<td>45</td>
<td>-0.00002</td>
<td>-0.00328</td>
</tr>
<tr>
<td>46</td>
<td>-0.00001</td>
<td>-0.00137</td>
</tr>
<tr>
<td>47</td>
<td>-0.00001</td>
<td>-0.00030</td>
</tr>
<tr>
<td>48</td>
<td>-0.00001</td>
<td>-0.00054</td>
</tr>
<tr>
<td>49</td>
<td>-0.00000</td>
<td>-0.00016</td>
</tr>
<tr>
<td>50</td>
<td>-0.00000</td>
<td>-0.00129</td>
</tr>
<tr>
<td>51</td>
<td>-0.00000</td>
<td>-0.00010</td>
</tr>
<tr>
<td>52</td>
<td>-0.00000</td>
<td>-0.00202</td>
</tr>
<tr>
<td>53</td>
<td>-0.00000</td>
<td>-0.00024</td>
</tr>
<tr>
<td>54</td>
<td>-0.00000</td>
<td>-0.00020</td>
</tr>
<tr>
<td>55</td>
<td>-0.00000</td>
<td>-0.00031</td>
</tr>
<tr>
<td>56</td>
<td>-0.00000</td>
<td>-0.00017</td>
</tr>
<tr>
<td>57</td>
<td>-0.00000</td>
<td>-0.00012</td>
</tr>
<tr>
<td>58</td>
<td>-0.00000</td>
<td>-0.00079</td>
</tr>
<tr>
<td>59</td>
<td>-0.00000</td>
<td>-0.00142</td>
</tr>
<tr>
<td>60</td>
<td>-0.00000</td>
<td>-0.00216</td>
</tr>
<tr>
<td>61</td>
<td>-0.00001</td>
<td>-0.00217</td>
</tr>
<tr>
<td>62</td>
<td>-0.00000</td>
<td>-0.00217</td>
</tr>
<tr>
<td>63</td>
<td>-0.00000</td>
<td>-0.00079</td>
</tr>
<tr>
<td>64</td>
<td>-0.00000</td>
<td>-0.00021</td>
</tr>
<tr>
<td>65</td>
<td>-0.00000</td>
<td>-0.00021</td>
</tr>
<tr>
<td>66</td>
<td>-0.00000</td>
<td>-0.00066</td>
</tr>
<tr>
<td>67</td>
<td>-0.00000</td>
<td>-0.00062</td>
</tr>
<tr>
<td>68</td>
<td>-0.00000</td>
<td>-0.00043</td>
</tr>
<tr>
<td>69</td>
<td>-0.00000</td>
<td>-0.00286</td>
</tr>
<tr>
<td>70</td>
<td>-0.00000</td>
<td>-0.00095</td>
</tr>
<tr>
<td>71</td>
<td>-0.00000</td>
<td>-0.00218</td>
</tr>
<tr>
<td>72</td>
<td>-0.00000</td>
<td>-0.00259</td>
</tr>
<tr>
<td>73</td>
<td>-0.00000</td>
<td>-0.00272</td>
</tr>
<tr>
<td>74</td>
<td>-0.00000</td>
<td>-0.00311</td>
</tr>
<tr>
<td>75</td>
<td>-0.00000</td>
<td>-0.00382</td>
</tr>
<tr>
<td>76</td>
<td>-0.00000</td>
<td>-0.00389</td>
</tr>
<tr>
<td>77</td>
<td>-0.00000</td>
<td>-0.00238</td>
</tr>
<tr>
<td>78</td>
<td>-0.00000</td>
<td>-0.00014</td>
</tr>
<tr>
<td>79</td>
<td>-0.00000</td>
<td>-0.00094</td>
</tr>
</tbody>
</table>

\[ \text{SUM OF SQUARES} = \text{.000000} \]
### TABLE II - CONTINUED

#### PAGE 9 OUTPUT

**THE FOLLOWING DATA HAVE BEEN PUNCHED**

**IOP = 0**

**NU = 40**

<table>
<thead>
<tr>
<th>DX</th>
<th>.000000</th>
<th>.000143</th>
<th>.000491</th>
<th>.001236</th>
<th>.002168</th>
<th>.003784</th>
<th>.004976</th>
<th>.007414</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>.000076</td>
<td>.001230</td>
<td>.001928</td>
<td>.003029</td>
<td>.004171</td>
<td>.004966</td>
<td>.005584</td>
<td>.006550</td>
</tr>
<tr>
<td>DL</td>
<td>.000086</td>
<td>.000724</td>
<td>.000374</td>
<td>.000249</td>
<td>.000383</td>
<td>.001029</td>
<td>.001058</td>
<td>.001087</td>
</tr>
<tr>
<td>RL</td>
<td>.000078</td>
<td>.000706</td>
<td>.000376</td>
<td>.000262</td>
<td>.000391</td>
<td>.001030</td>
<td>.001059</td>
<td>.001088</td>
</tr>
</tbody>
</table>

