NASA Technical Memorandum 84666

AUG
FACILITY
DEPT.

COMPUTER PROGRAMS FOR SMOOTHING AND SCALING AIRFOIL COORDINATES

(NASA-TM-84666) COMPUTER PROGRAMS FOR SMOOTHING AND SCALING AIRFOIL COORDINATES (NASA) 187 p HC A09/MF A01 CSCL 01A

N85-31011

Unclas
G3/02 16038

Harry L. Morgan, Jr.

July 1983

PUBLICLY RELEASED
AUG 1985
LIN

Review for general release July 31, 1985

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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.

ORIGINAL PAGE IS
OF POOR QUALITY

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

ORIGINAL PAGE IS
OF POOR QUALITY

SYMBOLS

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

Subscripts:

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

ORIGINAL PAGE IS
OF POOR QUALITY

DESCRIPTION OF SMOOTHING METHOD

Smoothing Criteria

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

where $0 \leq \theta \leq \pi$. However, this transformation function stretches the x-axis in both the leading- and trailing-edge regions. For application in the multi-component analysis code this stretching at both ends of the airfoil is necessary to ensure adequate definition of the maximum suction peak in the leading-edge region and to properly satisfy the Kutta flow condition in the trailing-edge region. To smooth an airfoil does not require as much stretching of the x-axis in the trailing-edge region as in the leading-edge region because the curvature is generally considerably less near the

ORIGINAL PAGE IS
OF POOR QUALITY

trailing edge of the airfoil. The hyperbolic functions behave in a manner similar to that for trigometric functions and, after considerable trial-and-error, the following transformation equation was found that reduced the amount of trailing-edge stretching and that could be mated with the trigonometric equation (2) for the leading edge:

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

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

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

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

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

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

The first and second derivatives of equation (3) are

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

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

respectively, and of equation (5) are

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

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

respectively. At $\theta = \pi/2$, the value of equations (3), (5), (6), and (8) is equal to K and the value of equations (7) and (9) is zero

ORIGINAL PAGE IS
OF POOR QUALITY

which verifies that the leading- and trailing-edge transformation equations are continuous at the matching point. A plot of the resultant transformation function and its first and second derivatives are presented in figure 1 and tabulated in table I.

The inverse of equation (3) is

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

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

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

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

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

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

Piecewise Least-Squares Polynomial Smoothing to Determine Second Derivative

The piecewise least-squares polynomial smoothing procedure requires that the independent variable increase monotonically to prevent simultaneous smoothing of upper and lower surface coordinates. This meant simply that the airfoil had to be unwrapped around the nose, which was easily accomplished by letting the lower surface transformation function run from 0 to $-\pi$ and the upper sur-

ORIGINAL PAGE IS
OF POOR QUALITY

face function run from 0 to $+\pi$. The remaining problem associated with computing the second derivatives using the least-squares polynomial procedure was to determine the number of points to include adjacent to the coordinate point and the degree of the polynomial. To determine these two quantities, the coordinates of the well-known NACA 0012 airfoil were input and various values were tried for each quantity until a combination was found that produced the best agreement between the calculated and theoretical values of the second derivatives. The number of points adjacent to the coordinate point was found to be 3 before and 3 after for a total of 7 points, and the degree of the polynomial was found to be 4. The computer code for the piecewise least-squares polynomial smoothing procedure is contained in subroutine LSQSMO.

Least-Squares Cubic Spline Smoothing of Second Derivative

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

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

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

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

at $\theta = \theta_i$ and

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

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

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

at $\theta = \theta_i$ and

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

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

$$\bar{y}'_i = 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1} \tag{25}$$

where from a similar analysis,

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

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

and

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

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

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

the $\bar{y}$ coordinates and the first- and second-derivatives in each interval are

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

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

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

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

and

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

Camber and Thickness Distribution

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

and for the lower surfaces are

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

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

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

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

which combines to obtain the equation for the local slope

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

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

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

ORIGINAL PAGE IS
OF POOR QUALITY

input cases have been input and smoothed. A call is then made to subroutine CALPLT to finalize the plot vector file.

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

Subroutine INTER

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

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

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

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

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

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

$$\Delta_1 = \epsilon_1(x - x_{i-1}) \text{ and } \Delta_2 = \epsilon_2(x_i - x). \tag{47}$$

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

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

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

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

Subroutine INPUT

The primary functions of subroutine INPUT are to read and print the input airfoil data and to prepare the input data in the proper format for input to the smoothing program. A detailed description of the required input airfoil data and the various options available

ORIGINAL PAGE IS
OF POOR QUALITY

for plotting and punching the output data is presented in the user-guide given in Appendix C. After reading the input data from the file JREAD and writing on output file JWRITE and if desired, the next execution step is to call subroutine BADPT to check the upper and lower surface coordinates for obvious bad points. If no errors occur during the check for bad points and again if desired, subroutine TRNSRT is called to translate and rotate the input airfoil to an axis system coincident with the longest chord of the airfoil.

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

The following input quantities are defined in this subroutine:

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

Subroutine TRNSRT

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

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

Subroutine BADPT

The function of subroutine BADPT is to identify and possibly to correct input $\bar{y}$ coordinates whose corresponding interpolated values exceeds a specified tolerance. The user may execute a call to this subroutine by specifying a nonzero value for the parameter IBAD in

ORIGINAL PAGE
OF POOR QUALITY

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

Subroutine SMOXY

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

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

If the second derivatives $\bar{y}''$ and the desired trailing-edge and nose $\bar{y}$ coordinates are input (IOP=3), a corresponding set of $\bar{y}$ coordinates are computed w. subroutine YNEW and a set of first derivatives computed with spline e (3). Then, regardless of

ORIGINAL PAGE IS
OF POOR QUALITY

the input option, program execution proceeds to the iterative smoothing process. Prior to the start of this iteration cycle, a search is made of the upper and lower surface $\bar{y}$ coordinates to determine the maximum upper surface and minimum lower surface values. During each smoothing cycle, these two coordinates are heavily weighted in an attempt to insure that the maximum thickness of the final smoothed airfoil is reasonably close to that of the original input airfoil.

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

$$\text{If} \quad \Delta_{i-1} = \left[\bar{y}_N - \bar{y}_I\right]_{i-1} \tag{48}$$

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

and, if not, the Newton-Raphson formula

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

with

$$\begin{aligned} a &= \left(\frac{\bar{y}''_i - \bar{y}''_{i+1}}{2h_i}\right) \\ b &= \left(\frac{\bar{y}''_{i+1}\theta_i - \bar{y}''_i\,\theta_{i+1}}{h_i}\right) \\ c &= \left(\frac{\bar{y}''_i\,\theta^2_{i+1} - \bar{y}''_{i+1}\theta^2_i}{2h_i}\right) + \frac{h_i}{6}\left(\bar{y}''_{i+1} - \bar{y}''_i\right) - \left(\frac{\bar{y}_{i+1} - \bar{y}_i}{h_i}\right) \end{aligned} \tag{52}$$

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

Subroutine YNEW

The primary function of subroutine YNEW is to control the iterative procedure that computes a set of new $\bar{y}$ coordinates from an input set of second derivatives and desired trailing edge and nose coordinates. The new set of coordinates can be computed using two different solution approaches. For the first approach (IPT = 0), the resultant simultaneous cubic-spline equations solved are generated using the combined upper and lower surface

ORIGINAL PAGE IS
OF POOR QUALITY

second derivatives and setting the end conditions equal to the leading- and trailing-edge coordinates. The value of the first derivative at the nose will, of course, be the same whether approached from either the upper or lower surface; however, the $\bar{y}$-coordinate at the nose may differ from the desired input value. The desired input nose coordinate can be obtained by adding a small constant incremental value to the input second derivatives. This small value acts the same as a constant of integration resulting in a small stretching or shrinking of the computed $\bar{y}$ coordinates. The incremental value is determined in this subroutine using the simple iterative Newton-Raphson equation

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

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

For the second approach (IPT = 1), the resultant simultaneous cubic-spline equations solved are generated in a piecewise manner first using the lower surface second derivatives and setting the end conditions equal to the trailing-edge and nose coordinates, and then using the corresponding quantitites for the upper surface. This approach ensures, of course, that the resultant airfoil will have the desired nose coordinate; however, the slope at the nose may differ when approached from the upper and lower surfaces. Here again, like the first approach, a better match can be obtained by

ORIGINAL PAGE IS
OF POOR QUALITY

adding a small incremental value to the input second derivatives. This increment is determined using the same iterative Newton-Raphson equation as that used for the first approach except the Δx represents the difference between upper and lower surface first derivatives at the nose. Both approaches should theoretically produce the same incremental values; however, experience has shown that the convergence of the second approach is generally quicker and more stable.

The following additional parameters are used internally in this subroutine:

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

Subroutine INVY

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

where

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

$$\bar{f}_i = f_i / \bar{d}_i \tag{55}$$

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

The tridiagonal terms from equation (29) are

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

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

$$f_i = 1/h_i \tag{56}$$

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

At the ends the coefficient terms are

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

and

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

Subroutine LSQSMO

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

Subroutine CSDS

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

Subroutine PCARD

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

IPUNCH = 1 x-coordinate, smoothed y-coordinate, and weighting

IPUNCH = 2 θ-value, smoothed $\bar{y}$ coordinates, and weighting

IPUNCH = 3 θ-value, smoothed $\bar{y}'$, and weighting

IPUNCH = 4 θ-value, smoothed $\bar{y}''$, and weighting

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

Subroutine PLOTAF

The function of subroutine PLOTAF is to plot the input and smoothed $\bar{y}$ coordinates, smoothed $\bar{y}'$, and smoothed $\bar{y}''$ versus the θ values (IPLOT=1) and to plot the input and smoothed $\bar{y}$ coordinates versus the input $\bar{x}$ coordinates (IPLOT=2). All plots are scaled for postprocess plotting on the Langley 33-inch CALCOMP drum plotters. The called subroutines CALPLT, NOTATE, AXES, PNTPLT, LINE, and NFRAME are all part of the Langley plotting subroutine

ORIGINAL PAGE IS
OF POOR QUALITY

package and are available by attaching the CALCOMP direct-access library file. Prior to plotting the smoothed $\bar{y}$ and $\bar{x}$ coordinates and the smoothed $\bar{y}'$ values, additional values are interpolated at each degree of θ from -180 to +180 degrees. The ordinate axes are automatically scaled to insure that all input values will be plotted. A sample of the two types of plots generated by this subroutine is presented in figure 3 for IPLOT=1 and in figure 4 for IPLOT=2. Except for the IPLOT parameter, all other input parameters are fully defined in the description of subroutine INPUT.

Subroutine PLOTCK

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

Subroutine CAMTK

The function of subroutine CAMTK is to compute the camber and thickness distribution of the final smoothed airfoil. A detailed explanation of the method used to compute the camberline is presented in the method section of this report. The first execution

ORIGINAL PAGE IS
OF POOR QUALITY

step in the subroutine is to load the x and y coordinates and y values into separate arrays for the upper and lower surfaces from the nose to the trailing edge. The first derivatives dy/dx are then computed at each input point on the upper surface.

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

After completing the camberline search for each point on the upper surface, the next execution step is to locate the intersection of the camberline with the airfoil leading edge which is the location of zero thickness. This intersection is found by fitting a second-order polynomial to the previous three camberline coordinates and then extrapolating to find the intersection with the nose region which is defined with cubic-spline functions. The upper surface coordinates, corresponding lower surface coordinates, camberline coordinates, thickness, and slope of the camberline are printed at each step during the search for the camberline and the nose inter-

ORIGINAL PAGE IS
OF POOR QUALITY

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 (IPLOT = 4, 8, 9, or 10), is to plot the camber and thickness distribution data. A sample of the type of plot generated is presented in figure 6. The camberline coordinates are plotted at the bottom part of the figure, the half-thickness distribution at the center, and the upper and lower surface search points at the top part of the figure.

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

Subroutine INTP

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

$\bar{x}$ = 0.0, 0.00025, 0.0005, 0.00075, 0.001, 0.0015, 0.002, 0.0025, 0 005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375, 0.4, 0.425, 0.45, 0.475, 0.5, 0.525, 0.55, 0.575, 0.6, 0.625, 0.65, 0.675, 0.7, 0.725, 0.75, 0.775, 0.8, 0.825, 0.85, 0.875, 0.9, 0.925, 0.95, 0.97, 0.98, 0.99, 1.0.

ORIGINAL PAGE IS
OF POOR QUALITY

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

Subroutine COORD

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

Function Subprograms SINH and COSH

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

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

and

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

respectively.

Program SCALE

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

The next execution step is to compute the x_c location and the magnitude of the maximum value of the input half-thickness distribution. A cubic spline is fit through the input thickness data and then all locations and corresponding thickness values where the first derivative of the spline function equals zero are computed

ORIGINAL PAGE IS
OF POOR QUALITY

using equations (51) and (52). The location of the maximum value is then determined and printed on the output file. If the value of the input parameter IOP equals 1, the slope of the camberline coordinates are then computed using spline equation (31). The angular value of the slope is then obtained by computing the arctangent of the value of the first derivative.

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

Subroutine SCTK

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

ORIGINAL PAGE IS
OF POOR QUALITY

argument list for this subroutine is presented in the description of program SCALE.

Subroutine CUBSPL

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

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

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

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

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

$$\begin{aligned} e_i &= h_{i-1}/6 \\ d_i &= \frac{h_{i-1} + h_i}{3} \\ f_i &= h_i/6 \end{aligned} \tag{62}$$

ORIGINAL PAGE IS
OF POOR QUALITY

and

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

The following parameters are used in this subroutine:

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

DISCUSSION OF PROGRAM APPLICATION AND RESULTS

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

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

and making the appropriate changes in the computer code.

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

ORIGINAL PAGE IS
OF POOR QUALITY

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 i recommended that the maximum of 300 iterations be specified. The convergence criteria for this smoothing program is rather stringent; however, the smoothing process should converge or be near convergence in less than 100 iterations for most airfoils. If the process has not conveyed in 300 iterations, the resultant coordinates can be written on the output file TAPE1 in the form of either x and y coordinates or θ and $\bar{y}$ values and then input again into the smoothing program for another 300 iterations. If, during the initial smoothing attempt, the process begins to oscillate, it is suggested that fewer coordinates be selected in the region where the

ORIGINAL PAGE IS
OF POOR QUALITY

oscillation occurs and the case be resubmitted. The oscillatory region can be located by setting the IPRINT parameter in program AIRSMO equal to 0 which will generate a summary print of the computed second derivatives for each iteration.

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

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

The airfoil smoothing program has been used extensively at Langley for the past several years and has worked successfully for a

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

CONCLUDING REMARKS

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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.

ORIGINAL PAGE IS
OF POOR QUALITY

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.

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      PROGRAM AIRSMO(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
C
C     THIS PROGRAM PRESENTS A TECHNIQUE FOR SMOOTHING AIRFOIL
C     COORDINATES USING LEAST SQUARES POLYNOMIAL AND CUBIC SPLINE
C     METHODS
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION TITLE(8), XINT(100), X(200), Y(200), W(200), YSMO(200),
     1YPS(200), YPPS(202), THETA(202)
C
      COMMON /HLM/ DUMMX(2000)
C
      COMMON /SMY/ DUMMY(2,30)
C
      COMMON /BLK1/ PI,PI2,RAD,CONS
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
C     SINH(X)=0.5*(EXP(X)-EXP(-X))
C
C     INITIALIZE PROGRAM CONSTANTS
C
      PI=ACOS(-1.)
      PI2=PI/2.
      RAD=180./PI
      CONS=1./(1.+ATAN(SINH(PI2)))
      JREAD=5
      JWRITE=6
      IPRINT=1
      EPS=1.E-6
      DF=1.E-4
      REWIND 1
C
C     INITIALIZE PLOTTING DEVICE
C
      CALL PSEUDO
      CALL LEROY
C
C     READ INPUT DATA
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

  81              1NCH,TITLE)                                                  AS  81
            C                                                                  AS  82
            C      INTERPOLATE NEW COORDINATES                                 AS  83
            C                                                                  AS  84
  85                IF (INTR.GT.0) CALL INTP (THETA,X,YSMO,YPPS,NP,NOSE,CHORD,TITLE,NI AS  85
                  1NT,XINT,CNEW,INTR,IPUNCH)                                   AS  86
            C                                                                  AS  87
            C      RETURN AND READ NEXT CASE                                   AS  88
            C                                                                  AS  89
  90                GO TO 1                                                    AS  90
            C                                                                  AS  91
            C      FINALIZE PLOTTING DEVICE                                    AS  92
            C                                                                  AS  93
            5     CALL CALPLT (0.,0.,999)                                      AS  94
  95                WRITE (JWRITE,6)                                           AS  95
                    END FILE 1                                                 AS  96
                    REWIND 1                                                   AS  97
                    STOP                                                       AS  98
            C                                                                  AS  99
 100        6      FORMAT (1H1////48X,38H-- THE LAST CASE HAS BEEN PROCESSED --)  AS 100
                    END                                                        AS 101-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      GO TO 6                                                           IP  41
    4 JJ=2                                                              IP  42
      J=N-1                                                             IP  43
      GO TO 6                                                           IP  44
    5 JJ=1                                                              IP  45
      J=3                                                               IP  46
    6 K=J-1                                                             IP  47
      M=J-2                                                             IP  48
      L=J+1                                                             IP  49
C        INTERPOLATE A YINT VALUE (YSL) BY FITTING A STRAIGHT LINE      IP  50
C        BETWEEN K AND J                                                IP  51
      D1=XINT-X(M)                                                      IP  52
      D2=XINT-X(K)                                                      IP  53
      D3=XINT-X(J)                                                      IP  54
      D=(XINT-X(K))/(X(J)-X(K))                                         IP  55
      YSL=D*Y(J)+(1.0-D)*Y(K)                                           IP  56
C        INTERPOLATE A YINT VALUE (YP1) BY FITTING A QUADRATIC BETWEEN  IP  57
C        M, K, AND J                                                    IP  58
      C1=D3*D2/((X(M)-X(K))*(X(M)-X(J)))                                IP  59
      C2=D1*D3/((X(K)-X(M))*(X(K)-X(J)))                                IP  60
      C3=D2*D1/((X(J)-X(M))*(X(J)-X(K)))                                IP  61
      YP1=C1*Y(M)+C2*Y(K)+C3*Y(J)                                       IP  62
C        INTERPOLATE A YINT VALUE (YP2) BY FITTING A QUADRATIC BETWEEN  IP  63
C        K, J, AND L                                                    IP  64
      D4=XINT-X(L)                                                      IP  65
      C1=D4*D3/((X(K)-X(J))*(X(K)-X(L)))                                IP  66
      C2=D2*D4/((X(J)-X(K))*(X(J)-X(L)))                                IP  67
      C3=D3*D2/((X(L)-X(K))*(X(L)-X(J)))                                IP  68
      YP2=C1*Y(K)+C2*Y(J)+C3*Y(L)                                       IP  69
C                                                                       IP  70
      IF (JJ-2) 7,8,9                                                   IP  71
    7 YP2=YP1                                                           IP  72
      D=(XINT-X(1))/(X(2)-X(1))                                         IP  73
      YSL=D*Y(2)+(1.0-D)*Y(1)                                           IP  74
      GO TO 9                                                           IP  75
    8 YP1=YP2                                                           IP  76
      D=(XINT-X(N-1))/(X(N)-X(N-1))                                     IP  77
      YSL=D*Y(N)+(1.0-D)*Y(N-1)                                         IP  78
C        COMPUTE DEVIATION BETWEEN LINEAR AND QUADRATIC YINT VALUES     IP  79
    9 DEV1=ABS(YP1-YSL)                                                 IP  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DEV2=ABS(YP2-YSL)                                                 IP  81
      IF (DEV1+DEV2) 10,10,11                                           IP  82
   10 YINT=YSL                                                          IP  83
      RETURN                                                            IP  84
C        COMPUTE WEIGHTING FACTORS                                      IP  85
   11 WT2=(DEV1*D)/(DEV1*D+(1.0-D)*DEV2)                                IP  86
      WT1=1.0-WT2                                                       IP  87
C        COMPUTE FINAL YINT                                             IP  88
      YINT=WT2*YP2+WT1*YP1                                              IP  89
      RETURN                                                            IP  90
   12 YINT=Y(N)                                                         IP  91
      JEND=N                                                            IP  92
      RETURN                                                            IP  93
   13 YINT=Y(JSTART)                                                    IP  94
      RETURN                                                            IP  95
C                                                                       IP  96
C     LINEAR INTERPOLATION METHOD                                       IP  97
C                                                                       IP  98
   14 DO 15 L=JSTART,N                                                  IP  99
      J=L                                                               IP 100
      D1=SGN*(X(J)-XINT)                                                IP 101
      IF (D1) 15,2,16                                                   IP 102
   15 JEND=J                                                            IP 103
   16 YINT=Y(J-1)+(Y(J)-Y(J-1))*(XINT-X(J-1))/(X(J)-X(J-1))             IP 104
      RETURN                                                            IP 105
      END                                                               IP 106-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE INPUT (TITLE,ITER,IPLOT,IPUNCH,IOP,ICAMTK,INTR,YLTE,YNO IU
     1SE,YUTE,NINT,XINT,CNEW,NP,X,Y,W,THETA,YPS,YPPS,NOSE,CHORD,IERR)    IU
C                                                                        IU
C     ROUTINE TO READ INPUT DATA FOR AIRFOIL SMOOTHING PROGRAM           IU
C                                                                        IU
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          IU
C                                                                        IU
C*************************************************************************IU
C*                                                                      * IU
C*          DESCRIPTION OF INPUT CARDS FOR SMOOTHING PROGRAM            * IU
C*                                                                      * IU
C* CARD NUMBER                    DESCRIPTION                           * IU
C*                                                                      * IU
C*......................................................................* IU
C*   1   FORMAT(8A10)                                                   * IU
C*            TITLE CARD                                                * IU
C*......................................................................* IU
C*   2   FORMAT(8F10.0)                                                 * IU
C*            ITER - MAXIMUM NUMBER OF SMOOTHING ITERATIONS             * IU
C*            IPLOT - PLOTTING OPTION                                   * IU
C*                    0 - NO PLOTS                                      * IU
C*                    1 - PLOT SMOOTHED AND UNSMOOTHED Y/C, SMOOTHED    * IU
C*                        YPS, AND SMOOTHED YPPS VS THETA               * IU
C*                    2 - PLOT SMOOTHED AND UNSMOOTHED Y/C VS X/C       * IU
C*                    3 - PLOT SMOOTHED CURVATURE VS THETA              * IU
C*                    4 - PLOT CAMBER AND THICKNESS DISTRIBUTION        * IU
C*                    5 - PLOT OPTIONS 1 AND 2                          * IU
C*                    6 - PLOT OPTIONS 1 AND 3                          * IU
C*                    7 - PLOT OPTIONS 1, 2, AND 3                      * IU
C*                    8 - PLOT OPTIONS 1 AND 4                          * IU
C*                    9 - PLOT OPTIONS 1, 2, AND 4                      * IU
C*                   10 - PLOT OPTIONS 1, 2, 3, AND 4                   * IU
C*            IPUNCH - PUNCH OUTPUT OPTION                              * IU
C*                     0 - NO PUNCHED OUTPUT                            * IU
C*                     1 - SMOOTHED (X,Y,W) PUNCHED                     * IU
C*                     2 - SMOOTHED (THETA,Y/C,W) PUNCHED               * IU
C*                     3 - SMOOTHED (THETA,YPS,W) PUNCHED (YLTE,        * IU
C*                         YNOSE, AND YUTE ALSO PUNCHED)                * IU
C*                     4 - SMOOTHED (THETA,YPPS,W) PUNCHED (YLTE,       * IU
C*                         YNOSE, AND YUTE ALSO PUNCHED)                * IU
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