#### PAGE 10 OUTPUT

**TITL--**

**GAW-1 AIRFOIL WITH BAD COORDINATE POINTS**

**THICKNESS AND CAMBER DISTRIBUTION**

<table>
<thead>
<tr>
<th>1</th>
<th>X/C</th>
<th>Y/C</th>
<th>X'L/C</th>
<th>Y'L/C</th>
<th>X'C</th>
<th>Y'C</th>
<th>T/C</th>
<th>SLOPE</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.000000</td>
<td>.000246</td>
<td>.000506</td>
<td>.001256</td>
<td>.002517</td>
<td>.003768</td>
<td>.005015</td>
<td>.007182</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>.102704</td>
<td>.124652</td>
<td>.150235</td>
<td>.175926</td>
<td>.200617</td>
<td>.225307</td>
<td>.250028</td>
<td>.275039</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>.400277</td>
<td>.605072</td>
<td>.802381</td>
<td>.999690</td>
<td>1.196999</td>
<td>1.394300</td>
<td>1.591600</td>
<td>1.788901</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.687460</td>
<td>.905961</td>
<td>1.124685</td>
<td>1.343370</td>
<td>1.562060</td>
<td>1.780760</td>
<td>1.999460</td>
<td>2.218160</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.085484</td>
<td>.900000</td>
<td>.905000</td>
<td>.900000</td>
<td>.905000</td>
<td>.900000</td>
<td>.905000</td>
<td>.900000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.000076</td>
<td>.000123</td>
<td>.000192</td>
<td>.0003029</td>
<td>.0004171</td>
<td>.0005234</td>
<td>.0006296</td>
<td>.0007358</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.000086</td>
<td>.000724</td>
<td>.000374</td>
<td>.000249</td>
<td>.000383</td>
<td>.001029</td>
<td>.001058</td>
<td>.001087</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>.000078</td>
<td>.000706</td>
<td>.000376</td>
<td>.000262</td>
<td>.000391</td>
<td>.001030</td>
<td>.001059</td>
<td>.001088</td>
<td></td>
</tr>
</tbody>
</table>

---

**PAGE 172**
### PAGE 11 OUTPUT

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

---UPPER SURFACE INTERPOLATED COORDINATES---

<table>
<thead>
<tr>
<th></th>
<th>XU</th>
<th>YU</th>
<th>DY/DX</th>
<th>D(DY/DX)/DX</th>
<th>CURVATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.008778</td>
<td>0.10000E+99</td>
<td>0.10000E+99</td>
<td>3.42209E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.01000</td>
<td>0.089074</td>
<td>-0.41726E+01</td>
<td>0.19293E+04</td>
<td>2.48476E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.02000</td>
<td>0.124016</td>
<td>-0.30102E+01</td>
<td>-0.71251E+03</td>
<td>2.23259E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.05000</td>
<td>0.194594</td>
<td>0.19236E+01</td>
<td>-0.19744E+03</td>
<td>1.93756E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.10000</td>
<td>0.273409</td>
<td>0.13238E+01</td>
<td>-0.76227E+02</td>
<td>1.67032E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.50000</td>
<td>0.559929</td>
<td>0.44951E+00</td>
<td>-0.63975E+01</td>
<td>0.16850E+00</td>
</tr>
<tr>
<td>7</td>
<td>1.00000</td>
<td>0.731472</td>
<td>0.26901E+01</td>
<td>-0.29689E+01</td>
<td>0.19136E+01</td>
</tr>
<tr>
<td>8</td>
<td>2.00000</td>
<td>0.925495</td>
<td>0.13748E+00</td>
<td>-0.87943E+00</td>
<td>0.14793E+00</td>
</tr>
<tr>
<td>9</td>
<td>3.00000</td>
<td>1.025230</td>
<td>0.66128E+00</td>
<td>-0.60911E+00</td>
<td>0.16279E+00</td>
</tr>
<tr>
<td>10</td>
<td>4.00000</td>
<td>1.060524</td>
<td>0.92650E-02</td>
<td>-0.54703E+00</td>
<td>0.31582E+00</td>
</tr>
<tr>
<td>11</td>
<td>5.00000</td>
<td>1.043058</td>
<td>0.49804E+00</td>
<td>-0.64979E+00</td>
<td>0.47366E+00</td>
</tr>
<tr>
<td>12</td>
<td>6.00000</td>
<td>0.957391</td>
<td>0.12540E+01</td>
<td>-0.87740E+00</td>
<td>0.85710E+00</td>
</tr>
<tr>
<td>13</td>
<td>7.00000</td>
<td>0.879865</td>
<td>0.20485E+01</td>
<td>-0.61051E+00</td>
<td>0.57400E+00</td>
</tr>
<tr>
<td>14</td>
<td>8.00000</td>
<td>0.780652</td>
<td>0.24829E+01</td>
<td>-0.29112E+00</td>
<td>0.26614E+00</td>
</tr>
<tr>
<td>15</td>
<td>9.00000</td>
<td>0.301692</td>
<td>0.26540E+00</td>
<td>-0.37385E+00</td>
<td>0.33756E+00</td>
</tr>
<tr>
<td>16</td>
<td>9.999682</td>
<td>0.036510</td>
<td>0.26836E+00</td>
<td>-0.35969E+00</td>
<td>0.32406E+00</td>
</tr>
</tbody>
</table>