C*                    5 - THICKNESS AND CAMBER DISTRIBUTION (X/C,     * IU
C*                        Y/C, T/C/2, AND SLOPE) PUNCHED              * IU
C*                    6 - INTERPOLATED COORDINATES PUNCHED            * IU
C*            IOP - INPUT DATA OPTION                                 * IU
C*                  0 - (X,Y,W) INPUT                                 * IU
C*                  1 - (THETA,Y/C,W) INPUT                           * IU
C*                  2 - (THETA,YPS,W) INPUT                           * IU
C*                  3 - (THETA,YPPS,W) INPUT                          * IU
C*            ICAMTK - THICKNESS AND CAMBER DISTRIBUTION OPTION       * IU
C*                    0 - DO NOT COMPUTE THICKNESS AND CAMBER         * IU
C*                    1 - COMPUTE THICKNESS AND CAMBER                * IU
C*            IBAD - BAD COORDINATE CHECK OPTION                      * IU
C*                   0 - DO NOT CHECK FOR BAD COORDINATES             * IU
C*                   1 - CHECK FOR BAD COORDINATES                    * IU
C*            ITRN - INPUT COORDINATE TRANSLATION AND ROTATION OPTION * IU
C*                   0 - DO NOT TRANSLATE AND ROTATE                  * IU
C*                   1 - TRANSLATE AND ROTATE SO THAT X-AXIS          * IU
C*                       CORRESPONDS TO THE LONGEST CHORDLINE         * IU
C*            INTR - COORDINATE INTERPOLATION OPTION                  * IU
C*                   0 - NO INTERPOLATION DESIRED                     * IU
C*                   1 - INTERPOLATE NEW COORDINATES USING STANDARD 57 * IU
C*                       X/C COORDINATES DEFINED IN SUBROUTINE INTP   * IU
C*                   2 - INTERPOLATE NEW COORDINATES AT INPUT X/C     * IU
C*                       VALUES  (0.0 .GE. X/C .LE. 1.0)              * IU
C*.......................................................................* IU
C*   3   FORMAT(I0.0)                                                  * IU
C*            NU - NUMBER OF UPPER SURFACE INPUT COORDINATES           * IU
C*.......................................................................* IU
C*   4   FORMAT(3F10.0)                                                * IU
C*            XU,YU,WU - UPPER SURFACE INPUT COORDINATES AND WEIGHTING * IU
C*                       (NU CARDS ARE INPUT)                          * IU
C*                     IF IOP=0, XU=X AND YU=Y COORDINATES             * IU
C*                     IF IOP=1, XU=THETA AND YU=Y/C                   * IU
C*                     IF IOP=2, XU=THETA AND YU=YPS                   * IU
C*                     IF IOP=3, XU=THETA AND YU=YPPS                  * IU
C*                     FOR ALL IOP, WU=WEIGHTING FACTOR                * IU
C*.......................................................................* IU
C*   5   FORMAT(I0.0)                                                  * IU
C*            NL - NUMBER OF LOWER SURFACE INPUT COORDINATES           * IU
C*.......................................................................* IU
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C*    6   FORMAT(3F10.0)                                                  * IU  81
C*              XL,YL,WL - LOWER SURFACE INPUT COORDINATES AND WEIGHTING * IU  82
C*                         (NL CARDS ARE INPUT)                          * IU  83
C*                       IF IOP=0, XL=X AND YL=Y COORDINATES             * IU  84
C*                       IF IOP=1, XL=THETA AND YL=Y/C                   * IU  85
C*                       IF IOP=2, XL=THETA AND YL=YPS                   * IU  86
C*                       IF IOP=3, XL=THETA AND YL=YPPS                  * IU  87
C*                       FOR ALL IOP, WL=WEIGHTING FACTOR                * IU  88
C*.......................................................................* IU  89
C*    7   FORMAT(3F10.0)        SKIP IF IOP=0 OR 1                      * IU  90
C*              YLTE,YNOSE,YUTE - LOWER SURFACE TRAILING-EDGE, NOSE,    * IU  91
C*                                AND UPPER SURFACE TRAILING-EDGE       * IU  92
C*                                Y/C COORDINATES                       * IU  93
C*.......................................................................* IU  94
C*    8   FORMAT(F10.0)         SKIP IF INTR=0 OR 1                     * IU  95
C*              NINT - NUMBER OF INTERPOLATION X/C COORDINATES          * IU  96
C*.......................................................................* IU  97
C*    9   FORMAT(8F10.0)        SKIP IF INTR=0 OR 1                     * IU  98
C*              XINT - INTERPOLATION X/C COORDINATES (NINT VALUES INPUT) * IU  99
C*.......................................................................* IU 100
C*   10   FORMAT(F10.0)         SKIP IF INTR=0                          * IU 101
C*              CNEW - DESIRED CHORD LENGTH OF INTERPOLATED COORDINATES * IU 102
C*.......................................................................* IU 103
C*                                                                       * IU 104
C*         RESTRICTIONS:                                                 * IU 105
C*               ITER NOT GREATER THAN 300                               * IU 106
C*               NU OR NL NOT GREATER THAN 100                           * IU 107
C*               NINT NOT GREATER THAN 100                               * IU 108
C*                                                                       * IU 109
C*********************************************************************** IU 110
C                                                                         IU 111
      DIMENSION VAR(8), TITLE(8), XINT(1), X(1), Y(1), W(1), THETA(1), Y  IU 112
     1PS(1), YPPS(1)                                                      IU 113
C                                                                         IU 114
      COMMON /SMY/ XU(100),YU(100),WU(100),XL(100),YL(100),WL(100)        IU 115
C                                                                         IU 116
      COMMON /BLK1/ PI,PI2,RAD,CONS                                       IU 117
C                                                                         IU 118
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                  IU 119
C                                                                         IU 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      C     SINH(X)=(EXP(X)-EXP(-X))/2.                                IU 121
      C                                                                IU 122
      C     INITIALIZE ROUTINE CONSTANTS                               IU 123
      C                                                                IU 124
            ITRMAX=300                                                 IU 125
            NMAX=100                                                   IU 126
            TOLP=1.E-2                                                 IU 127
            IERR=0                                                     IU 128
      C                                                                IU 129
      C     READ AND PRINT INPUT DATA                                  IU 130
      C                                                                IU 131
      C        READ AND WRITE TITLE                                    IU 132
            READ (JREAD,27) TITLE                                      IU 133
            IF (EOF(JREAD)) 25,1                                       IU 134
      1     WRITE (JWRITE,28) TITLE                                    IU 135
      C        READ AND WRITE OPTIONS                                  IU 136
            READ (JREAD,29) VAR                                        IU 137
            ITER=IFIX(VAR(1))                                          IU 138
            IPLOT=IFIX(VAR(2))                                         IU 139
            IPUNCH=IFIX(VAR(3))                                        IU 140
            IOP=IFIX(VAR(4))                                           IU 141
            ICAMTK=IFIX(VAR(5))                                        IU 142
            IBAD=IFIX(VAR(6))                                          IU 143
            ITRN=IFIX(VAR(7))                                          IU 144
            INTR=IFIX(VAR(8))                                          IU 145
      C              CHECK LIMITS OF OPTIONS                           IU 146
            IF (ITER.GT.ITRMAX) ITER=ITRMAX                            IU 147
            IF (IPLOT.GT.10) IPLOT=0                                   IU 148
            IF (IPUNCH.GT.6) IPUNCH=0                                  IU 149
            IF (IOP.GT.3) GO TO 23                                     IU 150
            IF (ICAMTK.NE.0) ICAMTK=1                                  IU 151
            IF (IBAD.NE.0) IBAD=1                                      IU 152
            IF (ITRN.NE.0) ITRN=1                                      IU 153
            IF (INTR.GT.2) INTR=0                                      IU 154
            WRITE (JWRITE,30) ITER,IPLOT,IPUNCH,IOP,ICAMTK,IBAD,ITRN,INTR  IU 155
      C        READ AND WRITE NUMBER OF UPPER SURFACE INPUT POINTS     IU 156
            READ (JREAD,29) VAR(1)                                     IU 157
            NU=IFIX(VAR(1))                                            IU 158
            IF (NU.GT.NMAX) GO TO 2[illegible]                         IU 159
            WRITE (JWRITE,31) NU                                       IU 160
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C        READ AND WRITE UPPER SURFACE INPUT POINTS AND WEIGHTING        IU 161
      READ (JREAD,32) (XU(I),YU(I),WU(I),I=1,NU)                        IU 162
      DO 2 I=1,NU                                                       IU 163
      IF (WU(I).LT.1.0) WU(I)=1.0                                       IU 164
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
      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
      IF (WL(I).LT.1.0) WL(I)=1.0                                       IU 180
3     CONTINUE                                                          IU 181
      IF (IOP.EQ.0) WRITE (JWRITE,40) (XL(I),I=1,NL)                    IU 182
      IF (IOP.NE.0) WRITE (JWRITE,41) (XL(I),I=1,NL)                    IU 183
      IF (IOP.LT.2) WRITE (JWRITE,42) (YL(I),I=1,NL)                    IU 184
      IF (IOP.EQ.2) WRITE (JWRITE,43) (YL(I),I=1,NL)                    IU 185
      IF (IOP.EQ.3) WRITE (JWRITE,44) (YL(I),I=1,NL)                    IU 186
      WRITE (JWRITE,45) (WL(I),I=1,NL)                                  IU 187
C        READ AND WRITE TRAILING-EDGE COORDINATES                       IU 188
      IF (IOP.LE.1) GO TO 4                                             IU 189
      READ (JREAD,29) YLTE,YNOSE,YUTE                                   IU 190
      WRITE (JWRITE,46) YLTE,YNOSE,YUTE                                 IU 191
C        READ AND WRITE NUMBER OF INTERPOLATION COORDINATES             IU 192
4     IF (INTR.EQ.0) GO TO 6                                            IU 193
      IF (INTR.NE.2) GO TO 5                                            IU 194
      READ (JREAD,29) VAR(1)                                            IU 195
      NINT=IFIX(VAR(1))                                                 IU 196
      IF (NINT.GT.NMAX) GO TO 24                                        IU 197
      WRITE (JWRITE,47) NINT                                            IU 198
C        READ AND WRITE INTERPOLATION COORDINATES                       IU 199
      READ (JREAD,29) (XINT(I),I=1,NINT)                                IU 200
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      WRITE (JWRITE,48) (XINT(I),I=1,NINT)                              IU 201
C        READ AND WRITE NEW CHORD OF INTERPOLATED COORDINATES           IU 202
5     READ (JREAD,29) CNEW                                              IU 203
      WRITE (JWRITE,49) CNEW                                            IU 204
C                                                                       IU 205
C     CHECK UPPER SURFACE COORDINATES FOR BAD POINTS                    IU 206
C                                                                       IU 207
6     IF (IOP.NE.0) GO TO 7                                             IU 208
      IF (IBAD.EQ.1) CALL BADPT (XU,YU,NU,TOLR,1,IERR)                  IU 209
      IF (IERR.NE.0) GO TO 26                                           IU 210
C                                                                       IU 211
C     CHECK LOWER SURFACE COORDINATES FOR BAD POINTS                    IU 212
C                                                                       IU 213
      IF (IBAD.EQ.1) CALL BADPT (XL,YL,NL,TOLR,2,IERR)                  IU 214
      IF (IERR.NE.0) GO TO 26                                           IU 215
C                                                                       IU 216
C     TRANSLATE AND ROTATE THE INPUT COORDINATES SO THAT THE X-AXIS     IU 217
C     CORRESPONDS TO THE LONGEST CHORDLINE OF THE AIRFOIL               IU 218
C                                                                       IU 219
      IF (ITRN.EQ.1) CALL TRNSRT (XU,YU,WU,NU,XL,YL,WL,NL,TITLE)        IU 220
C                                                                       IU 221
C     LOAD X, Y, THETA, YPS, AND YPPS ARRAYS                            IU 222
C                                                                       IU 223
7     IF (IOP) 8,8,15                                                   IU 224
C        IF IOP=0, COMPUTE THETA FROM INPUT X                           IU 225
C              COMPUTE THETA FOR LOWER SURFACE                          IU 226
8     CHORD=XL(NL)-XL(1)                                                IU 227
      DELTA=XU(NU)-XU(1)                                                IU 228
      IF (DELTA.GT.CHORD) CHORD=DELTA                                   IU 229
      NP=0                                                              IU 230
      DO 11 I=1,NL                                                      IU 231
      NP=NP+1                                                           IU 232
      J=NL+1-I                                                          IU 233
      W(NP)=WL(J)                                                       IU 234
      DELTA=(XL(J)-XL(1))/CHORD                                         IU 235
      IF (DELTA.LE.CONS) GO TO 9                                        IU 236
      DELTA=TAN(DELTA/CONS-1.)                                          IU 237
      THETA(NP)=-PI/2-ALOG(DELTA+SQRT(DELTA*DELTA+1.))                  IU 238
      GO TO 10                                                          IU 239
9     THETA(NP)=-ACOS(1.-DELTA/CONS)                                    IU 240
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
   10    X(NP)=XL(J)/CHORD                                          IU 241
   11    Y(NP)=YL(J)/CHORD                                          IU 242
         NOSE=NP                                                    IU 243
   C           COMPUTE THETA FOR UPPER SURFACE                      IU 244
         J=1                                                        IU 245
         IF (XL(1).EQ.XU(1).AND.YL(1).EQ.YU(1)) J=2                 IU 246
         DO 14 I=J,NU                                               IU 247
         NP=NP+1                                                    IU 248
         W(NP)=WU(I)                                                IU 249
         DELTA=(XU(I)-XU(1))/CHORD                                  IU 250
         IF (DELTA.LE.CONS) GO TO 12                                IU 251
         DELTA=TAN(DELTA/CONS-1.)                                   IU 252
         THETA(NP)=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))             IU 253
         GO TO 13                                                   IU 254
   12    THETA(NP)=ACOS(1.-DELTA/CONS)                              IU 255
   13    X(NP)=XU(I)/CHORD                                          IU 256
   14    Y(NP)=YU(I)/CHORD                                          IU 257
         GO TO 20                                                   IU 258
   C        IF IOP=1, 2, OR 3, COMPUTE X/C FROM INPUT THETA         IU 259
   C           COMPUTE X/C FOR LOWER SURFACE                        IU 260
   15    CHORD=1.0                                                  IU 261
         NP=0                                                       IU 262
         DO 17 I=1,NL                                               IU 263
         NP=NP+1                                                    IU 264
         J=NL+1-I                                                   IU 265
         W(NP)=WL(J)                                                IU 266
         IF (IOP.EQ.1) Y(NP)=YL(J)                                  IU 267
         IF (IOP.EQ.2) YPS(NP)=YL(J)                                IU 268
         IF (IOP.EQ.3) YPPS(NP)=YL(J)                               IU 269
         THETA(NP)=XL(J)/RAD                                        IU 270
         DELTA=ABS(THETA(NP))                                       IU 271
         IF (DELTA.GT.PI2) GO TO 16                                 IU 272
         XL(J)=CONS*(1.-COS(DELTA))                                 IU 273
         GO TO 17                                                   IU 274
   16    XL(J)=CONS*(ATAN(SINH(DELTA-PI2))+1.)                      IU 275
   17    X(NP)=XL(J)                                                IU 276
         NOSE=NP                                                    IU 277
   C           COMPUTE X/C FOR UPPER SURFACE                        IU 278
         XU(1)=XL(1)                                                IU 279
         DO 19 I=2,NU                                               IU 280
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      NP=NP+1                                                          IU 281
      W(NP)=WU(I)                                                      IU 282
      IF (IOP.EQ.1) Y(NP)=YU(I)                                        IU 283
      IF (IOP.EQ.2) YPS(NP)=YU(I)                                      IU 284
      IF (IOP.EQ.3) YPPS(NP)=YU(I)                                     IU 285
      THETA(NP)=XU(I)/RAD                                              IU 286
      DELTA=ABS(THETA(NP))                                             IU 287
      IF (DELTA.GT.PI2) GO TO 18                                       IU 288
      XU(I)=CONS*(1.-COS(DELTA))                                       IU 289
      GO TO 19                                                         IU 290
18    XU(I)=CONS*(ATAN(SINH(DELTA-PI2))+1.)                            IU 291
19    X(NP)=XU(I)                                                      IU 292
C                                                                      IU 293
C     PRINT SUMMARY OF INPUT DATA                                      IU 294
C                                                                      IU 295
20    WRITE (JWRITE,50) TITLE                                          IU 296
      DO 21 I=1,NP                                                     IU 297
      DELTA=THETA(I)*RAD                                               IU 298
      IF (IOP.LE.1) WRITE (JWRITE,51) I,X(I),Y(I),DELTA,W(I)           IU 299
      IF (IOP.EQ.2) WRITE (JWRITE,52) I,X(I),DELTA,YPS(I),W(I)         IU 300
      IF (IOP.EQ.3) WRITE (JWRITE,53) I,X(I),DELTA,YPPS(I),W(I)        IU 301
21    CONTINUE                                                         IU 302
      WRITE (JWRITE,54) CHORD                                          IU 303
      GO TO 26                                                         IU 304
C                                                                      IU 305
C     PRINT ERROR MESSAGES                                             IU 306
C                                                                      IU 307
22    NN=IFIX(VAR(1))                                                  IU 308
      WRITE (JWRITE,55) NN                                             IU 309
      GO TO 25                                                         IU 310
23    WRITE (JWRITE,56) IOP                                            IU 311
      GO TO 25                                                         IU 312
24    WRITE (JWRITE,57) NINT                                           IU 313
C                                                                      IU 314
C     NO ADDITIONAL INPUT DATA                                         IU 315
C                                                                      IU 316
25    IERR=2                                                           IU 317
C                                                                      IU 318
C     RETURN TO CALLING PROGRAM                                        IU 319
C                                                                      IU 320
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