**CHORD = 10.000000**

### PAGE 12 OUTPUT

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

---LOWER SURFACE INTERPOLATED COORDINATES---

<table>
<thead>
<tr>
<th></th>
<th>XL</th>
<th>YL</th>
<th>DY/DX</th>
<th>D(DY/DX)/DX</th>
<th>CURVATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.008778</td>
<td>0.10000E+99</td>
<td>0.10000E+99</td>
<td>3.42209E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.01000</td>
<td>-0.063026</td>
<td>-0.33526E+01</td>
<td>-0.19194E+04</td>
<td>4.48226E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.02000</td>
<td>-0.089958</td>
<td>-0.22619E+01</td>
<td>-0.68162E+03</td>
<td>4.68941E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.05000</td>
<td>-0.138530</td>
<td>-0.12409E+02</td>
<td>-0.16475E+03</td>
<td>4.07024E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.10000</td>
<td>-0.186616</td>
<td>-0.77581E+00</td>
<td>-0.52625E+02</td>
<td>2.58900E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.50000</td>
<td>-0.356231</td>
<td>-0.28377E+01</td>
<td>-0.35469E+01</td>
<td>3.15786E+01</td>
</tr>
<tr>
<td>7</td>
<td>0.80000</td>
<td>-0.428505</td>
<td>0.20576E+00</td>
<td>-0.19500E+00</td>
<td>1.83240E+01</td>
</tr>
<tr>
<td>8</td>
<td>1.00000</td>
<td>-0.466116</td>
<td>-0.17190E+00</td>
<td>-0.14790E+00</td>
<td>1.14580E+01</td>
</tr>
<tr>
<td>9</td>
<td>2.00000</td>
<td>-0.583469</td>
<td>-0.75352E+00</td>
<td>-0.66676E+00</td>
<td>0.65121E+00</td>
</tr>
<tr>
<td>10</td>
<td>3.00000</td>
<td>-0.630395</td>
<td>0.22062E+01</td>
<td>-0.46844E+00</td>
<td>0.44651E+00</td>
</tr>
<tr>
<td>11</td>
<td>4.00000</td>
<td>-0.630743</td>
<td>-0.21412E+01</td>
<td>-0.44857E+00</td>
<td>0.44826E+00</td>
</tr>
<tr>
<td>12</td>
<td>5.00000</td>
<td>-0.584799</td>
<td>-0.73057E+00</td>
<td>-0.60458E+00</td>
<td>0.59977E+00</td>
</tr>
<tr>
<td>13</td>
<td>6.00000</td>
<td>-0.478298</td>
<td>-0.14269E+01</td>
<td>-0.74850E+00</td>
<td>0.72621E+00</td>
</tr>
<tr>
<td>14</td>
<td>7.00000</td>
<td>-0.308580</td>
<td>-0.18541E+00</td>
<td>-0.12032E+00</td>
<td>0.11437E+00</td>
</tr>
<tr>
<td>15</td>
<td>8.00000</td>
<td>-0.125036</td>
<td>-0.17019E+00</td>
<td>-0.58157E+00</td>
<td>0.55719E+00</td>
</tr>
<tr>
<td>16</td>
<td>9.00000</td>
<td>-0.01376</td>
<td>-0.71375E+00</td>
<td>-0.12791E+00</td>
<td>0.12694E+00</td>
</tr>
<tr>
<td>17</td>
<td>10.00000</td>
<td>-0.036343</td>
<td>-0.25300E+00</td>
<td>-0.86583E+00</td>
<td>0.78889E+00</td>
</tr>
</tbody>
</table>
TABLE III.- SAMPLE OUTPUT FOR AIRFOIL SCALING PROGRAM

<table>
<thead>
<tr>
<th>TITLE= GA(W)-1 SMOOTHED *</th>
<th>INPUT DATA--</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT= 41</td>
<td>IPLOT= 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X/C=</th>
<th>V/C=</th>
<th>T/C/2=</th>
<th>SLOPE=</th>
<th>LT=</th>
</tr>
</thead>
<tbody>
<tr>
<td>.18200E-02</td>
<td>.902500E-02</td>
<td>.592326E+00</td>
<td>.393236E+00</td>
<td>2</td>
</tr>
<tr>
<td>.102793E+00</td>
<td>.603680E-02</td>
<td>.603360E-02</td>
<td>.613850E-02</td>
<td>.130000</td>
</tr>
<tr>
<td>.401010E+00</td>
<td>.652520E-02</td>
<td>.652970E-02</td>
<td>.653520E-02</td>
<td>.200000</td>
</tr>
<tr>
<td>.674533E+00</td>
<td>.693700E-02</td>
<td>.694050E-02</td>
<td>.694400E-02</td>
<td>.169405</td>
</tr>
<tr>
<td>.873470E+00</td>
<td>.768700E-02</td>
<td>.769050E-02</td>
<td>.769400E-02</td>
<td>.387925</td>
</tr>
</tbody>
</table>