 321          26    RETURN                                                    IU 321
              C                                                               IU 322
              27    FORMAT (8A10)                                             IU 323
              28    FORMAT (1H1,57X,14H--INPUT DATA--//5X,7HTITLE--,2X,8A10)  IU 324
 325          29    FORMAT (8F10.5)                                           IU 325
              30    FORMAT (/5X,6HITER =,I4,3X,7HIPLOT =,I3,3X,8HIPUNCH =,I3,3X,5HIOP  IU 326
                   1=,I3,3X,8HICAMTK =,I3,3X,6HIBAD =,I3,3X,6HITRN =,I3,3X,6HINTR =,I3 IU 327
                   2)                                                         IU 328
              31    FORMAT (/5X,5HNNU =,I4)                                   IU 329
 330          32    FORMAT (3F10.5)                                           IU 330
              33    FORMAT (/5X,3HXU=,8E15.6/(8X,8E15.6))                     IU 331
              34    FORMAT (/5X,3HTU=,8E15.6/(8X,8E15.6))                     IU 332
              35    FORMAT (/5X,3HYU=,8E15.6/(8X,8E15.6))                     IU 333
              36    FORMAT (/4X,4HYPU=,8E15.6/(8X,8E15.6))                    IU 334
 335          37    FORMAT (/3X,5HYPPU=,8E15.6/(8X,8E15.6))                   IU 335
              38    FORMAT (/5X,3HWU=,8E15.6/(8X,8E15.6))                     IU 336
              39    FORMAT (/5X,4HNL =,I4)                                    IU 337
              40    FORMAT (/5X,3HXL=,8E15.6/(8X,8E15.6))                     IU 338
              41    FORMAT (/5X,3HTL=,8E15.6/(8X,8E15.6))                     IU 339
 340          42    FORMAT (/5X,3HYL=,8E15.6/(8X,8E15.6))                     IU 340
              43    FORMAT (/4X,4HYPL=,8E15.6/(8X,8E15.6))                    IU 341
              44    FORMAT (/3X,5HYPPL=,8E15.6/(8X,8E15.6))                   IU 342
              45    FORMAT (/5X,3HWL=,8E15.6/(8X,8E15.6))                     IU 343
              46    FORMAT (/3X,6HYLTE =,E15.6,5X,7HYNOSE =,E15.6,5X,6HYUTE =,E15.6)  IU 344
 345          47    FORMAT (/3X,6HNINT =,I4)                                  IU 345
              48    FORMAT (/3X,5HXINT=,8E15.6/(8X,8E15.6))                   IU 346
              49    FORMAT (/3X,6HCNEW =,F10.3)                               IU 347
              50    FORMAT (1H1,29X,25H--SUMMARY OF INPUT DATA--//5X,9HTITLE--  ,8A10/ IU 348
                   1/9X,1HI,10X,3HX/C,12X,3HY/C,12X,5HTHETA,10X,3HYPS,12X,4HYPPS,14X,1 IU 349
 350               2HW)                                                      IU 350
              51    FORMAT (I10,2F15.6,F15.2,30X,F15.2)                       IU 351
              52    FORMAT (I10,F15.6,15X,F15.2,F15.6,15X,F15.2)              IU 352
              53    FORMAT (I10,F15.6,15X,F15.2,15X,F15.6,F15.2)              IU 353
              54    FORMAT (/5X,7HCHORD =,F15.6)                              IU 354
 355          55    FORMAT (//5X,28HINPUT CARD ERROR  NU OR NL =,I4)          IU 355
              56    FORMAT (//5X,23HINPUT CARD ERROR  IOP =,I4)               IU 356
              57    FORMAT (//5X,24HINPUT CARD ERROR  NINT =,I5)              IU 357
                    END                                                       IU 358
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DO 3 I=J,NU                                                       TR  41
      N=N+1                                                             TR  42
      W(N)=WU(I)                                                        TR  43
      X(N)=XU(I)                                                        TR  44
    3 Y(N)=YU(I)                                                        TR  45
C        COMPUTE MIDPOINT OF TRAILING-EDGE BASE                         TR  46
      XTE=0.5*(X(1)+X(N))                                               TR  47
      YTE=0.5*(Y(1)+Y(N))                                               TR  48
C        FIND MOST FORWARD LEADING-EDGE POINT AND LONGEST CHORD         TR  49
      CHORD=0.0                                                         TR  50
      DO 5 I=1,N                                                        TR  51
      DIST=SQRT((X(I)-XTE)**2+(Y(I)-YTE)**2)                            TR  52
      IF (DIST-CHORD) 5,5,4                                             TR  53
    4 CHORD=DIST                                                        TR  54
      NOSE=I                                                            TR  55
      XNOSE=X(I)                                                        TR  56
      YNOSE=Y(I)                                                        TR  57
    5 CONTINUE                                                          TR  58
C                                                                       TR  59
C     TRANSLATE AND ROTATE AIRFOIL                                      TR  60
C                                                                       TR  61
      IF (CHORD.LE.0.0) GO TO 6                                         TR  62
      COSA=(XTE-XNOSE)/CHORD                                            TR  63
      SINA=(YTE-YNOSE)/CHORD                                            TR  64
      ANGLE=ATAN(SINA/COSA)*RAD                                         TR  65
      GO TO 7                                                           TR  66
    6 COSA=0.0                                                          TR  67
      SINA=0.0                                                          TR  68
      ANGLE=0.0                                                         TR  69
    7 DO 8 I=1,N                                                        TR  70
      DIST=X(I)                                                         TR  71
      X(I)=(DIST-XNOSE)*COSA+(Y(I)-YNOSE)*SINA                          TR  72
    8 Y(I)=(Y(I)-YNOSE)*COSA-(DIST-XNOSE)*SINA                          TR  73
C                                                                       TR  74
C     REDEFINE LOWER AND UPPER SURFACE COORDINATES                      TR  75
C                                                                       TR  76
      DO 9 I=1,NOSE                                                     TR  77
      J=NOSE+1-I                                                        TR  78
      WL(I)=W(J)                                                        TR  79
      XL(I)=X(J)                                                        TR  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      9     YL(I)=Y(J)                                                    TR  81
            NL=NOSE                                                       TR  82
            DO 10 I=NOSE,N                                                TR  83
            J=I+1-NOSE                                                    TR  84
            WU(J)=W(I)                                                    TR  85
            XU(J)=X(I)                                                    TR  86
      10    YU(J)=Y(I)                                                    TR  87
            NU=J                                                          TR  88
      C                                                                   TR  89
      C     PRINT NEW AIRFOIL COORDINATES                                 TR  90
      C                                                                   TR  91
            WRITE (JWRITE,16) TITLE                                       TR  92
            J=NU                                                          TR  93
            IF (NL.GT.NU) J=NL                                            TR  94
            DO 11 I=1,J                                                   TR  95
            IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,14) I,XU(I),YU(I),XL(I),YL( TR  96
           1I)                                                            TR  97
            IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,14) I,XU(I),YU(I)      TR  98
            IF (I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,15) I,XL(I),YL(I)      TR  99
      11    CONTINUE                                                      TR 100
            WRITE (JWRITE,12) XNOSE,YNOSE,ANGLE                           TR 101
            RETURN                                                        TR 102
      C                                                                   TR 103
      12    FORMAT (/5X,7HXNOSE =,F15.6,5X,7HYNOSE =,F15.6,5X,7HANGLE =,F8.3) TR 104
      13    FORMAT (1H1,32X,21H--INPUT COORDINATES--//5X,7HTITLE--,2X,8A10//9X TR 105
           1,1HI,11X,2HXU,13X,2HYU,13X,2HXL,13X,2HYL)                     TR 106
      14    FORMAT (5X,I5,4F15.6)                                         TR 107
      15    FORMAT (5X,I5,30X,2F15.6)                                     TR 108
      16    FORMAT (1H1,21X,38H--TRANSLATED AND ROTATED COORDINATES--//5X,7HTI TR 109
           1TLE--,2X,8A10//9X,1HI,11X,2HXU,13X,2HYU,13X,2HXL,13X,2HYL)    TR 110
            END                                                           TR 111-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE BADPT (X,Y,NP,TOLR,ISURF,IERR)                         BD   1
C                                                                       BD   2
C     ROUTINE TO EDIT BAD POINTS FROM X AND Y INPUT COORDINATES         BD   3
C                                                                       BD   4
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB        1982        BD   5
C                                                                       BD   6
      DIMENSION X(1), Y(1), SURF(2)                                     BD   7
C                                                                       BD   8
      COMMON /HLM/ TI(100),YI(100),YN(100),THETA(100)                   BD   9
C                                                                       BD  10
      COMMON /BLK1/ PI,PI2,RAD,CONS                                     BD  11
C                                                                       BD  12
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                BD  13
C                                                                       BD  14
      DATA SURF(1)/5HUPPER/,SURF(2)/5HLOWER/                            BD  15
C                                                                       BD  16
C     IF TOLERANCE IS ZERO OR NEGATIVE RETURN                           BD  17
C                                                                       BD  18
      IERR=0                                                            BD  19
      IF (TOLR.LE.0.0) RETURN                                           BD  20
C                                                                       BD  21
C     COMPUTE LOCAL CHORD                                               BD  22
C                                                                       BD  23
      CHORD=X(NP)-X(1)                                                  BD  24
C                                                                       BD  25
C     INITIALIZE ITERATION PARAMETERS                                   BD  26
C                                                                       BD  27
      ICD=0                                                             BD  28
      IPTP=0                                                            BD  29
      N1=NP-1                                                           BD  30
      NMAX=0                                                            BD  31
      TOLC=TOLR*CHORD                                                   BD  32
C                                                                       BD  33
C     COMPUTE THETA EQUIVALENT OF X                                     BD  34
C                                                                       BD  35
      DO 2 I=1,NP                                                       BD  36
      DELTA=(X(I)-X(1))/CHORD                                           BD  37
      IF (DELTA.LE.CONS) GO TO 1                                        BD  38
      DELTA=TAN(DELTA/CONS-1.)                                          BD  39
      THETA(I)=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))                     BD  40
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      GO TO 2                                                           BD  41
    1 THETA(I)=ACOS(1.-DELTA/CONS)                                      BD  42
    2 CONTINUE                                                          BD  43
    C                                                                   BD  44
    C LOOP TO SEARCH FOR BAD POINTS                                     BD  45
    C                                                                   BD  46
    3 NMAX=NMAX+1                                                       BD  47
      JSTART=1                                                          BD  48
    C    COMPUTE NEW Y VALUE BY INTERPOLATION                           BD  49
      DO 5 I=2,N1                                                       BD  50
      K=0                                                               BD  51
    C          LOAD TI AND YI ARRAY - OMIT THE I(TH) INPUT DATA POINT   BD  52
      DO 4 J=1,NP                                                       BD  53
      IF (I.EQ.J) GO TO 4                                               BD  54
      K=K+1                                                             BD  55
      TI(K)=THETA(J)                                                    BD  56
      YI(K)=Y(J)                                                        BD  57
    4 CONTINUE                                                          BD  58
    C          INTERPOLATE I(TH) DATA POINT                             BD  59
      CALL INTER (THETA(I),YN(I),K,TI,YI,JSTART,JEND,ICD)               BD  60
      JSTART=JEND                                                       BD  61
    5 CONTINUE                                                          BD  62
    C    CHECK TOLERANCE OF INTERPOLATED POINTS                         BD  63
      IPT=0                                                             BD  64
      ERRMAX=0.                                                         BD  65
      DO 7 I=2,N1                                                       BD  66
      ERRMIN=0.                                                         BD  67
      ERR=ABS(YN(I)-Y(I))                                               BD  68
      IF (ERR.GE.TOLC) ERRMIN=ERR                                       BD  69
      IF (ERRMIN-ERRMAX) 7,7,6                                          BD  70
    6 IPT=I                                                             BD  71
      ERRMAX=ERRMIN                                                     BD  72
    7 CONTINUE                                                          BD  73
      IF (IPT.EQ.0) RETURN                                              BD  74
    C    PRINT COORDINATES OF BAD POINTS                                BD  75
      IF (NMAX.EQ.1) WRITE (JWRITE,9) SURF(ISURF),TOLC                  BD  76
      WRITE (JWRITE,10) IPT,X(IPT),Y(IPT),YN(IPT)                       BD  77
    C    REPLACE BAD POINT WITH INTERPOLATED VALUE                      BD  78
      Y(IPT)=YN(IPT)                                                    BD  79
    C    CHECK TO SEE IF THIS BAD POINT IS ADJACENT TO THE PREVIOUS BAD BD  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C        POINT -- IF IT IS, PRINT A WARNING MESSAGE AND TERMINATE        BD  81
C        PROGRAM EXECUTION                                               BD  82
      IF ((IPTP.EQ.IPT-1).OR.(IPTP.EQ.IPT+1)) GO TO 8                    BD  83
      IF (IPTP.EQ.IPT) GO TO 8                                           BD  84
      IPTP=IPT                                                           BD  85
      IF (NMAX.GE.NP) RETURN                                             BD  86
C                                                                        BD  87
C     RETURN TO START OF LOOP AND SEARCH FOR NEXT BAD POINT              BD  88
C                                                                        BD  89
      GO TO 3                                                            BD  90
C                                                                        BD  91
C     WARNING MESSAGE PRINT STATEMENT                                    BD  92
C                                                                        BD  93
8     WRITE (JWRITE,11)                                                  BD  94
      IERR=1                                                             BD  95
      RETURN                                                             BD  96
C                                                                        BD  97
9     FORMAT (1H1//1X,44HWARNING -- BAD POINTS HAVE BEEN FOUND ON THE,1X BD  98
     1,A5,1X,37HSURFACE BASED ON AN EDIT TOLERANCE OF,F10.6/)            BD  99
10    FORMAT (1X,15HBAD POINT AT I=,I4,5X,4HX = ,F10.6,5X,4HY = ,F10.6,5 BD 100
     1X,18HREPLACED WITH Y = ,F10.6/)                                    BD 101
11    FORMAT (1X,93HADJACENT BAD POINTS HAVE BEEN FOUND -- PLEASE CORREC BD 102
     1T YOUR INPUT DATA AND RESUBMIT THIS CASE.)                         BD 103
      END                                                                BD 104-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE SMOXY (THETA,X,Y,W,YSMO,YPS,YPPS,NP,NOSE,YLTF,YNOSE,YUT SO   1
     1E,EPS,DF,ITER,TITLE,IOP,IERR)                                      SO   2
C                                                                        SO   3
C     THIS SUBROUTINE PRESENTS A TECHNIQUE FOR SMOOTHING Y INPUT         SO   4
C     COORDINATES USING LEAST SQUARES POLYNOMIAL AND CUBIC SPLINE        SO   5
C     METHODS                                                            SO   6
C                                                                        SO   7
C     IF IOP=0 OR 1, COMPUTE YPPU (UNSMOOTHED SECOND DERIVATIVES) FROM   SO   8
C     LEAST SQUARES POLYNOMIAL FITTING OF Y VS THETA. THEN COMPUTE       SO   9
C     YPPS (SMOOTHED SECOND DERIVATIVES) FROM LEAST SQUARES CUBIC        SO  10
C     SPLINE FITTING OF YPPU VS THETA. FINALLY COMPUTE YSMO (SMOOTHED Y  SO  11
C     COORDINATES) USING INVERSE CUBIC SPLINE METHOD.                    SO  12
C                                                                        SO  13
C     IF IOP=2, COMPUTE SECOND DERIVATIVES FROM INPUT FIRST DERIVATIVES. SO  14
C     THEN COMPUTE UNSMOOTHED Y COORDINATES FROM SECOND DERIVATIVES AND  SO  15
C     FOLLOW SAME PROCEDURES AS OUTLINED ABOVE FOR IOP 0 OR 1.           SO  16
C                                                                        SO  17
C     IF IOP=3, COMPUTE UNSMOOTHED Y COORDINATES FROM INPUT SECOND       SO  18
C     DERIVATIVES. THEN FOLLOW SAME PROCEDURES AS OUTLINED ABOVE FOR     SO  19
C     IOP 0 OR 1.                                                        SO  20
C                                                                        SO  21
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          SO  22
C                                                                        SO  23
C        DIMENSION THETA, X, Y, W, YSMO, YPS, AND YPPS BY NP IN CALLING SO  24
C        PROGRAM                                                         SO  25
      DIMENSION TITLE(8), THETA(1), X(1), Y(1), W(1), YSMO(1), YPS(1), Y SO  26
     1PPS(1)                                                             SO  27
C                                                                        SO  28
      COMMON /HLM/ WK(200,10)                                            SO  29
C                                                                        SO  30
      COMMON /SMY/ YPP(200),YUSMO(200),DUM(200),A(200,4),YN(200),YPPU(20 SO  31
     10),SUMY(300),LTFR(30)                                              SO  32
C                                                                        SO  33
      COMMON /BLK1/ PI,PT2,RAD,CONS                                      SO  34
C                                                                        SO  35
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                 SO  36
C                                                                        SO  37
      DATA LMX/200/,WT/100./                                             SO  38
C                                                                        SO  39
C     SINH(X)=(EXP(X)-EXP(-X))/2.                                        SO  40
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C     COSH(X)=(EXP(X)+EXP(-X))/2.                                       SO  41
C                                                                       SO  42
      IERR=0                                                            SO  43
      IF (IOP.EQ.0.OR.IOP.EQ.1) GO TO 13                                SO  44
      IF (IOP.EQ.2) GO TO 1                                             SO  45
      IF (IOP.EQ.3) GO TO 11                                            SO  46
C                                                                       SO  47
C        IF IOP=2, COMPUTE SECOND DERIVATIVES FROM INPUT FIRST          SO  48
C        DERIVATIVES. THEN COMPUTE INITIAL Y/C COORDINATES FROM SECOND  SO  49
C        DERIVATIVES.                                                   SO  50
C                                                                       SO  51
C              COMPUTE SECOND DERIVATIVES USING CSDS                    SO  52
1     DO 2 I=1,NP                                                       SO  53
2     DUM(I)=1.0                                                        SO  54
      T1=0.0                                                            SO  55
      CALL CSDS (LMX,NP,THETA,YPS,DUM,T1,-1,A,WK,IERR)                  SO  56
      IF (IERR.NE.0) GO TO 71                                           SO  57
      DO 4 I=1,NP                                                       SO  58
      IF (I.EQ.NP) GO TO 3                                              SO  59
      YPPS(I)=A(I,2)                                                    SO  60
      GO TO 4                                                           SO  61
3     DELTA=THETA(I)-THETA(I-1)                                         SO  62
      YPPS(I)=(3.*A(I-1,4)*DELTA+2.*A(I-1,3))*DELTA+A(I-1,2)            SO  63
4     CONTINUE                                                          SO  64
C              COMPUTE SECOND DERIVATIVES USING LSQSMO                  SO  65
      DELTA=1.                                                          SO  66
      CALL LSQSMO (THETA,YPS,W,DUM,YPP,YUSMO,NP,1,NP,NOSE,DELTA,EPS,IERR SO  67
     1)                                                                 SO  68
      IF (IERR.NE.0) RETURN                                             SO  69
C              COMPUTE Y/C COORDINATES                                  SO  70
      CALL YNEW (THETA,YPPS,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRITE, SO  71
     10)                                                                SO  72
      CALL YNEW (THETA,YPP,YUSMO,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRI SO  73
     1TE,0)                                                             SO  74
C              COMPUTE NEW FIRST DERIVATIVES AND COMPARE WITH INPUT     SO  75
C              FIRST DERIVATIVES                                        SO  76
      WRITE (JWRITE,73) TITLE                                           SO  77
      SUM1=0.0                                                          SO  78
      SUM2=0.0                                                          SO  79
      DO 7 I=1,NP                                                       SO  80
```

CARD NO.