(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)\(_{\text{MAX}}\) = .2000

<table>
<thead>
<tr>
<th>I</th>
<th>X/C</th>
<th>Y/C</th>
<th>X/C</th>
<th>Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>0.002481</td>
<td>0.000000</td>
<td>0.002481</td>
</tr>
<tr>
<td>2</td>
<td>0.001540</td>
<td>0.016339</td>
<td>0.002428</td>
<td>0.008425</td>
</tr>
<tr>
<td>3</td>
<td>0.004400</td>
<td>0.023152</td>
<td>0.007977</td>
<td>0.017262</td>
</tr>
<tr>
<td>4</td>
<td>0.011888</td>
<td>0.035353</td>
<td>0.015506</td>
<td>0.024450</td>
</tr>
<tr>
<td>5</td>
<td>0.024575</td>
<td>0.048189</td>
<td>0.020174</td>
<td>0.027946</td>
</tr>
<tr>
<td>6</td>
<td>0.037252</td>
<td>0.057165</td>
<td>0.027742</td>
<td>0.032821</td>
</tr>
<tr>
<td>7</td>
<td>0.049869</td>
<td>0.064188</td>
<td>0.037858</td>
<td>0.038124</td>
</tr>
<tr>
<td>8</td>
<td>0.074987</td>
<td>0.075064</td>
<td>0.048093</td>
<td>0.042493</td>
</tr>
<tr>
<td>9</td>
<td>1.00029</td>
<td>0.083506</td>
<td>0.059051</td>
<td>0.046403</td>
</tr>
<tr>
<td>10</td>
<td>1.25043</td>
<td>0.090425</td>
<td>0.082383</td>
<td>0.053177</td>
</tr>
<tr>
<td>11</td>
<td>1.50050</td>
<td>0.096227</td>
<td>1.06619</td>
<td>0.058773</td>
</tr>
<tr>
<td>12</td>
<td>1.75058</td>
<td>0.101134</td>
<td>1.131197</td>
<td>0.063407</td>
</tr>
<tr>
<td>13</td>
<td>2.00067</td>
<td>0.105290</td>
<td>1.155864</td>
<td>0.067239</td>
</tr>
<tr>
<td>14</td>
<td>2.25087</td>
<td>0.111761</td>
<td>1.180557</td>
<td>0.070401</td>
</tr>
<tr>
<td>15</td>
<td>3.00103</td>
<td>0.116186</td>
<td>2.05149</td>
<td>0.073007</td>
</tr>
<tr>
<td>16</td>
<td>3.50115</td>
<td>0.118486</td>
<td>2.54418</td>
<td>0.076856</td>
</tr>
<tr>
<td>17</td>
<td>4.00120</td>
<td>0.119861</td>
<td>3.03713</td>
<td>0.079188</td>
</tr>
<tr>
<td>18</td>
<td>4.50118</td>
<td>0.119240</td>
<td>3.53067</td>
<td>0.080180</td>
</tr>
<tr>
<td>19</td>
<td>5.00107</td>
<td>0.116891</td>
<td>4.02503</td>
<td>0.078968</td>
</tr>
<tr>
<td>20</td>
<td>5.50096</td>
<td>0.112631</td>
<td>4.52037</td>
<td>0.078182</td>
</tr>
<tr>
<td>21</td>
<td>5.75094</td>
<td>0.109702</td>
<td>5.01656</td>
<td>0.074963</td>
</tr>
<tr>
<td>22</td>
<td>6.00098</td>
<td>0.106187</td>
<td>5.51283</td>
<td>0.069994</td>
</tr>
<tr>
<td>23</td>
<td>6.25114</td>
<td>0.102049</td>
<td>5.76059</td>
<td>0.066783</td>
</tr>
<tr>
<td>24</td>
<td>6.50142</td>
<td>0.097283</td>
<td>6.00756</td>
<td>0.063060</td>
</tr>
<tr>
<td>25</td>
<td>6.75175</td>
<td>0.091932</td>
<td>6.25312</td>
<td>0.058842</td>
</tr>
<tr>
<td>26</td>
<td>7.00204</td>
<td>0.086063</td>
<td>6.49722</td>
<td>0.054196</td>
</tr>
<tr>
<td>27</td>
<td>7.25225</td>
<td>0.079760</td>
<td>6.74073</td>
<td>0.049212</td>
</tr>
<tr>
<td>28</td>
<td>7.50239</td>
<td>0.073100</td>
<td>6.98472</td>
<td>0.043985</td>
</tr>
<tr>
<td>29</td>
<td>7.75248</td>
<td>0.066155</td>
<td>7.22971</td>
<td>0.038604</td>
</tr>
<tr>
<td>30</td>
<td>8.00256</td>
<td>0.058983</td>
<td>7.47555</td>
<td>0.033161</td>
</tr>
<tr>
<td>31</td>
<td>8.25259</td>
<td>0.051639</td>
<td>7.72179</td>
<td>0.027777</td>
</tr>
<tr>
<td>32</td>
<td>8.50250</td>
<td>0.044177</td>
<td>7.96831</td>
<td>0.022595</td>
</tr>
<tr>
<td>33</td>
<td>8.75225</td>
<td>0.036647</td>
<td>8.21550</td>
<td>0.017767</td>
</tr>
<tr>
<td>34</td>
<td>9.00182</td>
<td>0.029101</td>
<td>8.46409</td>
<td>0.013434</td>
</tr>
<tr>
<td>35</td>
<td>9.25129</td>
<td>0.021596</td>
<td>8.71465</td>
<td>0.009118</td>
</tr>
<tr>
<td>36</td>
<td>9.50069</td>
<td>0.014189</td>
<td>8.96722</td>
<td>0.006722</td>
</tr>
<tr>
<td>37</td>
<td>9.75017</td>
<td>0.006946</td>
<td>9.22126</td>
<td>0.004591</td>
</tr>
<tr>
<td>38</td>
<td>9.99999</td>
<td>0.002703</td>
<td>9.47583</td>
<td>0.003612</td>
</tr>
<tr>
<td>39</td>
<td>9.94998</td>
<td>0.001308</td>
<td>9.72961</td>
<td>0.004394</td>
</tr>
<tr>
<td>40</td>
<td>1.00000</td>
<td>-0.00079</td>
<td>0.988032</td>
<td>0.006184</td>
</tr>
<tr>
<td>41</td>
<td>0.992999</td>
<td>-0.00081</td>
<td>0.992999</td>
<td>0.008867</td>
</tr>
<tr>
<td>42</td>
<td>1.00000</td>
<td>-0.00867</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE III. - CONCLUDED