```
      IF (I.EQ.1) GO TO 5                                               SO  81
      DELTA=THETA(I)-THETA(I-1)                                         SO  82
      YN(I)=YPPS(I-1)*DELTA/6.+YPPS(I)*DELTA/3.+(Y(I)-Y(I-1))/DELTA     SO  83
      DUM(I)=YPP(I-1)*DELTA/6.+YPP(I)*DELTA/3.+(YUSMO(I)-YUSMO(I-1))/DEL SO  84
     1TA                                                                SO  85
      GO TO 6                                                           SO  86
    5 DELTA=THETA(2)-THETA(1)                                           SO  87
      YN(1)=-YPPS(1)*DELTA/3.-YPPS(2)*DELTA/6.+(Y(2)-Y(1))/DELTA        SO  88
      DUM(1)=-YPP(1)*DELTA/3.-YPP(2)*DELTA/6.+(YUSMO(2)-YUSMO(1))/DELTA SO  89
    6 T1=YPS(I)-YN(I)                                                   SO  90
      T2=YPS(I)-DUM(I)                                                  SO  91
      SUM1=SUM1+T1*T1                                                   SO  92
      SUM2=SUM2+T2*T2                                                   SO  93
    7 WRITE (JWRITE,74) I,YPS(I),YN(I),T1,DUM(I),T2                     SO  94
      WRITE (JWRITE,75) SUM1,SUM2                                       SO  95
    C           SELECT OUTPUT FROM EITHER CSDS OR LSQSMO                SO  96
      DO 10 I=1,NP                                                      SO  97
      IF (SUM2.LT.SUM1) GO TO 8                                         SO  98
      YPP(I)=YPPS(I)                                                    SO  99
      GO TO 9                                                           SO 100
    8 Y(I)=YUSMO(I)                                                     SO 101
      YN(I)=DUM(I)                                                      SO 102
    9 YSMO(I)=Y(I)                                                      SO 103
   10 YUSMO(I)=Y(I)                                                     SO 104
      IF (SUM2.GE.SUM1) WRITE (JWRITE,76)                               SO 105
      IF (SUM2.LT.SUM1) WRITE (JWRITE,77)                               SO 106
      IF (ITER.EQ.0) GO TO 48                                           SO 107
      GO TO 13                                                          SO 108
    C                                                                   SO 109
    C        IF IOP=3, COMPUTE INITIAL Y/C FROM INPUT SECOND DERIVATIVES SO 110
    C        AND Y/C COORDINATES AT THE UPPER AND LOWER SURFACE TRAILING SO 111
    C        EDGE AND NOSE                                              SO 112
    C                                                                   SO 113
   11 CALL YNEW (THETA,YPPS,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JWRITE, SO 114
     10)                                                                SO 115
    C           COMPUTE FIRST DERIVATIVES                               SO 116
      DO 12 I=1,NP                                                      SO 117
      YSMO(I)=Y(I)                                                      SO 118
      YUSMO(I)=Y(I)                                                     SO 119
      IF (I.EQ.1) GO TO 12                                              SO 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
                DELTA=THETA(I)-THETA(I-1)                                       SO 121
                YN(I)=YPPS(I-1)*DELTA/6.+YPPS(I)*DELTA/3.+(Y(I)-Y(I-1))/DELTA   SO 122
          12    YPP(I)=YPPS(I)                                                  SO 123
                DELTA=THETA(2)-THETA(1)                                         SO 124
                YN(1)=-YPPS(1)*DELTA/3.-YPPS(2)*DELTA/6.+(Y(2)-Y(1))/DELTA      SO 125
                IF (ITER.EQ.0) GO TO 48                                         SO 126
          C                                                                     SO 127
          C        INITIALIZE ARRAYS                                            SO 128
          C                                                                     SO 129
          13    DO 14 I=1,NP                                                    SO 130
                YUSMO(I)=Y(I)                                                   SO 131
                IF (IOP.LT.2) YPP(I)=0.0                                        SO 132
                YSMO(I)=THETA(I)*RAD                                            SO 133
          14    DUM(I)=1.                                                       SO 134
                IF (ITER.GT.0) GO TO 17                                         SO 135
          C                                                                     SO 136
          C        IF IOP=0 OR 1 AND NO SMOOTHING DESIRED (I.E. ITER=0) , COMPUTE SO 137
          C        SECOND DERIVATIVE FROM CUBIC SPLINE SUBROUTINE               SO 138
          C                                                                     SO 139
                CALL CSDS (LMX,NP,THETA,Y,DUM,0.0,-1,A,WK,IERR)                 SO 140
                IF (IERR.NE.0) GO TO 71                                         SO 141
          C        COMPUTE Y AND SECOND DERIVATIVE                              SO 142
                DO 16 I=1,NP                                                    SO 143
                IF (I.EQ.NP) GO TO 15                                           SO 144
                YSMO(I)=A(I,1)                                                  SO 145
                YN(I)=A(I,2)                                                    SO 146
                YPP(I)=2.*A(I,3)                                                SO 147
                GO TO 16                                                        SO 148
          15    DELTA=THETA(I)-THETA(I-1)                                       SO 149
                YSMO(I)=((A(I-1,4)*DELTA+A(I-1,3))*DELTA+A(I-1,2))*DELTA+A(I-1,1) SO 150
                YN(I)=(3.*A(I-1,4)*DELTA+2.*A(I-1,3))*DELTA+A(I-1,2)            SO 151
                YPP(I)=6.*A(I-1,4)*DELTA+2.*A(I-1,3)                            SO 152
          16    CONTINUE                                                        SO 153
                GO TO 48                                                        SO 154
          C                                                                     SO 155
          C        FIND MAXIMUM INPUT Y VALUE AND ITS LOCATION FOR UPPER AND    SO 156
          C        LOWER SURFACES                                               SO 157
          C             LOWER SURFACE                                           SO 158
          17    YMAX=0.0                                                        SO 159
                JMAXL=1                                                         SO 160
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.
                DO 19 I=1,NOSE                                          SO 161
                J=NOSE+1-I                                              SO 162
                IF (ABS(Y(J)).GT.YMAX) GO TO 18                         SO 163
                GO TO 19                                                SO 164
          18    YMAX=ABS(Y(J))                                          SO 165
                JMAXL=J                                                 SO 166
          19    CONTINUE                                                SO 167
          C             UPPER SURFACE                                   SO 168
                YMAX=0.0                                                SO 169
                JMAXU=1                                                 SO 170
                DO 21 I=NOSE,NP                                         SO 171
                IF (ABS(Y(I)).GT.YMAX) GO TO 20                         SO 172
                GO TO 21                                                SO 173
          20    YMAX=ABS(Y(I))                                          SO 174
                JMAXU=I                                                 SO 175
          21    CONTINUE                                                SO 176
          C                                                             SO 177
          C        COMPUTE UNSMOOTHED SECOND DERIVATIVE USING LEAST     SO 178
          C        SQUARES POLYNOMIAL METHOD                            SO 179
          C                                                             SO 180
                J1=0                                                    SO 181
                ICON=0                                                  SO 182
                MTER=0                                                  SO 183
                J=ITER                                                  SO 184
                KTI=0                                                   SO 185
                IF (IPRINT.NE.0) WRITE (JWRITE,78) TITLE                SO 186
                DO 23 I=1,30                                            SO 187
                KTI=KTI+1                                               SO 188
                LTER(I)=10                                              SO 189
                J=J-10                                                  SO 190
                IF (J) 22,24,23                                         SO 191
          22    LTER(I)=10+J                                            SO 192
                GO TO 24                                                SO 193
          23    CONTINUE                                                SO 194
          24    DO 39 LL=1,KTI                                          SO 195
                N1=LTER(LL)                                             SO 196
                DO 34 I=1,N1                                            SO 197
          C             CALL LEAST SQUARES POLYNOMIAL SMOOTHING ROUTINE SO 198
                CALL LSQSMO (THETA,YUSMO,W,YN,DUM,YPPU,NP,JMAXL,JMAXU,NOSE,WT,EPS, SO 199
               1IERR)                                                   SO 200
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.
                IF (IERR.NE.0) RETURN                                   SO 201
          C             COMPUTE ERROR TERM                              SO 202
                SUMY(I)=0.0                                             SO 203
                DO 25 J=1,NP                                            SO 204
          25    SUMY(I)=SUMY(I)+(YPPU(J)-YPP(J))**2                     SO 205
                J1=J1+1                                                 SO 206
                IF ((I.LE.3).AND.(LL.EQ.1)) GO TO 26                    SO 207
                IF (I.EQ.1) GO TO 26                                    SO 208
          C             CHECK FOR OSCILLATIONS IN CONVERGENCE OF ERROR TERM  SO 209
                IF (SUMY(I)-SUMY(I-1)) 26,26,32                         SO 210
          C             LOAD ARRAYS FOR NEXT ITERATION                  SO 211
          26    DO 31 J=1,NP                                            SO 212
                WK(J,I)=YPPU(J)                                         SO 213
                IF (LL.EQ.1.AND.I.EQ.1) YPPS(J)=YPPU(J)                 SO 214
                YPP(J)=YPPU(J)                                          SO 215
                CC=YUSMO(J)                                             SO 216
                IF (J1-2) 29,28,27                                      SO 217
          27    AA=YN(J)-YUSMO(J)                                       SO 218
                BB=A(J,1)-A(J,2)                                        SO 219
                T1=SIGN(1.,AA)                                          SO 220
                T2=SIGN(1.,BB)                                          SO 221
                IF (T1.EQ.T2.OR.AA.EQ.BB) GO TO 28                      SO 222
                YUSMO(J)=A(J,2)-BB*(YUSMO(J)-A(J,2))/(AA-BB)            SO 223
                GO TO 30                                                SO 224
          28    YUSMO(J)=0.5*(YUSMO(J)+YN(J))                           SO 225
                GO TO 30                                                SO 226
          29    YUSMO(J)=YN(J)                                          SO 227
          30    A(J,1)=YN(J)                                            SO 228
          31    A(J,2)=CC                                               SO 229
                GO TO 33                                                SO 230
          32    NTER=I-1                                                SO 231
                ICON=2                                                  SO 232
                GO TO 36                                                SO 233
          33    NTER=I                                                  SO 234
          C             CHECK FOR CONVERGENCE BASED ON INPUT EPS        SO 235
                IF (SUMY(I).LE.EPS) GO TO 35                            SO 236
          34    CONTINUE                                                SO 237
                GO TO 36                                                SO 238
          35    ICON=1                                                  SO 239
          C                                                             SO 240
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C        PRINT SECOND DERIVATIVES GENERATED DURING SMOOTHING PROCESS      SO 241
C                                                                         SO 242
36     IF (IPRINT.NE.0) GO TO 38                                          SO 243
       WRITE (JWRITE,80) TITLE                                            SO 244
       DO 37 J=1,NP                                                       SO 245
37     WRITE (JWRITE,81) J,YSPP(J),(WK(J,I),I=1,NTER)                     SO 246
       WRITE (JWRITE,82) (SUMY(I),I=1,NTER)                               SO 247
38     IF (IPRINT.NE.0) WRITE (JWRITE,79) LL,(SUMY(I),I=1,NTER)           SO 248
       NTER=NTER+NTER                                                     SO 249
       IF (ICON.NE.0) GO TO 40                                            SO 250
39     CONTINUE                                                           SO 251
40     IF (ICON.EQ.0) WRITE (JWRITE,83) NTER                              SO 252
       IF (ICON.EQ.1) WRITE (JWRITE,84) NTER                              SO 253
       IF (ICON.EQ.2) WRITE (JWRITE,85) NTER                              SO 254
C                                                                         SO 255
C        COMPUTE SMOOTHED SECOND DERIVATIVE USING LEAST SQUARES           SO 256
C        CUBIC SPLINE                                                     SO 257
C                                                                         SO 258
       DO 41 I=1,NP                                                       SO 259
41     DUM(I)=DF                                                          SO 260
C              CALL LEAST SQUARES CUBIC SPLINE ROUTINE                    SO 261
       CALL CSDS (LMX,NP,THETA,YPPU,DUM,FLOAT(NP),-1,A,WK,IERR)           SO 262
       IF (IERR.NE.0) GO TO 71                                            SO 263
C              CO   YC SECOND DERIVATIVE                                  SO 264
       SUM=0.0                                                            SO 265
       DO 44 I=1,..                                                       SO 266
       IF (I.EQ.NP) GO TO 42                                              SO 267
       YPP(I)=A(I,1)                                                      SO 268
       GO TO 43                                                           SO 269
42     DELTA=THETA(I)-THETA(I-1)                                          SO 270
       YPP(I)=((A(I-1,4)*DELTA+A(I-1,3))*DELTA+A(I-1,2))*DELTA+A(I-1,1)   SO 271
43     SUM=   +(YPPU(I)-YPP(I))**2                                        SO 272
44     YPPU(I)=YPPS(I)                                                    SO 273
       WRITE (JWRITE,88) SUM                                              SO 274
C                                                                         SO 275
C        COMPUTE NEW Y COORDINATES FROM SMOOTHED SECOND DERIVATIVES       SO 276
C                                                                         SO 277
       CALL YNEW (THETA,YPP,YSMO,NOSE,NP,YUSMO(1),YUSMO(NOSE),YUSMO(NP),E SO 278
      1PS,DUM,WK,JWRITE,1)                                                SO 279
C                                                                         SO 280
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C        CHECK NEW Y COORDINATES FOR SMOOTHNESS                          SO 281
C                                                                        SO 282
C              CALL LEAST SQUARES POLYNOMIAL ROUTINE                     SO 283
      DO 45 I=1,NP                                                       SO 284
45    A(I,1)=1.0                                                         SO 285
      CALL LSOSMO (THETA,YSMO,A,YN,DUM,YPPS,NP,1,NP,NOSE,WT,EPS,IERR)    SO 286
      IF (IERR.NE.0) RETURN                                              SO 287
C              COMPUTE ERROR TERMS                                       SO 288
      SUM1=0.0                                                           SO 289
      SUM2=0.0                                                           SO 290
      DO 46 I=1,NP                                                       SO 291
      A(I,1)=YSMO(I)-YN(I)                                               SO 292
      A(I,2)=YPP(I)-YPPS(I)                                              SO 293
      SUM1=SUM1+A(I,1)**2                                                SO 294
46    SUM2=SUM2+A(I,2)**2                                                SO 295
C                                                                        SO 296
C        COMPUTE FIRST DERIVATIVE FROM SMOOTHED SECOND DERIVATIVE        SO 297
C                                                                        SO 298
      N1=NP-1                                                            SO 299
      DO 47 I=1,N1                                                       SO 300
      DELTA=THETA(I+1)-THETA(I)                                          SO 301
47    YN(I)=-YPP(I)*DELTA/3.-YPP(I+1)*DELTA/6.+(YSMO(I+1)-YSMO(I))/DELTA SO 302
      DELTA=THETA(NP)-THETA(N1)                                          SO 303
      YN(NP)=YPP(N1)*DELTA/6.+YPP(NP)*DELTA/3.+(YSMO(NP)-YSMO(N1))/DELTA SO 304
C                                                                        SO 305
C        PRINT SUMMARY OF SMOOTHED AND UNSMOOTHED DATA                   SO 306
C                                                                        SO 307
48    WRITE (JWRITE,86) TITLE                                            SO 308
      DO 53 I=1,NP                                                       SO 309
      YPS(I)=YN(I)                                                       SO 310
      IF (THETA(I).LE.0.) YN(I)=-YN(I)                                   SO 311
      T1=ABS(THETA(I))                                                   SO 312
      IF (T1.GT.PI2) GO TO 49                                            SO 313
      GP=CONS*SIN(T1)                                                    SO 314
      GPP=CONS*COS(T1)                                                   SO 315
      GO TO 50                                                           SO 316
49    DIF=COSH(T1-PI2)                                                   SO 317
      DELTA=SINH(T1-PI2)                                                 SO 318
      GP=CONS/DIF                                                        SO 319
      GPP=-CONS*DELTA/(DIF*DIF)                                          SO 320
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      50    IF (I.EQ.NOSE) GO TO 51                                   SO 321
            DYDX=YN(I)/GP                                             SO 322
            DY2DX=(YPP(I)*GP-YN(I)*GPP)/(GP**3)                       SO 323
            CURV=ABS(DY2DX)/(SQRT(1.+DYDX**2)**3)                     SO 324
            GO TO 52                                                  SO 325
      51    DYDX=0.1E99                                               SO 326
            DY2DX=0.1E99                                              SO 327
            CURV=CONS/(YN(I)**2)                                      SO 328
            RLE=1./CURV                                               SO 329
      52    DELTA=THETA(I)*RAD                                        SO 330
            DIF=Y(I)-YSMO(I)                                          SO 331
            YPPS(I)=YPP(I)                                            SO 332
      53    WRITE (JWRITE,87) I,DELTA,X(I),Y(I),YUSMO(I),YSMO(I),DIF,YPS(I),YP SO 333
           1P(I),DYDX,DY2DX,CURV                                      SO 334
            WRITE (JWRITE,89) RLE                                     SO 335
      C                                                               SO 336
      C     CHECK FOR INTERSECTION OF UPPER AND LOWER SURFACES        SO 337
      C                                                               SO 338
      C        DEFINE ITERATION INTERVAL                              SO [illegible]
            KRT=1001                                                  [illegible]
            N1=2*KRT                                                  [illegible]
            TE=THETA(NP)                                              SO [illegible]
            TN=-THETA(1)                                              SO [illegible]
            IF (TN.LT.TE) TE=TN                                       SO 344
            DIF=TE/FLOAT(KRT-1)                                       SO 345
            BB=0.5*DIF                                                SO 346
            AA=0.85*TE                                                SO 347
            YL1=YU1=YSMO(NOSE)                                        SO 348
            TP=TN=0.0                                                 SO 349
            J1=NOSE                                                   SO 350
            J2=2                                                      SO 351
      C        DO-LOOP TO SEARCH FOR INTERSECTION                     SO 352
            DO 59 I=2,N1                                              SO 353
            IF (TP.LE.AA) TN=TN+DIF                                   SO 354
            IF (TP.GT.AA) TN=TN+BB                                    SO 355
            IF (TN.GT.TE) GO TO 61                                    SO 356
            TI=TN                                                     SO 357
      C        FIND UPPER SURFACE Y-COORDINATE AT THETA = TN          SO 358
            DO 54 K=J1,NP                                             SO 359
            J=K-1                                                     SO 360
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      T1=T1*RAD                                                         SO 401
      WRITE (JWRITE,72) T1                                              SO 402
      IERR=1                                                            SO 403
      RETURN                                                            SO 404
C                                                                       SO 405
C        FIND LOCATIONS WHERE DY/DX=0.                                  SO 406
C                                                                       SO 407
61    KRT=0                                                             SO 408
      N1=NP-1                                                           SO 409
      DO 66 I=1,N1                                                      SO 410
      DELTA=THETA(I+1)-THETA(I)                                         SO 411
      AA=(YPP(I)-YPP(I+1))/(2.*DELTA)                                   SO 412
      BB=(YPP(I+1)*THETA(I)-YPP(I)*THETA(I+1))/DELTA                    SO 413
      CC=(YPP(I)*THETA(I+1)**2-YPP(I+1)*THETA(I)**2)/(2.*DELTA)+(YPP(I+1 SO 414
     1)-YPP(I))*DELTA/6.-(YSMO(I+1)-YSMO(I))/DELTA                      SO 415
      GP=BB*BB-4.*AA*CC                                                 SO 416
      IF (GP) 66,62,62                                                  SO 417
62    GP=SQRT(GP)                                                       SO 418
      T1=(-BB+GP)/(2.*AA)                                               SO 419
      T2=(-BB-GP)/(2.*AA)                                               SO 420
      IF (T1.GE.THETA(I).AND.T1.LE.THETA(I+1)) GO TO 63                 SO 421
      GO TO 64                                                          SO 422
63    KRT=KRT+1                                                         SO 423
      WK(KRT,1)=T1                                                      SO 424
64    IF (T2.GE.THETA(I).AND.T2.LE.THETA(I+1)) GO TO 65                 SO 425
      GO TO 66                                                          SO 426
65    KRT=KRT+1                                                         SO 427
      WK(KRT,1)=T2                                                      SO 428
66    CONTINUE                                                          SO 429
      IF (KRT.EQ.0) GO TO 70                                            SO 430
C             FIND X/C AND Y/C WHERE DY/DX=0.                           SO 431
      DO 69 I=1,KRT                                                     SO 432
      CALL INTER (WK(I,1),WK(I,2),NP,THETA,X,1,KTI,0)                   SO 433
      DO 67 J=1,N1                                                      SO 434
      J1=J                                                              SO 435
      J2=J+1                                                            SO 436
      IF (WK(I,1).GE.THETA(J).AND.WK(I,1).LE.THETA(J+1)) GO TO 68       SO 437
67    CONTINUE                                                          SO 438
68    AA=THETA(J2)-WK(I,1)                                              SO 439
      BB=WK(I,1)-THETA(J1)                                              SO 440
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
441          WK(I,1)=WK(I,1)*RAD                                          SO 441
             DELTA=THETA(J2)-THETA(J1)                                    SO 442
        69   WK(I,3)=YPP(J1)*(AA**3/(6.*DELTA)-AA*DELTA/6.)+YPP(J2)*(BB**3/(6.* SO 443
            1DELTA)-BB*DELTA/6.)+(YSMO(J1)*AA+YSMO(J2)*BB)/DELTA          SO 444
445         CONTINUE                                                     SO 445
             IF (KRT.GT.0) WRITE (JWRITE,90) (WK(I,2),WK(I,3),WK(I,1),I=1,KRT) SO 446
        C                                                                SO 447
        C       PRINT RESULTS OF SMOOTHNESS CHECK                        SO 448
        C                                                                SO 449
450          IF (ITER.EQ.0) RETURN                                        SO 450
             WRITE (JWRITE,91) TITLE,DF                                   SO 451
             WRITE (JWRITE,92) (I,A(I,1),A(I,2),I=1,NP)                   SO 452
             WRITE (JWRITE,93) SUM1,SUM2                                  SO 453
             RETURN                                                       SO 454
455     C                                                                SO 455
        C       PRINT WARNING MESSAGE IF ERROR OCCURRED IN CALL TO CSDS  SO 456
        C                                                                SO 457
        71   WRITE (JWRITE,94) IERR                                       SO 458
             RETURN                                                       SO 459
460     C                                                                SO 460
        72   FORMAT (/5X,108HERROR MESSAGE  ---  SMOOTHING PROCESS RESULTED .N SO 461
            1AN INTERSECTION OF THE UPPER AND LOWER SURFACES AT THETA =,F10.3) SO 462
        73   FORMAT (1H1,1X,7HTITLE--,2X,8A10//12X,62H--CHECK OF FIRST DERIVATI SO 463
            1VES GENERATED FROM IOP=2 INPUT DATA---///9X,1HI,5X,12HDY/DT(INPUT), SO 464
465         24X,11HDY/DT(CSDS),8X,3HDIF,6X,13HDY/DT(LSQSMO),8X,3HDIF/)   SO 465
        74   FORMAT (5X,I5,5(5X,F10.6))                                   SO 466
        75   FORMAT (/25X,16HSUM OF SQUARES =,4X,F10.6,20X,F10.6)         SO 467
        76   FORMAT (/10X,25HOUTPUT FROM CSDS SELECTED)                   SO 468
        77   FORMAT (/10X,27HOUTPUT FROM LSQSMO SELECTED)                 SO 469
470     78   FORMAT (1H1,1X,7HTITLE--,2X,8A10//30X,53H--SUM OF SQUARES GENERATE SO 470
            1D DURING SMOOTHING PROCESS--)                                SO 471
        79   FORMAT (/1X,I5,10F12.7)                                      SO 472
        80   FORMAT (1H1,1X,7HTITLE--,2X,8A10//30X,67H--SECOND DERIVATIVES W/R SO 473
            1THETA GENERATED DURING SMOOTHING PROCESS--/4X,1HI,5X,5HTHETA,10(5X SO 474
475         2,6HDY2/DT)/)                                                SO 475
        81   FORMAT (I5,F10.2,10F11.6)                                    SO 476
        82   FORMAT (/1X,14HSUM OF SQUARES,(10F11.6))                     SO 477
        83   FORMAT (/3X,41HSMOOTHING PROCESS HAS NOT CONVERGED AFTER,I4,1X,10H SO 478
            1ITERATIONS)                                                  SO 479
480     84   FORMAT (/3X,33HSMOOTHING PROCESS CONVERGED AFTER,I4,1X,10HITERATIO SO 480
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE YNEW (THETA,YPP,Y,NOSE,NP,YLTE,YNOSE,YUTE,EPS,DUM,WK,JW YW   1
     1RITE,IPT)                                                          YW   2
C                                                                        YW   3
C     ROUTINE TO COMPUTE NEW Y/C COORDINATES USING AN ITERATION          YW   4
C     PROCEDURE THAT INSURES A DESIRED Y/C COORDINATE AT THE NOSE        YW   5
C     (IPT=0) OR THAT INSURES CONTINUITY OF THE FIRST DERIVATIVE W/R TO  YW   6
C     THETA AT THE NOSE (IPT=1)                                          YW   7
C                                                                        YW   8
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          YW   9
C                                                                        YW  10
C        DIMENSION THETA, YPP, Y, AND DUM BY NP AND WK BY 2*NP IN        YW  11
C        CALLING PROGRAM                                                 YW  12
      DIMENSION THETA(1), YPP(1), Y(1), DUM(1), WK(1)                    YW  13
C                                                                        YW  14
C     INITIALIZE ITERATION PARAMETERS                                    YW  15
C                                                                        YW  16
      NMAX=20                                                            YW  17
      N1=-1                                                              YW  18
      DELTA=0.                                                           YW  19
      T1=THETA(NOSE)-THETA(NOSE-1)                                       YW  20
      T2=THETA(NOSE+1)-THETA(NOSE)                                       YW  21
      DO 1 I=1,NP                                                        YW  22
    1 DUM(I)=YPP(I)                                                      YW  23
C                                                                        YW  24
C     ITERATION LOOP TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND         YW  25
C     DERIVATIVE TO INSURE THAT THE DESIRED CONVERGENCE OPTION AT        YW  26
C     THE NOSE IS OBTAINED                                               YW  27
C                                                                        YW  28
    2 N1=N1+1                                                            YW  29
      IF (N1.GT.NMAX) GO TO 11                                           YW  30
      IF (IPT.EQ.1) GO TO 3                                              YW  31
C        IF IPT=0, COMPUTE UPPER AND LOWER SURFACE Y/C COORDINATES       YW  32
C        CONCURRENTLY                                                    YW  33
      CALL INVY (THETA,DUM,1,NP,Y,YLTE,YUTE,WK)                          YW  34
C        COMPUTE DIFFERENCE BETWEEN OUTPUT AND DESIRED Y/C COORDINATE    YW  35
C        AT THE NOSE                                                     YW  36
      DIF=Y(NOSE)-YNOSE                                                  YW  37
      GO TO 4                                                            YW  38
C        IF IPT=1, COMPUTE UPPER AND LOWER SURFACE Y/C COORDINATES       YW  39
C        CONSECUTIVELY                                                   YW  40
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      3     CALL INVY (THETA,DUM,NOSE,NP,Y,YNOSE,YUTE,WK)                YW  41
            CALL INVY (THETA,DUM,1,NOSE,Y,YLTE,YNOSE,WK)                 YW  42
      C        COMPUTE DIFFERENCE BETWEEN UPPER AND LOWER SURFACE FIRST  YW  43
      C        DERIVATIVES AT THE NOSE                                   YW  44
            AA=-DUM(NOSE)*T2/3.-DUM(NOSE+1)*T2/6.+(Y(NOSE+1)-Y(NOSE))/T2 YW  45
            BB=DUM(NOSE-1)*T1/6.+DUM(NOSE)*T1/3.+(Y(NOSE)-Y(NOSE-1))/T1  YW  46
            DIF=AA-BB                                                    YW  47
      C        CHECK FOR CONVERGENCE                                     YW  48
      4     IF (ABS(DIF).LE.EPS) GO TO 9                                 YW  49
      C        COMPUTE ADJUSTMENT VALUE TO SECOND DERIVATIVE             YW  50
            IF (N1.EQ.0) GO TO 6                                         YW  51
            IF (DIF.EQ.DIFP) GO TO 5                                     YW  52
            SP=(DELTA-DELTAP)/(DIF-DIFP)                                 YW  53
            DELTAP=DELTA                                                 YW  54
            DIFP=DIF                                                     YW  55
            DELTA=DELTA-DIF*SP                                           YW  56
            GO TO 7                                                      YW  57
      5     DELTA=0.5*(DELTA+DELTAP)                                     YW  58
            GO TO 7                                                      YW  59
      6     DELTAP=DELTA                                                 YW  60
            DIFP=DIF                                                     YW  61
            DELTA=DELTA+DIF                                              YW  62
      C        ADD ADJUSTMENT VALUE TO SECOND DERIVATIVE                 YW  63
      7     DO 8 I=1,NP                                                  YW  64
      8     DUM(I)=YPP(I)+DELTA                                          YW  65
      C        CONTINUE TO ITERATE                                       YW  66
            GO TO 2                                                      YW  67
      C                                                                  YW  68
      C     PRINT CONVERGENCE MESSAGE                                    YW  69
      C                                                                  YW  70
      9     WRITE (JWRITE,14) N1,DELTA                                   YW  71
      C        REDEFINE THE SECOND DERIVATIVE                            YW  72
            DO 10 I=1,NP                                                 YW  73
      10    YPP(I)=DUM(I)                                                YW  74
            IF (IPT.EQ.1) GO TO 12                                       YW  75
            GO TO 13                                                     YW  76
      C                                                                  YW  77
      C     PRINT NON-CONVERGENCE MESSAGE                                YW  78
      C                                                                  YW  79
      11    N1=N1-1                                                      YW  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      WRITE (JWRITE,15) N1                                              YW  81
C                                                                       YW  82
C     COMPUTE NEW UPPER AND LOWER SURFACE Y/C COORDINATES CONCURRENTLY  YW  83
C                                                                       YW  84
12    CALL INVY (THETA,YPP,1,NP,Y,YLTE,YUTE,WK)                         YW  85
C                                                                       YW  86
C     RETURN TO CALLING PROGRAM                                         YW  87
C                                                                       YW  88
13    RETURN                                                            YW  89
C                                                                       YW  90
14    FORMAT (/3X,88HITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTME YW  91
     1NT TO SECOND DERIVATIVE CONVERGED IN ,I3,23H ITERATIONS AND DELTA  YW  92
     2=,E12.4)                                                          YW  93
15    FORMAT (///10X,40HWARNING THE FOLLOWING ERROR HAS OCCURRED//3X,95H YW  94
     1ITERATION PROCEDURE TO COMPUTE INCREMENTAL ADJUSTMENT TO SECOND DE YW  95
     2RIVATIVE DID NOT CONVERGE IN ,I3,11H ITERATIONS)                  YW  96
      END                                                               YW  97-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE LSQSMO (X,Y,W,YN,YP,YPP,N,IMAX,JMAX,NOSE,WT,EPS,IERR)
C
C     THIS SUBROUTINE IS USED TO SMOOTH X AND Y BY CONSECUTIVELY FITTING
C     A LEAST SQUARES POLYNOMIAL OF DEGREE 4 THRU 7 POINTS AT A TIME
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982
C
      DIMENSION X(1), Y(1), W(1), YN(1), YP(1), YPP(1)
C
      DIMENSION XI(7), YI(7), WW(7), A(5,6), B(5)
C
      COMMON /INOUT/ JREAD,JWRITE,IPRINT
C
C     CHECK NOSE REGION FOR SYMMETRY
C
      ISYM=1
      DO 1 I=1,3
      IF (ABS(X(NOSE-I)+X(NOSE+I)).GT.EPS) ISYM=0
      IF (ABS(Y(NOSE-I)+Y(NOSE+I)).GT.EPS) ISYM=0
    1 CONTINUE
      IERR=0
C
C     FIT A LEAST SQUARES POLYNOMIAL OF DEGREE 4 THRU 7 POINTS
C
      DO 14 I=1,N
C        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
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      GO TO 7                                                   LM  41
    5 J1=NOSE-6                                                 LM  42
      J2=NOSE                                                   LM  43
      GO TO 7                                                   LM  44
    6 J1=NOSE                                                   LM  45
      J2=NOSE+6                                                 LM  46
    7 DO 8 L=J1,J2                                              LM  47
      J=L                                                       LM  48
      IF (I.LE.NOSE) J=J1+J2-L                                  LM  49
      KK=KK+1                                                   LM  50
      WW(KK)=1.0                                                LM  51
      IF (I.EQ.J) WW(KK)=W(I)                                   LM  52
      IF (J.EQ.IMAX.OR.J.EQ.JMAX) WW(KK)=WT*W(J)                LM  53
      XI(KK)=X(J)                                               LM  54
    8 YI(KK)=Y(J)                                               LM  55
      IF (I.LE.4) WW(7)=7.*W(1)                                 LM  56
      IF (I.GE.N-3) WW(7)=7.*W(N)                               LM  57
C        COMPUTE LEAST SQUARES MATRIX                           LM  58
      DO 9 L=1,5                                                LM  59
      DO 9 J=1,6                                                LM  60
    9 A(L,J)=0.                                                 LM  61
      DO 11 K=1,7                                               LM  62
      T1=1.                                                     LM  63
      DO 11 J=1,5                                               LM  64
      T2=T1                                                     LM  65
      DO 10 L=1,5                                               LM  66
      A(J,L)=A(J,L)+T2*WW(K)                                    LM  67
   10 T2=T2*XI(K)                                               LM  68
      A(J,6)=A(J,6)-YI(K)*T1*WW(K)                              LM  69
   11 T1=T1*XI(K)                                               LM  70
C        SOLVE FOR COEFFICIENTS OF LEAST SQUARES POLYNOMIAL     LM  71
      DO 12 K=1,4                                               LM  72
      DO 12 J=K,4                                               LM  73
      T1=A(J+1,K)/A(K,K)                                        LM  74
      DO 12 L=K,6                                               LM  75
   12 A(J+1,L)=A(J+1,L)-A(K,L)*T1                               LM  76
      B(5)=-A(5,6)/A(5,5)                                       LM  77
      DO 13 L=2,5                                               LM  78
      K=6-L                                                     LM  79
      B(K)=-A(K,6)/A(K,K)                                       LM  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
81          K1=K+1                                                    LM  81
            DO 13 J=K1,5                                              LM  82
      13    B(K)=B(K)-B(J)*A(K,J)/A(K,K)                              LM  83
      C        COMPUTE NEW Y , FIRST , AND SECOND DERIVATIVE          LM  84
85          YN(I)=(((B(5)*X(I)+B(4))*X(I)+B(3))*X(I)+B(2))*X(I)+B(1)  LM  85
            YP(I)=((4.*B(5)*X(I)+3.*B(4))*X(I)+2.*B(3))*X(I)+B(2)     LM  86
            YPP(I)=(12.*B(5)*X(I)+6.*B(4))*X(I)+2.*B(3)               LM  87
      14    CONTINUE                                                  LM  88
            IF (ISYM.EQ.0) RETURN                                     LM  89
90          YN(NOSE)=0.0                                              LM  90
            YPP(NOSE)=0.0                                             LM  91
            YP(NOSE)=1.0                                              LM  92
            RETURN                                                    LM  93
            END                                                       LM  94-
```


C-2

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

CARD NO.

```
C*        COEF     A TWO-DIMENSIONAL OUTPUT ARRAY DIMENSIONED (MAX,4)  * CS  41
C*                 IN THE CALLING PROGRAM.  UPON RETURN, COEF(I,J)     * CS  42
C*                 CONTAINS THE J-TH COEFFICIENT OF THE SPLINE FOR     * CS  43
C*                 THE INTERVAL BEGINNING AT POINT X(I).  THE          * CS  44
C*                 FUNCTIONAL VALUE OF THE SPLINE AT ABSCISSA X1,      * CS  45
C*                 WHERE X(I)<X1<X(I+1), IS GIVEN BY:                  * CS  46
C*                    F(X1)=(((COEF(I,4)*H+COEF(I,3))*H+COEF(I,2))*H   * CS  47
C*                    +COEF(I,1)                                       * CS  48
C*                 WHERE H=X1-X(I)                                     * CS  49
C*                                                                     * CS  50
C*          WK     A ONE-DIMENSIONAL WORK AREA ARRAY DIMENSIONED AT    * CS  51
C*                 LEAST (7*IX+9) IN THE CALLING PROGRAM.              * CS  52
C*                                                                     * CS  53
C*        IERR     OUTPUT ERROR PARAMETER:                             * CS  54
C*                 =0   NORMAL RETURN.  NO ERROR DETECTED.             * CS  55
C*                 =J   THE J-TH ELEMENT OF THE X ARRAY IS NOT IN      * CS  56
C*                      STRICTLY INCREASING ORDER.                     * CS  57
C*                 =-1  THERE ARE LESS THAN FOUR VALUES IN THE X ARRAY.* CS  58
C*                                                                     * CS  59
C*                 UPON RETURN FROM CSDS, THIS PARAMETER SHOULD BE     * CS  60
C*                 TESTED IN THE CALLING PROGRAM.                      * CS  61
C*                                                                     * CS  62
C*                                                                     * CS  63
C*                                                                     * CS  64
C*    REQUIRED ROUTINES                -NONE                           * CS  65
C*                                                                     * CS  66
C*    SOURCE                           IMSL ROUTINE ICSSMU MODIFIED BY * CS  67
C*                                     COMPUTER SCIENCES CORPORATION   * CS  68
C*                                                                     * CS  69
C*    LANGUAGE                         -FORTRAN                        * CS  70
C*                                                                     * CS  71
C*    DATE RELEASED                    SEPTEMBER 5, 1972               * CS  72
C*                                                                     * CS  73
C*    LATEST REVISION                  MARCH 1975                      * CS  74
C*********************************************************************** CS  75
C                                                                        CS  76
C                                                                        CS  77
      DIMENSION X(1), F(1), DF(1), COEF(MAX,4), WK(1)                    CS  78
C                                                                        CS  79
C                                 SET UP WORKING AREAS                   CS  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C                                                                       CS  81
      IERR=0                                                            CS  82
      IF (IPT.NE.-1) GO TO 4                                            CS  83
      IPT=0                                                             CS  84
      IF (IX.LT.4) GO TO 1                                              CS  85
      GO TO 2                                                           CS  86
1     IERR=-1                                                           CS  87
      RETURN                                                            CS  88
2     IX1=IX-1                                                          CS  89
      DO 3 I=1,IX1                                                      CS  90
      IF (X(I+1)-X(I).GT.0) GO TO 3                                     CS  91
      IERR=I+1                                                          CS  92
      RETURN                                                            CS  93
3     CONTINUE                                                          CS  94
      NP1=IX+1                                                          CS  95
      IB1=NP1                                                           CS  96
      IB2=IB1+NP1                                                       CS  97
      IB3=IB2+NP1+1                                                     CS  98
      IB4=IB3+NP1                                                       CS  99
      IB5=IB4+NP1                                                       CS 100
      IB6=IB5+NP1+1                                                     CS 101
      WK(1)=0.                                                          CS 102
      WK(2)=0.                                                          CS 103
      WK(IB2)=0.                                                        CS 104
      WK(IB3)=0.                                                        CS 105
      IJK2=IB2+NP1                                                      CS 106
      WK(IJK2)=0.                                                       CS 107
      IJK5=IB5+1                                                        CS 108
      WK(IJK5)=0.                                                       CS 109
      IJK5=IB5+2                                                        CS 110
      WK(IJK5)=0.                                                       CS 111
      WK(IB6)=0.                                                        CS 112
      IJK5=IB5+NP1                                                      CS 113
      WK(IJK5)=0.                                                       CS 114
4     CONTINUE                                                          CS 115
      P=0.                                                              CS 116
      H=X(2)-X(1)                                                       CS 117
      F2=-S                                                             CS 118
      FF=(F(2)-F(1))/H                                                  CS 119
      IF (IX.LT.3) GO TO 10                                             CS 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