<table>
<thead>
<tr>
<th></th>
<th>CAMBER</th>
<th>TLNCNTHNESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.002428</td>
<td>0.008425</td>
</tr>
<tr>
<td>2</td>
<td>0.003988</td>
<td>0.007390</td>
</tr>
<tr>
<td>3</td>
<td>0.008523</td>
<td>0.004555</td>
</tr>
<tr>
<td>4</td>
<td>0.012267</td>
<td>0.002397</td>
</tr>
<tr>
<td>5</td>
<td>0.019815</td>
<td>0.001266</td>
</tr>
<tr>
<td>6</td>
<td>0.031216</td>
<td>0.005023</td>
</tr>
<tr>
<td>7</td>
<td>0.042673</td>
<td>0.007336</td>
</tr>
<tr>
<td>8</td>
<td>0.054440</td>
<td>0.008892</td>
</tr>
<tr>
<td>9</td>
<td>0.078685</td>
<td>0.10944</td>
</tr>
<tr>
<td>10</td>
<td>0.103324</td>
<td>0.12366</td>
</tr>
<tr>
<td>11</td>
<td>0.128120</td>
<td>0.13509</td>
</tr>
<tr>
<td>12</td>
<td>0.152957</td>
<td>0.14494</td>
</tr>
<tr>
<td>13</td>
<td>0.177787</td>
<td>0.15366</td>
</tr>
<tr>
<td>14</td>
<td>0.202608</td>
<td>0.16142</td>
</tr>
<tr>
<td>15</td>
<td>0.252252</td>
<td>0.17453</td>
</tr>
<tr>
<td>16</td>
<td>0.301908</td>
<td>0.18499</td>
</tr>
<tr>
<td>17</td>
<td>0.351591</td>
<td>0.19333</td>
</tr>
<tr>
<td>18</td>
<td>0.401312</td>
<td>0.19997</td>
</tr>
<tr>
<td>19</td>
<td>0.451077</td>
<td>0.20529</td>
</tr>
<tr>
<td>20</td>
<td>0.500882</td>
<td>0.20964</td>
</tr>
<tr>
<td>21</td>
<td>0.550690</td>
<td>0.21319</td>
</tr>
<tr>
<td>22</td>
<td>0.575576</td>
<td>0.21459</td>
</tr>
<tr>
<td>23</td>
<td>0.600427</td>
<td>0.21563</td>
</tr>
<tr>
<td>24</td>
<td>0.625213</td>
<td>0.21603</td>
</tr>
<tr>
<td>25</td>
<td>0.649932</td>
<td>0.21543</td>
</tr>
<tr>
<td>26</td>
<td>0.674624</td>
<td>0.21360</td>
</tr>
<tr>
<td>27</td>
<td>0.699338</td>
<td>0.21039</td>
</tr>
<tr>
<td>28</td>
<td>0.724098</td>
<td>0.20578</td>
</tr>
<tr>
<td>29</td>
<td>0.748897</td>
<td>0.19970</td>
</tr>
<tr>
<td>30</td>
<td>0.773714</td>
<td>0.19189</td>
</tr>
<tr>
<td>31</td>
<td>0.798544</td>
<td>0.18194</td>
</tr>
<tr>
<td>32</td>
<td>0.823405</td>
<td>0.16936</td>
</tr>
<tr>
<td>33</td>
<td>0.848329</td>
<td>0.15371</td>
</tr>
<tr>
<td>34</td>
<td>0.873345</td>
<td>0.13465</td>
</tr>
<tr>
<td>35</td>
<td>0.898452</td>
<td>0.11189</td>
</tr>
<tr>
<td>36</td>
<td>0.923628</td>
<td>0.08502</td>
</tr>
<tr>
<td>37</td>
<td>0.948826</td>
<td>0.05289</td>
</tr>
<tr>
<td>38</td>
<td>0.973989</td>
<td>0.01276</td>
</tr>
<tr>
<td>39</td>
<td>0.998015</td>
<td>-0.01741</td>
</tr>
<tr>
<td>40</td>
<td>0.993998</td>
<td>-0.02887</td>
</tr>
<tr>
<td>41</td>
<td>0.998960</td>
<td>-0.04118</td>
</tr>
</tbody>
</table>
Figure 1. - Properties of $\theta$-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.
Figure 8. - Comparison between unsmoothed and smoothed first (YPS) and second (YPPS) derivatives for a typical airfoil.
Figure 9. - Comparison between unsmoothed and smoothed square-root of curvature for a typical airfoil.
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".

Available: NASA's Industrial Applications Centers