 161                WK(IJK5)=COEF(I-1,1)-WK(IJK1)*WK(IJKN)-WK(IJK2)*WK(IJKO)        CS 161
                    IJK4=IB4+I                                                       CS 162
                    FF=P*COEF(I-1,3)+WK(IJK4)-H*WK(IJK1)                             CS 163
                    G=H                                                              CS 164
 165                H=COEF(I-1,4)*P                                                  CS 165
              8     CONTINUE                                                         CS 166
                    DO 9 I=3,IX                                                      CS 167
                    J=IX-I+3                                                         CS 168
                    IJK5=IB5+J                                                       CS 169
 170                IJK6=IJK5+1                                                      CS 170
                    IJK7=IJK6+1                                                      CS 171
                    IJK1=IB1+J                                                       CS 172
                    IJK2=IB2+J                                                       CS 173
                    WK(IJK5)=WK(J)*WK(IJK5)-WK(IJK1)*WK(IJK6)-WK(IJK2)*WK(IJK7)      CS 174
 175          9     CONTINUE                                                         CS 175
             10     E=0                                                              CS 176
                    H=0                                                              CS  7
              C                                                                      CS 178
              C                                 COMPUTE U AND ACCUMULATE E           CS 179
 180          C                                                                      CS 180
                    DO 11 I=2,IX                                                     CS 181
                    G=H                                                              CS 182
                    IJK5=IB5+I                                                       CS 183
                    H=(WK(IJK5+1)-WK(IJK5))/(X(I)-X(I-1))                            CS 184
 185                IJK6=IB6+I                                                       CS 185
                    WK(IJK6)=(H-G)*DF(I-1)*DF(I-1)                                   CS 186
                    E=E+WK(IJK6)*(H-G)                                               CS 187
             11     CONTINUE                                                         CS 188
                    G=-H*DF(IX)*DF(IX)                                               CS 189
 190                IJK6=IB6+NP1                                                     CS 190
                    WK(IJK6)=G                                                       CS 191
                    E=E-G*H                                                          CS 192
                    G=F2                                                             CS 193
                    F2=E*P*P                                                         CS 194
 195                IF (F2.GE.S.OR.F2.LE.G) GO TO 14                                 CS 195
                    FF=0.                                                            CS 196
                    IJK6=IB6+2                                                       CS 197
                    H=(WK(IJK6+1)-WK(IJK6))/(X(2)-X(1))                              CS 198
                    IF (IX.LT.3) GO TO 13                                            CS 199
 200                DO 12 I=3,IX                                                     CS 200
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      G=H                                                               CS 201
      IJK6=IB6+I                                                        CS 202
      H=(WK(IJK6+1)-WK(IJK6))/(X(I)-X(I-1))                             CS 203
      IJK1=IB1+I-1                                                      CS 204
      IJK2=IB2+I-2                                                      CS 205
      G=H-G-WK(IJK1)*WK(I-1)-WK(IJK2)*WK(I-2)                           CS 206
      FF=FF+G*WK(I)*G                                                   CS 207
      WK(I)=G                                                           CS 208
   12 CONTINUE                                                          CS 209
   13 H=E-P*FF                                                          CS 210
      IF (H.LE.0) GO TO 14                                              CS 211
C                                                                       CS 212
C                                   UPDATE THE LAGRANGE MULTIPLIER P    CS 213
C                                      FOR THE NEXT ITERATION           CS 214
C                                                                       CS 215
      P=P+(S-F2)/((SQRT(S/E)+P)*H)                                      CS 216
      GO TO 7                                                           CS 217
C                                                                       CS 218
C                                   IF E LESS THAN OR EQUAL TO S,       CS 219
C                                   COMPUTE THE COEFFICIENTS AND RETURN. CS 220
C                                                                       CS 221
   14 DO 15 I=2,NP1                                                     CS 222
      IJK6=IB6+I                                                        CS 223
      COEF(I-1,1)=F(I-1)-P*WK(IJK6)                                     CS 224
      IJK5=IB5+I                                                        CS 225
      COEF(I-1,3)=WK(IJK5)                                              CS 226
   15 CONTINUE                                                          CS 227
      DO 16 I=2,IX                                                      CS 228
      H=X(I)-X(I-1)                                                     CS 229
      COEF(I-1,4)=(COEF(I,3)-COEF(I-1,3))/(3.*H)                        CS 230
      COEF(I-1,2)=(COEF(I,1)-COEF(I-1,1))/H-(H*COEF(I-1,4)+COEF(I-1,3))* CS 231
     1H                                                                 CS 232
   16 CONTINUE                                                          CS 233
      RETURN                                                            CS 234
      END                                                               CS 235-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE PCARD (IPUNCH,X,Y,W,THETA,YSMO,YPS,YPPS,NOSE,NP,CHORD,T PH   1
     1ITLE)                                                              PH   2
C                                                                        PH   3
C     ROUTINE TO PUNCH OUTPUT DATA   (TAPE 1 IS PUNCH FILE)              PH   4
C                                                                        PH   5
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          PH   6
C                                                                        PH   7
      DIMENSION TITLE(8), X(1), Y(1), W(1), THETA(1), YSMO(1), YPS(1), Y PH   8
     1PPS(1)                                                             PH   9
C                                                                        PH  10
      COMMON /HLM/ DX(200),DY(200),DW(200)                               PH  11
C                                                                        PH  12
      COMMON /BLK1/ PI,PI2,RAD,CONS                                      PH  13
C                                                                        PH  14
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                 PH  15
C                                                                        PH  16
      IF (IPUNCH.LE.0.OR.IPUNCH.GE.5) RETURN                             PH  17
C                                                                        PH  18
C     PUNCH TITLE CARD                                                   PH  19
C                                                                        PH  20
      WRITE (JWRITE,10) IPUNCH,TITLE                                     PH  21
      WRITE (1,11) TITLE                                                 PH  22
C                                                                        PH  23
C     DETERMINE OUTPUT PUNCH OPTION                                      PH  24
C                                                                        PH  25
      IOP=0                                                              PH  26
      IF (IPUNCH.EQ.2) IOP=1                                             PH  27
      IF (IPUNCH.EQ.3) IOP=2                                             PH  28
      IF (IPUNCH.EQ.4) IOP=3                                             PH  29
      WRITE (JWRITE,12) IOP                                              PH  30
      XI=FLOAT(IOP)                                                      PH  31
      WRITE (1,13) XI                                                    PH  32
C                                                                        PH  33
C     PUNCH UPPER SURFACE QUANTITIES                                     PH  34
C                                                                        PH  35
      J=KP=0                                                             PH  36
      DO 1 I=NOSE,NP                                                     PH  37
      J=J+1                                                              PH  38
      DW(J)=W(I)                                                         PH  39
      IF (W(I).GT.1.0) KP=1                                              PH  40
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      IF (IOP.EQ.0) DX(J)=X(I)*CHORD                                    PH  41
      IF (IOP.NE.0) DX(J)=THETA(I)*RAD                                  PH  42
      IF (IOP.EQ.0) DY(J)=YSMO(I)*CHORD                                 PH  43
      IF (IOP.EQ.1) DY(J)=YSMO(I)                                       PH  44
      IF (IOP.EQ.2) DY(J)=YPS(I)                                        PH  45
      IF (IOP.EQ.3) DY(J)=YPPS(I)                                       PH  46
    1 CONTINUE                                                          PH  47
      WRITE (JWRITE,14) J                                               PH  48
      XI=FLOAT(J)                                                       PH  49
      WRITE (1,15) XI                                                   PH  50
      IF (IOP.EQ.0) WRITE (JWRITE,16) (DX(I),I=1,J)                     PH  51
      IF (IOP.NE.0) WRITE (JWRITE,7) (DX(I),I=1,J)                      PH  52
      WRITE (JWRITE,17) (DY(I),I=1,J)                                   PH  53
      IF (KP.EQ.1) WRITE (JWRITE,21) (DW(I),I=1,J)                      PH  54
      DO 3 I=1,J                                                        PH  55
      IF (IOP.NE.0) GO TO 2                                             PH  56
      IF (DW(I).GT.1.0) WRITE (1,22) DX(I),DY(I),DW(I)                  PH  57
      IF (DW(I).LE.1.0) WRITE (1,18) DX(I),DY(I)                        PH  58
      GO TO 3                                                           PH  59
    2 IF (DW(I).GT.1.0) WRITE (1,8) DX(I),DY(I),DW(I)                   PH  60
      IF (DW(I).LE.1.0) WRITE (1,9) DX(I),DY(I)                         PH  61
    3 CONTINUE                                                          PH  62
    C                                                                   PH  63
    C PUNCH LOWER SURFACE QUANTITIES                                    PH  64
    C                                                                   PH  65
      J=KP=0                                                            PH  66
      DO 4 I=1,NOSE                                                     PH  67
      J=J+1                                                             PH  68
      K=NOSE+1-I                                                        PH  69
      DW(J)=W(K)                                                        PH  70
      IF (W(K).GT.1.0) KP=1                                             PH  71
      IF (IOP.EQ.0) DX(J)=X(K)*CHORD                                    PH  72
      IF (IOP.NE.0) DX(J)=THETA(K)*RAD                                  PH  73
      IF (IOP.EQ.0) DY(J)=YSMO(K)*CHORD                                 PH  74
      IF (IOP.EQ.1) DY(J)=YSMO(K)                                       PH  75
      IF (IOP.EQ.2) DY(J)=YPS(K)                                        PH  76
      IF (IOP.EQ.3) DY(J)=YPPS(K)                                       PH  77
    4 CONTINUE                                                          PH  78
      WRITE (JWRITE,19) J                                               PH  79
      XI=FLOAT(J)                                                       PH  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      WRITE (1,15) XI                                                   PH  81
      IF (IOP.EQ.0) WRITE (JWRITE,16) (DX(I),I=1,J)                     PH  82
      IF (IOP.NE.0) WRITE (JWRITE,7) (DX(I),I=1,J)                      PH  83
      WRITE (JWRITE,17) (DY(I),I=1,J)                                   PH  84
      IF (KP.EQ.1) WRITE (JWRITE,21) (DW(I),I=1,J)                      PH  85
      DO 6 I=1,J                                                        PH  86
      IF (IOP.NE.0) GO TO 5                                             PH  87
      IF (DW(I).GT.1.0) WRITE (1,22) DX(I),DY(I),DW(I)                  PH  88
      IF (DW(I).LE.1.0) WRITE (1,18) DX(I),DY(I)                        PH  89
      GO TO 6                                                           PH  90
    5 IF (DW(I).GT.1.0) WRITE (1,8) DX(I),DY(I),DW(I)                   PH  91
      IF (DW(I).LE.1.0) WRITE (1,9) DX(I),DY(I)                         PH  92
    6 CONTINUE                                                          PH  93
    C                                                                   PH  94
    C PUNCH YLTE AND YUTE                                               PH  95
    C                                                                   PH  96
      IF (IOP.LE.1) RETURN                                              PH  97
      YLTE=YSMO(1)                                                      PH  98
      YNOSE=YSMO(NOSE)                                                  PH  99
      YUTE=YSMO(NP)                                                     PH 100
      WRITE (JWRITE,20) YLTE,YNOSE,YUTE                                 PH 101
      WRITE (1,18) YLTE,YNOSE,YUTE                                      PH 102
    C                                                                   PH 103
    C RETURN TO CALLING PROGRAM                                         PH 104
    C                                                                   PH 105
      RETURN                                                            PH 106
    C                                                                   PH 107
    7 FORMAT (/3X,4HTH =,8F10.5/(7X,8F10.5))                            PH 108
    8 FORMAT (F10.5,F10.6,F10.2)                                        PH 109
    9 FORMAT (F10.5,F10.6)                                              PH 110
   10 FORMAT (1H1,10X,36HTHE FOLLOWING DATA HAVE BEEN PUNCHED,5X,7HIPUNC PH 111
     1H=,I4//3X,8A10)                                                   PH 112
   11 FORMAT (8A10)                                                     PH 113
   12 FORMAT (/3X,5HIOP =,I4)                                           PH 114
   13 FORMAT (30X,F10.2)                                                PH 115
   14 FORMAT (/3X,4HNU =,I4)                                            PH 116
   15 FORMAT (F10.2)                                                    PH 117
   16 FORMAT (/3X,4HDX =,8F10.6/(7X,8F10.6))                            PH 118
   17 FORMAT (/3X,4HDY =,8F10.6/(7X,8F10.6))                            PH 119
   18 FORMAT (3F10.6)                                                   PH 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

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,8F10.2))
            22    FORMAT (2F10.6,F10.2)                                                     PH 124
 125              END                                                                       PH 125
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
41                J=K-1                                                    PF  41
                  IF (YP.GE.THETA(J).AND.YP.LE.THETA(K)) GO TO 4           PF  42
            3     CONTINUE                                                 PF  43
            4     M=J                                                      PF  44
45                DELTA=THETA(J+1)-THETA(J)                                PF  45
                  X2=THETA(J+1)-YP                                         PF  46
                  X1=YP-THETA(J)                                           PF  47
                  YI(I)=YPPS(J)*(X2**3/(6.*DELTA)-X2*DELTA/6.)+YPPS(J+1)*(X1**3/(6.* PF  48
                 1DELTA)-X1*DELTA/6.)+(YSMO(J)*X2+YSMO(J+1)*X1)/DELTA      PF  49
50                YPSI(I)=YPPS(J)*(DELTA/6.-X2**2/(2.*DELTA))+YPPS(J+1)*(X1*X1/(2.*D PF  50
                 1ELTA)-DELTA/6.)+(YSMO(J+1)-YSMO(J))/DELTA                PF  51
                  IF (ABS(YI(I)).GE.YMAX) YMAX=ABS(YI(I))                  PF  52
            5     CONTINUE                                                 PF  53
            C                                                              PF  54
55          C        PRINT INTERPOLATED Y/C-COORDINATES                    PF  55
            C                                                              PF  56
                  IF (IPRINT.NE.0) GO TO 6                                 PF  57
                  WRITE (JWRITE,15) TITLE                                  PF  58
                  WRITE (JWRITE,16) (I,TI(I),YI(I),YI(I),I=1,NM)           PF  59
60          C                                                              PF  60
            C        DETERMINE SCALING FACTOR FOR Y/C AXIS                 PF  61
            C                                                              PF  62
            6     YSCALE=0.1                                               PF  63
                  IF (YMAX.LE.0.06) YSCALE=0.01                            PF  64
65                IF ((YMAX.GT.0.06).AND.(YMAX.LE.0.12)) YSCALE=0.02       PF  65
                  IF ((YMAX.GT.0.12).AND.(YMAX.LE.0.24)) YSCALE=0.04       PF  66
                  IF ((YMAX.GT.0.24).AND.(YMAX.LE.0.30)) YSCALE=0.05       PF  67
                  YMIN=-6.*YSCALE                                          PF  68
                  YSAV=YSCALE                                              PF  69
70          C                                                              PF  70
            C        DRAW AND LABEL Y/C AND THETA AXIS                     PF  71
            C                                                              PF  72
                  IF (IPLOT.EQ.2) GO TO 11                                 PF  73
                  CALL CALPLT (2.,1.,-3)                                   PF  74
75                CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)                      PF  75
                  CALL AXES (0.,2.,0.,36.,-180.,10.,-2.,1.,10HTHETA,DEG.,SIZ,-10,0) PF  76
                  CALL AXES (0.,2.,90.,12.,YMIN,YSCALE,-1.,0.,3HY/C,SIZ,3,2) PF  77
                  CALL NOTATE (0.,0,13.1,.4,2,0.,-1)                       PF  78
                  CALL NOTATE (1.5,12.9,SIZ,8HSMOOTHED,0.,8)               PF  79
80                CALL NOTATE (1.0,13.7,.4,3,0.,-1)                        PF  80
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

 81              CALL NOTATE (1.5,13.5,SIZ,5HINPUT,0.,5)                    PF  81
                 CALL CALPLT (0.,8.,-3)                                     PF  82
            C                                                               PF  83
            C       PLOT INPUT Y/C-COORDINATES VS THETA                     PF  84
 85         C                                                               PF  85
                 DO 7 I=1,NP                                                PF  86
                 TP=THETA(I)*RAD/10.+18.0                                   PF  87
                 YP=Y(I)/YSCALE                                             PF  88
                 CALL PNTPLT (TP,YP,22,ISIZ)                                PF  89
 90          7   CONTINUE                                                   PF  90
            C                                                               PF  91
            C       PLOT SMOOTHED Y/C-COORDINATES VS THETA                  PF  92
            C                                                               PF  93
                 TI(NM+1)=-180.0                                            PF  94
 95              TI(NM+2)=10.0                                              PF  95
                 YI(NM+1)=0.                                                PF  96
                 YI(NM+2)=YSCALE                                            PF  97
                 CALL LINE (TI,YI,NM,1,0,0,0.)                              PF  98
            C                                                               PF  99
100         C       DETERMINE SCALING FACTOR FOR FIRST DERIVATIVE AXIS (YP AXIS)  PF 100
            C                                                               PF 101
                 YMAX=0.0                                                   PF 102
                 DO 8 I=1,NM                                                PF 103
                 IF (ABS(YPSI(I)).GT.YMAX) YMAX=ABS(YPSI(I))                PF 104
105          8   CONTINUE                                                   PF 105
                 CSCALE=.1                                                  PF 106
                 IF ((YMAX.LE.0.30).AND.(YMAX.GT.0.24)) CSCALE=.05          PF 107
                 IF ((YMAX.LE.0.24).AND.(YMAX.GT.0.12)) CSCALE=.04          PF 108
                 IF ((YMAX.LE.0.12).AND.(YMAX.GT.0.06)) CSCALE=.02          PF 109
110              IF ((YMAX.LE.0.06).AND.(YMAX.GE.0.00)) CSCALE=.01          PF 110
                 CMIN=-6.*CSCALE                                            PF 111
            C                                                               PF 112
            C       DETERMINE SCALING FACTOR FOR SECOND DERIVATIVE AXIS (YPP AXIS)  PF 113
            C                                                               PF 114
115              YMAX=0.0                                                   PF 115
                 DO 9 I=1,NP                                                PF 116
                 IF (ABS(YPPS(I)).GT.YMAX) YMAX=ABS(YPPS(I))                PF 117
            9    CONTINUE                                                   PF 118
                 YSCALE=1.                                                  PF 119
120              IF ((YMAX.LE.3.00).AND.(YMAX.GT.2.40)) YSCALE=.5           PF 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      IF ((YMAX.LE.2.40).AND.(YMAX.GT.1.20)) YSCALE=.4                  PF 121
      IF ((YMAX.LE.1.20).AND.(YMAX.GT.0.60)) YSCALE=.2                  PF 122
      IF ((YMAX.LE.0.60).AND.(YMAX.GT.0.30)) YSCALE=.1                  PF 123
      IF ((YMAX.LE.0.30).AND.(YMAX.GT.0.24)) YSCALE=.05                 PF 124
      IF ((YMAX.LE.0.24).AND.(YMAX.GT.0.12)) YSCALE=.04                 PF 125
      IF ((YMAX.LE.0.12).AND.(YMAX.GT.0.06)) YSCALE=.02                 PF 126
      IF ((YMAX.LE.0.06).AND.(YMAX.GE.0.00)) YSCALE=.01                 PF 127
      YMIN=-6.*YSCALE                                                   PF 128
C                                                                       PF 129
C        DRAW AND LABEL YP, YPP, AND THETA AXIS                         PF 130
C                                                                       PF 131
      CALL CALPLT (0.,8.,-3)                                            PF 132
      CALL AXES (0.,0.,0.,36.,-180.,10.,-2.,1.,10HTHETA,DEG.,SIZ,-10,0) PF 133
C             DRAW AND LABEL YPS AXES                                   PF 134
      CALL AXES (0.,0.,90.,12.,CMIN,CSCALE,-1.,0.,3HYPS,SIZ,3,2)        PF 135
C             DRAW AND LABEL YPPS AXES                                  PF 136
      CALL AXES (36.,0.,90.,12.,YMIN,YSCALE,-1.,0.,4HYPPS,SIZ,-4,2)     PF 137
      CALL NOTATE (1.0,11.1,.4,3,0.,-1)                                 PF 138
      CALL NOTATE (1.5,10.9,SIZ,4HYPPS,0.,4)                            PF 139
      CALL NOTATE (1.0,11.7,.4,2,0.,-1)                                 PF 140
      CALL NOTATE (1.5,11.5,SIZ,3HYPS,0.,3)                             PF 141
      CALL CALPLT (0.,6.,-3)                                            PF 142
C                                                                       PF 143
C        PLOT SMOOTHED FIRST DERIVATIVES YP VS THETA                    PF 144
C                                                                       PF 145
      YPSI(NM+1)=0.0                                                    PF 146
      YPSI(NM+2)=CSCALE                                                 PF 147
      CALL LINE (TI,YPSI,NM,1,0,0,0.)                                   PF 148
C                                                                       PF 149
C        PLOT SMOOTHED SECOND DERIVATIVES YPP VS THETA                  PF 150
C                                                                       PF 151
      THETA(NP+1)=-PI                                                   PF 152
      THETA(NP+2)=10./RAD                                               PF 153
      YPPS(NP+1)=0.0                                                    PF 154
      YPPS(NP+2)=YSCALE                                                 PF 155
      CALL LINE (THETA,YPPS,NP,1,0,0,0.)                                PF 156
      DO 10 I=1,NP                                                      PF 157
      TP=THETA(I)*RAD/10.+18.0                                          PF 158
      YP=YPPS(I)/YSCALE                                                 PF 159
      CALL PNTPLT (TP,YP,22,ISIZ)                                       PF 160
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
  10    CONTINUE                                                        PF 161
        CALL NFRAME                                                     PF 162
C               CHECK PLOT OPTION                                       PF 163
        IF (IPLOT.EQ.1.OR.IPLOT.EQ.  RETURN                             PF 164
        IF (IPLOT.EQ.8) RETURN                                          PF 165
C                                                                       PF 166
C          DETERMINE SCALING FACTOR FOR Y/C AXIS                        PF 167
C                                                                       PF 168
  11    IF (YSAV.EQ.0.01) YMAX=8                                        PF 169
        IF (YSAV.EQ.0.02) YMAX=12                                       PF 170
        IF (YSAV.EQ.0.04) YMAX=20                                       PF 171
        IF (YSAV.EQ.0.05) YMAX=24                                       PF 172
        YMIN=-0.0125*YMAX                                               PF 173
C                                                                       PF 174
C          PLOT INPUT AND SMOOTHED Y/C-COORDINATES VS X/C               PF 175
C                                                                       PF 176
C          DRAW AND LABEL Y/C AND X/C AXIS                              PF 177
C                                                                       PF 178
        CALL CALPLT (2.,2.,-3)                                          PF 179
        CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)                             PF 180
        CALL CALPLT (0.,2.,-3)                                          PF 181
        CALL AXES (0.,0.,0.,40.,0.,.025,-2.,1.,3HX/C,SIZ,-3,2)          PF 182
        CALL AXES (0.,0.,90.,YMAX,YMIN,0.025,-2.,1.,3HY/C,SIZ,3,2)      PF 183
        YP=YMAX-0.9                                                     PF 184
        CALL NOTATE (1.0,YP,SIZ,2,0.,-1)                                PF 185
        YP=YMAX-1.1                                                     PF 186
        CALL NOTATE (1.5,YP,SIZ,8HSMOOTHED,0.,8)                        PF 187
        YP=YMAX-.3                                                      PF 188
        CALL NOTATE (1.0,YP,SIZ,3,0.,-1)                                PF 189
        YP=YMAX-.5                                                      PF 190
        CALL NOTATE (1.5,YP,SIZ,5HINPUT,0.,5)                           PF 191
        YP=0.5*YMAX                                                     PF 192
        CALL CALPLT (0.,YP,-3)                                          PF 193
C                                                                       PF 194
C          PLOT INPUT Y/C-COORDINATES                                   PF 195
C                                                                       PF 196
        DO 14 I=1,NP                                                    PF 197
        TP=ABS(THETA(I))                                                PF 198
        IF (TP.LE.PI2) GO TO 12                                         PF 199
        XP=CONS*(ATAN(SINH(TP-PI2))+1.)/.025                            PF 200
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      GO TO 13                                                          PF 201
   12 XP=CONS*(1.-COS(TP))/.025                                         PF 202
   13 YP=Y(I)/.025                                                      PF 203
      CALL PNTPLT (XP,YP,22,ISIZ)                                       PF 204
   14 CONTINUE                                                          PF 205
C                                                                       PF 206
C        PLOT SMOOTHED Y/C-COORDINATES                                  PF 207
C                                                                       PF 208
      XI(NM+1)=YI(NM+1)=0.0                                             PF 209
      XI(NM+2)=YI(NM+2)=.025                                            PF 210
      CALL LINE (XI-YI,NM,1,0,0,0.)                                     PF 211
      CALL NFRAME                                                       PF 212
C                                                                       PF 213
C        RETURN TO CALLING PROGRAM                                      PF 214
C                                                                       PF 215
      RETURN                                                            PF 216
C                                                                       PF 217
   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 PF 219
     2/C,7X,3HY/C,12X,1HI,3X,5HTHETA,5X,3HX/C,7X,3HY/C/)                PF 220
   16 FORMAT (3(7X,I4,F8.2,2F10.6))                                     PF 221
      END                                                               PF 222-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
C                                                                       PC  81
      CALL GRIDCK                                                       PC  82
      CALL CALPLT (2.,2.,-3)                                            PC  83
      CALL NOTATE (0.,0.,SIZ,TITLE,0.,80)                               PC  84
      CALL CALPLT (0.,2.,-3)                                            PC  85
      CALL AXES (0.,0.,0.,36.,-180.,10.,-2.,1.,10HTHETA,DEG.,SIZ,-10,0) PC  86
      CALL AXES (0.,0.,90.,20.,0.,CMAX,-2.,1.,15HSQRT(CURVATURE),SIZ,15, PC  87
     12)                                                                PC  88
C                                                                       PC  89
C        PLOT INTERPOLATED CURVATURE POINTS                             PC  90
C                                                                       PC  91
      TI(NM+1)=-180.0                                                   PC  92
      CURV(NM+1)=0.0                                                    PC  93
      TI(NM+2)=10.                                                      PC  94
      CURV(NM+2)=CMAX                                                   PC  95
      CALL LINE (TI,CURV,NM,1,0,0,0.0)                                  PC  96
C                                                                       PC  97
C        COMPUTE AND PLOT CURVATURE AT INPUT THETA POINTS               PC  98
C                                                                       PC  99
      DO 14 I=1,NP                                                      PC 100
      DELTA=YPS(I)                                                      PC 101
      IF (THETA(I).LE.0.0) DELTA=-DELTA                                 PC 102
      TP=ABS(THETA(I))                                                  PC 103
      IF (TP.GT.PI2) GO TO 10                                           PC 104
      GP=CONS*SIN(TP)                                                   PC 105
      GPP=CONS*COS(TP)                                                  PC 106
      GO TO 11                                                          PC 107
   10 T1=COSH(TP-PI2)                                                   PC 108
      T2=SINH(TP-PI2)                                                   PC 109
      GP=CONS/T1                                                        PC 110
      GPP=-CONS*T2/(T1*T1)                                              PC 111
   11 IF (TP.LE.0.0.OR.GP.EQ.0.0) GO TO 12                              PC 112
      DYDX=DELTA/GP                                                     PC 113
      DY2DX=(YPPS(I)*GP-DELTA*GPP)/(GP**3)                              PC 114
      T1=ABS(DY2DX)/(SQRT(1.+DYDX**2)**3)                               PC 115
      GO TO 13                                                          PC 116
   12 T1=CONS/(DELTA*DELTA)                                             PC 117
   13 T2=THETA(I)*RAD/10.+18.0                                          PC 118
      T1=SQRT(T1)/CMAX                                                  PC 119
      CALL PNTPLT (T2,T1,22,ISIZ)                                       PC 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      NL=NOSE                                                           CK  41
      J=NOSE+1                                                          CK  42
      DO 4 I=1,NOSE                                                     CK  43
      J=J-1                                                             CK  44
      TL(J)=THETA(I)                                                    CK  45
      YL(J)=YSMO(I)                                                     CK  46
      TP=ABS(THETA(I))                                                  CK  47
      IF (TP.GT.PI2) GO TO 3                                            CK  48
      XL(J)=CONS*(1.-COS(TP))                                           CK  49
      GO TO 4                                                           CK  50
    3 XL(J)=CONS*(ATAN(SINH(TP-PI2))+1.)                                CK  51
    4 YPPL(J)=YPPS(I)                                                   CK  52
    C         COMPUTE FIRST DERIVATIVES OF UPPER SURFACE                CK  53
      DO 5 I=2,NU                                                       CK  54
      DELTA=TU(I)-TU(I-1)                                               CK  55
      DYXU(I)=YPPU(I)*DELTA/3.+YPPU(I-1)*DELTA/6.+(YU(I)-YU(I-1))/DELTA CK  56
      IF (TU(I).LE.PI2) DYXU(I)=DYXU(I)/(CONS*SIN(TU(I)))               CK  57
      IF (TU(I).GT.PI2) DYXU(I)=DYXU(I)*COSH(TU(I)-PI2)/CONS            CK  58
    5 CONTINUE                                                          CK  59
      DYXU(1)=0.1E99                                                    CK  60
    C                                                                   CK  61
    C    COMPUTE THICKNESS AND CAMBER DISTRIBUTIONS BY FINDING LOWER    CK  62
    C    SURFACE COORDINATE (XLS,YLS) CORRESPONDING TO INPUT UPPER      CK  63
    C    SURFACE COORDINATE (XU,YU)                                     CK  64
    C                                                                   CK  65
      NT=0                                                              CK  66
      KSAVE=1                                                           CK  67
      NS=1                                                              CK  68
      NL1=NL-1                                                          CK  69
      NM1=NM-1                                                          CK  70
      A1=PI/FLOAT(NM1)                                                  CK  71
      DEL=1./(FLOAT(NM1)**2)                                            CK  72
      DO 12 I=1,NU                                                      CK  73
    C         LOAD XU AND YU                                            CK  74
      IJ=NU+1-I                                                         CK  75
      XXU=XU(IJ)                                                        CK  76
      YYU=YU(IJ)                                                        CK  77
      DYU=DYXU(IJ)                                                      CK  78
      NN=1                                                              CK  79
    C         FIND XLS                                                  CK  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DO 9 K=NS,NM                                                      CK  81
      TP=A1*FLOAT(NM-K)                                                 CK  82
      IF (K.EQ.1) TP=ABS(TL(NL))                                        CK  83
      IF (K.EQ.NM) TP=ABS(TL(1))                                        CK  84
      IF (TP.LE.PI2) XXL=CONS*(1.-COS(TP))                              CK  85
      IF (TP.GT.PI2) XXL=CONS*(ATAN(SINH(TP-PI2))+1.)                   CK  86
      IF (NN.EQ.NL) NN=NL1                                              CK  87
      DO 6 J=NN,NL1                                                     CK  88
      J2=NL-J                                                           CK  89
      J1=J2+1                                                           CK  90
      IF (TP.GE.ABS(TL(J2)).AND.TP.LE.ABS(TL(J1))) GO TO 7              CK  91
    6 CONTINUE                                                          CK  92
    7 DELTA=TL(J2)-TL(J1)                                               CK  93
      T1=-TP-TL(J1)                                                     CK  94
      T2=TL(J2)+TP                                                      CK  95
      YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DE CK  96
     1LTA)-T1*DELTA/6.)+(YL(J1)*T2+YL(J2)*T1)/DELTA                     CK  97
      DYL=YPPL(J1)*(DELTA/6.-T2*T2/(2.*DELTA))+YPPL(J2)*(T1*T1/(2.*DELTA CK  98
     1)-DELTA/6.)+(YL(J2)-YL(J1))/DELTA                                 CK  99
      IF (TP.LE.PI2) DELTA=CONS*SIN(TP)                                 CK 100
      IF (TP.GT.PI2) DELTA=CONS/COSH(TP-PI2)                            CK 101
      IF (TP.LE.0.0) DYL=0.1E99                                         CK 102
      IF (TP.GT.0.0) DYL=-DYL/DELTA                                     CK 103
      NN=NL+1-J1                                                        CK 104
      D=SQRT((XXL-XXU)**2+(YYL-YYU)**2)                                 CK 105
      IF (I.EQ.1.AND.D.LE.DEL) GO TO 10                                 CK 106
      IF (D.LE.DEL) GO TO 9                                             CK 107
      COST=(YYU-YYL)/D                                                  CK 108
      SINT=(XXL-XXU)/D                                                  CK 109
      IF (DYU.NE.0.1E99) DU=(COST*DYU-SINT)/(SINT*DYU+COST)             CK 110
      IF (DYU.EQ.0.1E99.AND.SINT.NE.0.0) DU=COST/SINT                   CK 111
      IF (DYU.EQ.0.1E99.AND.SINT.EQ.0.0) DU=0.1E99                      CK 112
      IF (DYL.NE.0.1E99) DL=-(COST*DYL-SINT)/(SINT*DYL+COST)            CK 113
      IF (DYL.EQ.0.1E99.AND.SINT.NE.0.0) DL=-COST/SINT                  CK 114
      IF (DYL.EQ.0.1E99.AND.SINT.EQ.0.0) DL=-0.1E99                     CK 115
      IF (K.EQ.NS) GO TO 8                                              CK 116
      DKL=(DL-DLP)/(XXL-XP)                                             CK 117
      DKU=(DU-DUP)/(XXL-XP)                                             CK 118
      IF (DKU.EQ.DKL) GO TO 8                                           CK 119
      XK=XP+(DLP-DUP)/(DKU-DKL)                                         CK 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      IF (XK.LE.XP+DEL.AND.XK.GE.XXL-DEL) GO TO 11                      CK 121
    8 KSAVE=K                                                           CK 122
      XP=XXL                                                            CK 123
      DUP=DU                                                            CK 124
      DLP=DL                                                            CK 125
    9 CONTINUE                                                          CK 126
      IF (I.GT.1) GO TO 12                                              CK 127
   10 XK=XL(NL)                                                         CK 128
      KSAVE=NS                                                          CK 129
   11 NT=NT+1                                                           CK 130
      LX(NT)=IJ                                                         CK 131
      XLS.NT)=XK                                                        CK 132
      NS=KSAVE                                                          CK 133
   12 CONTINUE                                                          CK 134
C            COMPUTE YLS FOR EACH XLS AND PRINT RESULTS                 CK 135
      WRITE (JWRITE,44) TITLE                                           CK 136
      DO 19 I=1,NT                                                      CK 137
      IJ=LX(I)                                                          CK 138
      DELTA=XLS(I)                                                      CK 139
      IF (DELTA.GT.1.) DELTA=1.                                         CK 140
      IF (DELTA.LE.CONS) GO TO 13                                       CK 141
      DELTA=TAN(DELTA/CONS-1.)                                          CK 142
      TP=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))                           CK 143
      GO TO 14                                                          CK 144
   13 TP=ACOS(1.-DELTA/CONS)                                            CK 145
   14 DO 15 J=1,NL1                                                     CK 146
      J2=NL-J                                                           CK 147
      J1=J2+1                                                           CK 148
      IF (TP.GE.ABS(TL(J2)).AND.TP.LE.ABS(TL(J1))) GO TO 16             CK 149
   15 CONTINUE                                                          CK 150
   16 DELTA=TL(J2)-TL(J1)                                               CK 151
      T1=-TP-TL(J1)                                                     CK 152
      T2=TL(J2)+TP                                                      CK 153
      YYL=YPPL(J1)*(T2**3/(6.*DELTA)-T2*DELTA/6.)+YPPL(J2)*(T1**3/(6.*DE CK 154
     1LTA)-T1*DELTA/6.)+(YL(J1)*T2+YL(J2)*T1)/DELTA                     CK 155
      YLS(I)=YYL                                                        CK 156
      XC(I)=(XU(IJ)+XLS(I))/2.                                          CK 157
      YC(I)=(YU(IJ)+YYL)/2.                                             CK 158
      TK(I)=0.5*SQRT((XU(IJ)-XLS(I))**2+(YU(IJ)-YYL)**2)                CK 159
      IF (YU(IJ).EQ.YYL) TH(I)=0.0                                      CK 160
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

ARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DO 5 I=1,NINT                                                     IT  41
      XU(I)=XINT(I)*CHORD+XNOSE                                         IT  42
      XL(I)=XU(I)                                                       IT  43
      IF (XU(I).GT.XUP) XU(I)=XUP                                       IT  44
      IF (XL(I).GT.XLO) XL(I)=XLO                                       IT  45
      XU(I)=(XU(I)-XNOSE)*RATIO                                         IT  46
      XL(I)=(XL(I)-XNOSE)*RATIO                                         IT  47
      DELTA=XINT(I)                                                     IT  48
      IF (DELTA.LE.CONS) GO TO 3                                        IT  49
      DELTA=TAN(DELTA/CONS-1.)                                          IT  50
      TU=PI2+ALOG(DELTA+SQRT(DELTA*DELTA+1.))                           IT  51
      GO TO 4                                                           IT  52
    3 TU=ACOS(1.-DELTA/CONS)                                            IT  53
    4 TL=-TU                                                            IT  54
      IF (TL.LT.THETA(1)) TL=THETA(1)                                   IT  55
      IF (TU.GT.THETA(NP)) TU=THETA(NP)                                 IT  56
      TLS(I)=TL                                                         IT  57
      CALL COORD (THETA,YPPS,YSMO,NP,TU,YU(I),DYDX,DY2DX,CURV)          IT  58
      YU(I)=YU(I)*CNEW                                                  IT  59
      WRITE (JWRITE,8) I,XU(I),YU(I),DYDX,DY2DX,CURV                    IT  60
    5 CONTINUE                                                          IT  61
      WRITE (JWRITE,9) CNEW                                             IT  62
    C                                                                   IT  63
    C INTERPOLATE LOWER SURFACE COORDINATES                             IT  64
    C                                                                   IT  65
      WRITE (JWRITE,10) TITLE                                           IT  66
      DO 6 I=1,NINT                                                     IT  67
      TL=TLS(I)                                                         IT  68
      CALL COORD (THETA,YPPS,YSMO,NP,TL,YL(I),DYDX,DY2DX,CURV)          IT  69
      YL(I)=YL(I)*CNEW                                                  IT  70
      WRITE (JWRITE,8) I,XL(I),YL(I),DYDX,DY2DX,CURV                    IT  71
    6 CONTINUE                                                          IT  72
    C                                                                   IT  73
    C PUNCH COORDINATES                                                 IT  74
    C                                                                   IT  75
      IF (IPUNCH.NE.6) RETURN                                           IT  76
      WRITE (JWRITE,11) CNEW,TITLE                                      IT  77
      WRITE (1,12) TITLE                                                IT  78
      WRITE (JWRITE,13) NINT                                            IT  79
      XNT=FLOAT(NINT)                                                   IT  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      WRITE (1,14) XNT                                                  IT  81
      WRITE (JWRITE,15) (XU(I),I=1,NINT)                                IT  82
      WRITE (JWRITE,16) (YU(I),I=1,NINT)                                IT  83
      WRITE (1,17) (XU(I),YU(I),I=1,NINT)                               IT  84
      WRITE (JWRITE,18) NINT                                            IT  85
      WRITE (1,14) XNT                                                  IT  86
      WRITE (JWRITE,19) (XL(I),I=1,NINT)                                IT  87
      WRITE (JWRITE,20) (YL(I),I=1,NINT)                                IT  88
      WRITE (1,17) (XL(I),YL(I),I=1,NINT)                               IT  89
C                                                                       IT  90
C     RETURN TO CALLING PROGRAM                                         IT  91
C                                                                       IT  92
      RETURN                                                            IT  93
C                                                                       IT  94
7     FORMAT (1H1,5X,9HTITLE--  ,8A10//26X,42H--UPPER SURFACE INTERPOLAT IT  95
     1ED COORDINATES--//9X,1HI,10X,2HXU,13X,2HYU,11X,5HDY/DX,6X,11HD(DY/ IT  96
     2DX)/DX,6X,9HCURVATURE)                                            IT  97
8     FORMAT (I10,2F15.6,3E15.6)                                        IT  98
9     FORMAT (/10X,7HCHORD =,F10.6)                                     IT  99
10    FORMAT (1H1,5X,9HTITLE--  ,8A10//26X,42H--LOWER SURFACE INTERPOLAT IT 100
     1ED COORDINATES--//9X,1HI,10X,2HYL,13X,2HYL,11X,5HDY/DX,6X,11HD(DY/ IT 101
     2DX)/DX,6X,9HCURVATURE)                                            IT 102
11    FORMAT (1H1,10X,50HTHE FOLLOWING DATA HAVE BEEN PUNCHED FOR A CHOR IT 103
     1D =,F10.6//3X,9HTITLE--  ,8A10)                                   IT 104
12    FORMAT (8A10)                                                     IT 105
13    FORMAT (5X,4HNU =,I4)                                             IT 106
14    FORMAT (F10.2)                                                    IT 107
15    FORMAT (5X,4HXU =,8F10.6/(9X,8F10.6))                             IT 108
16    FORMAT (5X,4HYU =,8F10.6/(9X,8F10.6))                             IT 109
17    FORMAT (2F10.6)                                                   IT 110
18    FORMAT (5X,4HNL =,I4)                                             IT 111
19    FORMAT (5X,4HXL =,8F10.6/(9X,8F10.6))                             IT 112
20    FORMAT (5X,4HYL =,8F10.6/(9X,8F10.6))                             IT 113
      END                                                               IT 114-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

LISTING OF DECK: SINH

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

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.

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      PROGRAM SCALE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
C
C     THIS PROGRAM PRESENTS A TECHNIQUE FOR SCALING THE COORDINATES OF
C     AN AIRFOIL FROM ITS INPUT MAXIMUM THICKNESS RATIO TO A DESIRED
C     OUTPUT MAXIMUM THICKNESS RATIO
C
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB.      1982
C
C*********************************************************************
C*                                                                   *
C*            DESCRIPTION OF INPUT CARDS FOR SCALING PROGRAM         *
C*                                                                   *
C*...................................................................*
C*                                                                   *
C* CARD NUMBER                DESCRIPTION                            *
C*                                                                   *
C*...................................................................*
C*   1                            FORMAT(8A10)                       *
C*            TITLE CARD                                             *
C*...................................................................*
C*   2                            FORMAT(4F10.0)                     *
C*            NT - NUMBER OF INPUT CAMBER, THICKNESS, AND SLOPE      *
C*                 POINTS                                            *
C*            IPLOT - PLOT OPTION                                    *
C*                    0 - NO PLOTS DESIRED                           *
C*                    1 - PLOTS DESIRED                              *
C*            IPUNCH - PUNCH OUTPUT OPTION                           *
C*                     0 - NO PUNCHED OUTPUT DESIRED                 *
C*                     1 - PUNCH COORDINATES OF SCALED AIRFOIL       *
C*            IOP - SLOPE OF CAMBERLINE OPTION                       *
C*                  0 - SLOPES INPUT ON CARD 3                       *
C*                  1 - SLOPES COMPUTED BY PROGRAM                   *
C*...................................................................*
C*   3                            FORMAT(4F10.0)                     *
C*            XC - X/C COORDINATES OF CAMBERLINE                     *
C*            YC - Y/C COORDINATES OF CAMBERLINE                     *
C*            TK - T/C/2 THICKNESS DISTRIBUTION                      *
C*            TH - SLOPE OF CAMBERLINE IN RADIANS                    *
C*                 NOTE -- CARD 3 IS READ NT TIMES                   *
C*...................................................................*
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      IF (EOF(JREAD)) 25,2                                              SC  81
    2 READ (JREAD,27) VAR                                               SC  82
      NT=IFIX(VAR(1))                                                   SC  83
      IF (NT.GT.NTMAX) GO TO 24                                         SC  84
      IPLOT=IFIX(VAR(2))                                                SC  85
      IF (IPLOT.NE.0) IPLOT=1                                           SC  86
      IPUNCH=IFIX(VAR(3))                                               SC  87
      IF (IPUNCH.NE.0) IPUNCH=1                                         SC  88
      IOP=IFIX(VAR(4))                                                  SC  89
      IF (IOP.NE.0) IOP=1                                               SC  90
      WRITE (JWRITE,28) TITLE,NT,IPLOT,IPUNCH,IOP                       SC  91
      READ (JREAD,29) (XC(I),YC(I),TK(I),TH(I),I=1,NT)                  SC  92
      WRITE (JWRITE,30) (XC(I),I=1,NT)                                  SC  93
      WRITE (JWRITE,31) (YC(I),I=1,NT)                                  SC  94
      WRITE (JWRITE,32) (TK(I),I=1,NT)                                  SC  95
      IF (IOP.EQ.0) WRITE (JWRITE,33) (TH(I),I=1,NT)                    SC  96
      READ (JREAD,34) VAR(1)                                            SC  97
      LT=IFIX(VAR(1))                                                   SC  98
      IF (LT.LE.0) GO TO 1                                              SC  99
      IF (LT.GT.10) LT=10                                               SC 100
      READ (JREAD,34) (TKNEW(I),I=1,LT)                                 SC 101
      WRITE (JWRITE,35) LT,(TKNEW(I),I=1,LT)                            SC 102
C                                                                       SC 103
C        INITIALIZE PLOTTING DEVICE                                     SC 104
C                                                                       SC 105
      CALL PSEUDO                                                       SC 106
      CALL LEROY                                                        SC 107
C                                                                       SC 108
C        CHECK FOR INCREASING XC                                        SC 109
C                                                                       SC 110
      DO 3 I=2,NT                                                       SC 111
      IF (XC(I).LE.XC(I-1)) GO TO 4                                     SC 112
    3 CONTINUE                                                          SC 113
      GO TO 5                                                           SC 114
    4 WRITE (JWRITE,36)                                                 SC 115
      GO TO 1                                                           SC 116
C                                                                       SC 117
C        FIND MAXIMUM THICKNESS RATIO OF INPUT AIRFOIL                  SC 118
C                                                                       SC 119
C             COMPUTE THETA EQUIVALENT OF XC                            SC 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DO 13 J=1,N1                                                      SC 161
      J1=J                                                              SC 162
      J2=J+1                                                            SC 163
      IF (WK(I,1).GE.THETA(J).AND.WK(I,1).LE.THETA(J+1)) GO TO 14       SC 164
   13 CONTINUE                                                          SC 165
   14 AA=THETA(J2)-WK(I,1)                                              SC 166
      BB=WK(I,1)-THETA(J1)                                              SC 167
      DELTA=THETA(J2)-THETA(J1)                                         SC 168
   15 WK(I,3)=YPP(J1)*(AA**3/(6.*DELTA)-AA*DELTA/6.)+YPP(J2)*(BB**3/(6.* SC 169
     1DELTA)-BB*DELTA/6.)+(TK(J1)*AA+TK(J2)*BB)/DELTA                   SC 170
   16 CONTINUE                                                          SC 171
C            COMPUTE AND PRINT MAXIMUM THICKNESS RATIO                  SC 172
      IF (KRT.EQ.0) GO TO 23                                            SC 173
      TKMAX=0.0                                                         SC 174
      DO 18 I=1,KRT                                                     SC 175
      IF (WK(I,3).GE.TKMAX) GO TO 17                                    SC 176
      GO TO 18                                                          SC 177
   17 N1=I                                                              SC 178
      TKMAX=WK(I,3)                                                     SC 179
   18 CONTINUE                                                          SC 180
      TKMAX=2.*TKMAX                                                    SC 181
      DELTA=WK(N1,2)*CHORD+XC(1)                                        SC 182
      WRITE (JWRITE,37) TKMAX,DELTA                                     SC 183
      IF (TKMAX.LE.0.0) GO TO 1                                         SC 184
C                                                                       SC 185
C        IF IOP=1, COMPUTE SLOPES OF CAMBERLINE                         SC 186
C                                                                       SC 187
      IF (IOP.NE.1) GO TO 21                                            SC 188
      CALL CUBSPL (XC,YC,YPP,NT,WK)                                     SC 189
      DO 20 I=1,NT                                                      SC 190
      IF (I.EQ.NT) GO TO 19                                             SC 191
      DELTA=XC(I+1)-XC(I)                                               SC 192
      TH(I)=-YPP(I)*DELTA/3.-YPP(I+1)*DELTA/6.+(YC(I+1)-YC(I))/DELTA    SC 193
      GO TO 20                                                          SC 194
   19 DELTA=XC(NT)-XC(NT-1)                                             SC 195
      TH(I)=YPP(NT-1)*DELTA/6.+YPP(NT)*DELTA/3.+(YC(NT)-YC(NT-1))/DELTA SC 196
   20 TH(I)=ATAN(TH(I))                                                 SC 197
C                                                                       SC 198
C        COMPUTE AND PRINT COORDINATES OF INPUT AIRFOIL                 SC 199
C                                                                       SC 200
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      21    CALL SCTK (XC,YC,TK,TH,NT,TITLE,TKMAX,TKMAX,IPUNCH,IPLOT,IERR)     SC 201
            IF (IERR.NE.0) GO TO 1                                             SC 202
      C                                                                        SC 203
      C        COMPUTE AND PRINT COORDINATES OF SCALED AIRFOILS                SC 204
      C                                                                        SC 205
            DO 22 I=1,LT                                                       SC 206
            CALL SCTK (XC,YC,TK,TH,NT,TITLE,TKNEW(I),TKMAX,IPUNCH,IPLOT,IERR)  SC 207
            IF (IERR.NE.0) GO TO 1                                             SC 208
      22    CONTINUE                                                           SC 209
      C                                                                        SC 210
      C        READ NEXT CASE                                                  SC 211
      C                                                                        SC 212
            GO TO 1                                                            SC 213
      C                                                                        SC 214
      C        PRINT ERROR MESSAGE                                             SC 215
      C                                                                        SC 216
      23    WRITE (JWRITE,38)                                                  SC 217
            GO TO 1                                                            SC 218
      24    WRITE (JWRITE,39) NTMAX                                            SC 219
            GO TO 1                                                            SC 220
      C                                                                        SC 221
      C        FINALIZE PLOTTING DEVICE                                        SC 222
      C                                                                        SC 223
      25    CALL CALPLT (0.,0.,999)                                            SC 224
            STOP                                                               SC 225
      C                                                                        SC 226
      26    FORMAT (8A10)                                                      SC 227
      27    FORMAT (4F10.6)                                                    SC 228
      28    FORMAT (1H1,57X,14H--INPUT DATA--//5X,7HTITLE--,2X,8A10//5X,3HNT=, SC 229
           1I3,5X,6HIPLOT=,I3,5X,7HIPUNCH=,I3,5X,4HIOP=,I3)                    SC 230
      29    FORMAT (4F10.6)                                                    SC 231
      30    FORMAT (/4X,4HX/C=,8E15.6/(8X,8E15.6))                             SC 232
      31    FORMAT (/4X,4HY/C=,8E15.6/(8X,8E15.6))                             SC 233
      32    FORMAT (/2X,6HT/C/2=,8E15.6/(8X,8E15.6))                           SC 234
      33    FORMAT (/2X,6HSLOPE=,8E15.6/(8X,8E15.6))                           SC 235
      34    FORMAT (F10.2)                                                     SC 236
      35    FORMAT (/2X,3HLT=,I3,5X,9HNEW T/C =,10F12.6)                       SC 237
      36    FORMAT (/5X,40HXC ARRAY IS NOT MONOTONICALLY INCREASING)           SC 238
      37    FORMAT (/5X,28H(T/C)MAX FOR INPUT AIRFOIL =,F10.6,2X,8HAT X/C =,   SC 239
           1F10.6)                                                             SC 240
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      SUBROUTINE SCTK (XC,YC,TK,TH,NT,TITLE,TKNEW,TKMAX,IPUNCH,IPLOT,IER SK   1
     1R)                                                                 SK   2
C                                                                        SK   3
C     THIS SUBROUTINE SCALES THE COORDINATES OF AN AIRFOIL FROM A BASIC  SK   4
C     MAXIMUM THICKNESS RATIO (TKMAX) TO A NEW MAXIMUM THICKNESS RATIO   SK   5
C     (TKNEW)                                                            SK   6
C                                                                        SK   7
C        CODED BY -- HARRY MORGAN  NASA/LARC/TAD/AAB       1982          SK   8
C                                                                        SK   9
      DIMENSION XC(1), YC(1), TK(1), TH(1), TITLE(8)                     SK  10
C                                                                        SK  11
      COMMON /HLM/ X(220),Y(220),XU(110),YU(110),XL(110),YL(110),XPRT(11 SK  12
     10),YPRT(110),TPRT(110)                                             SK  13
C                                                                        SK  14
      COMMON /BLK1/ PI,PI2,RAD,CONS                                      SK  15
C                                                                        SK  16
      COMMON /INOUT/ JREAD,JWRITE,IPRINT                                 SK  17
C                                                                        SK  18
C        SCALE THICKNESS AND COMPUTE UPPER AND LOWER SURFACE COORDINATES SK  19
C        OF NEW AIRFOIL                                                  SK  20
C                                                                        SK  21
      IERR=0                                                             SK  22
      DELT1=TKNEW/TKMAX                                                  SK  23
      DO 1 I=1,NT                                                        SK  24
      DELT2=COS(TH(I))                                                   SK  25
      DELT4=SIN(TH(I))                                                   SK  26
      XU(I)=XC(I)-TK(I)*DELT4*DELT1                                      SK  27
      YU(I)=YC(I)+TK(I)*DELT2*DELT1                                      SK  28
      XL(I)=XC(I)+TK(I)*DELT4*DELT1                                      SK  29
    1 YL(I)=YC(I)-TK(I)*DELT2*DELT1                                      SK  30
C                                                                        SK  31
C        LOAD SURFACE COORDINATES INTO X AND Y ARRAYS                    SK  32
C                                                                        SK  33
      DO 2 I=1,NT                                                        SK  34
      J=NT+1-I                                                           SK  35
      X(I)=XL(J)                                                         SK  36
    2 Y(I)=YL(J)                                                         SK  37
      N=NT                                                               SK  38
      M=1                                                                SK  39
      IF (XU(1).EQ.XL(1).AND.YU(1).EQ.YL(1)) M=2                         SK  40
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DO 3 I=M,NT                                                       SK  41
      N=N+1                                                             SK  42
      X(N)=XU(I)                                                        SK  43
    3 Y(N)=YU(I)                                                        SK  44
C                                                                       SK  45
C        INTERPOLATE OR EXTRAPOLATE TRAILING EDGE COORDINATES           SK  46
C                                                                       SK  47
      IF (X(1)-X(N)) 4,6,5                                              SK  48
    4 DELT1=X(2)-X(1)                                                   SK  49
      DELT2=X(3)-X(1)                                                   SK  50
      DELT3=Y(2)-Y(1)                                                   SK  51
      DELT4=Y(3)-Y(1)                                                   SK  52
      Y(1)=Y(1)+(X(N)-X(1))*((DELT3*DELT2-DELT4*DELT1)*(X(N)-X(1))+(DELT SK  53
     14*DELT1*DELT1-DELT3*DELT2*DELT2))/(DELT2*DELT1*DELT1-DELT1*DELT2*D SK  54
     2ELT2)                                                             SK  55
      X(1)=X(N)                                                         SK  56
      GO TO 6                                                           SK  57
    5 DELT1=X(N-1)-X(N-2)                                               SK  58
      DELT2=X(N)-X(N-2)                                                 SK  59
      DELT3=Y(N-1)-Y(N-2)                                               SK  60
      DELT4=Y(N)-Y(N-2)                                                 SK  61
      Y(N)=Y(N-2)+(X(1)-X(N-2))*((DELT3*DELT2-DELT4*DELT1)*(X(1)-X(N-2)) SK  62
     1+(DELT4*DELT1*DELT1-DELT3*DELT2*DELT2))/(DELT2*DELT1*DELT1-DELT1*D SK  63
     2ELT2*DELT2)                                                       SK  64
      X(N)=X(1)                                                         SK  65
C                                                                       SK  66
C        COMPUTE LONGEST CHORD                                          SK  67
C                                                                       SK  68
    6 CHORD=0.0                                                         SK  69
      DO 8 I=2,N                                                        SK  70
      DELT=X(1)-X(I)                                                    SK  71
      IF (DELT.GT.CHORD) GO TO 7                                        SK  72
      GO TO 8                                                           SK  73
    7 CHORD=DELT                                                        SK  74
      NOSE=I                                                            SK  75
    8 CONTINUE                                                          SK  76
C                                                                       SK  77
C        ADJUST COORDINATES FOR LONGEST CHORD                           SK  78
C                                                                       SK  79
      DELT=X(NOSE)                                                      SK  80
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
      DO 9 I=1,N                                                        SK  81
      X(I)=(X(I)-DELT)/CHORD                                            SK  82
    9 Y(I)=Y(I)/CHORD                                                   SK  83
C                                                                       SK  84
C        CHECK UPPER AND LOWER SURFACE X VALUES TO DETECT CROSSOVER OF  SK  85
C        PERPENDICULARS TO CAMBERLINE AND TO FIND NOSE POINT            SK  86
C                                                                       SK  87
      DO 10 I=2,NOSE                                                    SK  88
      IF (X(I)-X(I-1)) 10,20,20                                         SK  89
   10 CONTINUE                                                          SK  90
      J=NOSE+1                                                          SK  91
      DO 11 I=J,N                                                       SK  92
      IF (X(I)-X(I-1)) 20,20,11                                         SK  93
   11 CONTINUE                                                          SK  94
C                                                                       SK  95
C        LOAD COORDINATES INTO UPPER AND LOWER SURFACE ARRAYS           SK  96
C                                                                       SK  97
      DO 12 I=1,NOSE                                                    SK  98
      J=NOSE+1-I                                                        SK  99
      XL(I)=X(J)                                                        SK 100
   12 YL(I)=Y(J)                                                        SK 101
      DO 13 I=NOSE,N                                                    SK 102
      J=I+1-NOSE                                                        SK 103
      XU(J)=X(I)                                                        SK 104
   13 YU(J)=Y(I)                                                        SK 105
      NL=NOSE                                                           SK 106
      NU=N-NOSE+1                                                       SK 107
C                                                                       SK 108
C        PRINT SCALED SURFACE COORDINATES                               SK 109
C                                                                       SK 110
      WRITE (JWRITE,21) TITLE,TKNEW                                     SK 111
      J=NU                                                              SK 112
      IF (NL.GT.NU) J=NL                                                SK 113
      DO 14 I=1,J                                                       SK 114
      IF (I.LE.NU.AND.I.LE.NL) WRITE (JWRITE,22) I,XU(I),YU(I),XL(I),YL( SK 115
     1I)                                                                SK 116
      IF (I.LE.NU.AND.I.GT.NL) WRITE (JWRITE,22) I,XU(I),YU(I)          SK 117
      IF (I.GT.NU.AND.I.LE.NL) WRITE (JWRITE,23) I,XL(I),YL(I)          SK 118
   14 CONTINUE                                                          SK 119
C                                                                       SK 120
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

```
 19     CONTINUE                                                          SK 201
        CALL LINE (XPRT,TPRT,NT,1,0,0,0.0)                                SK 202
        CALL NFRAME                                                       SK 203
        RETURN                                                            SK 204
C                                                                         SK 205
C          PRINT ERROR MESSAGE                                            SK 206
C                                                                         SK 207
 20     WRITE (JWRITE,36) TITLE,TKNEW                                     SK 208
        WRITE (JWRITE,37) (I,X(I),Y(I),I=1,N)                             SK 209
        IERR=1                                                            SK 210
        RETURN                                                            SK 211
C                                                                         SK 212
 21     FORMAT (1H1,3X,7HTITLE--,2X,8A10//12X,33HSCALED COORDINATES FOR (T SK 213
       1/C)MAX =,F7.4//24X,5HUPPER,20X,5HLOWER//9X,1HI,10X,3HX/C,7X,3HY/C, SK 214
       212X,3HX/C,7X,3HY/C)                                               SK 215
 22     FORMAT (5X,I5,5X,2F10.6,5X,2F10.6)                                SK 216
 23     FORMAT (5X,I5,30X,2F10.6)                                         SK 217
 24     FORMAT (1H1,3X,7HTITLE--,2X,8A10//12X,49HCAMBER AND THICKNESS DIST SK 218
       1RIBUTIONS FOR (T/C)MAX =,F7.4//34X,6HCAMBER,22X,9HTHICKNESS//9X,1H SK 219
       2I,10X,3HX/C,12X,3HY/C,11X,5HSLOPE,10X,5HT/C/2)                    SK 220
 25     FORMAT (5X,I5,2(5X,F10.6),5X,F10.4,5X,F10.6)                      SK 221
 26     FORMAT (1H1,10X,36HTHE FOLLOWING DATA HAVE BEEN PUNCHED//5X,9HTITL SK 222
       1E--  ,6A10,10H(T/C)MAX =,F10.6)                                   SK 223
 27     FORMAT (6A10,10H(T/C)MAX =,F10.6)                                 SK 224
 28     FORMAT (/5X,4HNU =,I4)                                            SK 225
 29     FORMAT (F10.2)                                                    SK 226
 30     FORMAT (/5X,4HXU =,8F10.6/(9X,8F10.6))                            SK 227
 31     FORMAT (/5X,4HYU =,8F10.6/(9X,8F10.6))                            SK 228
 32     FORMAT (2F10.6)                                                   SK 229
 33     FORMAT (/5X,4HNL =,I4)                                            SK 230
 34     FORMAT (/5X,4HXL =,8F10.6/(9X,8F10.6))                            SK 231
 35     FORMAT (/5X,4HYL =,8F10.6/(9X,8F10.6))                            SK 232
 36     FORMAT (1H1,3X,7HTITLE--,2X,8A10//3X,38HATTEMPT TO SCALE AIRFOIL T SK 233
       10 (T/C)MAX =,F7.4,2X,55HFAILED DUE TO CROSSOVER OF PERPENDICULARS SK 234
       2TO CAMBER  ,E//9X,1HI,9X,3HX/C,13X,3HY/C)                         SK 235
 37     FORMAT (5X,I5,5X,F10.6,5X,F10.6)                                  SK 236
        END                                                               SK 237-
```

ORIGINAL PAGE IS
OF POOR QUALITY

CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY


```
CARD NO.

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

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX C

DESCRIPTION OF INPUT FOR AIRFOIL SMOOTHING PROGRAM AFSMO

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX D

DESCRIPTION OF OUTPUT FOR AIRFOIL SMOOTHING PROGRAM AFSMO

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

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

A summary of the smoothed airfoil properties are printed on page 7. The quantities listed under the THETA, X/C, and Y/C headings are the θ-transformation values and the input $\bar{x}$ and $\bar{y}$ coordinates, respectively. The quantities listed under the YT/C heading are the partially smoothed $\bar{y}$ coordinates generated during the least-squares polynomial smoothing process and under the YSMO/C heading the final smoothed values following the solution of the cubic-spline matrix. The quantity listed under the DELTA heading are the differences between the input and final smoothed $\bar{y}$ coordinates (Y/C - YSMO/C). The quantities listed under the YPS, YPPS, DY/DX, D(DY/DX)/DX and CURVATURE headings are $\bar{y}'$, $\bar{y}''$, dy/dx, d^2y/dx^2, and

ORIGINAL PAGE IS
OF POOR QUALITY

k, respectively. The value of the leading-edge radius is printed next and is simply the reciprocal of the curvature at the nose. The locations of the upper and lower surface inflection points are printed at the bottom of page 7. A summary of the check of the final smoothed $\bar{y}$ and $\bar{y}''$ values is printed on page 8. The check values are obtained by making a call to the least-squares polynomial smoothing subroutine LSQSMO input with the final smoothed $\bar{y}$ coordinates and a uniform weighting factor of 1.0.

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

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

The results of the interpolation process are printed on pages 11 and 12 for the upper and lower surfaces, respectively. The x and

y coordinate values are listed under the XU and YU or XL and YL headings and are based on a chord equal to the value of the input parameter CNEW. The quantities listed under the DY/DX, D(DY/DX)/DX, and CURVATURE headings are dy/dx, d^2y/dx^2, and k, respectively.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX E

DESCRIPTION OF INPUT FOR AIRFOIL SCALING PROGRAM AFSCL

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX F

DESCRIPTION OF OUTPUT FOR AIRFOIL SCALING PROGRAM AFSCL

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES

1. Abbott, Ira H.; and VonDoenhoff, Albert E.: Theory of Wing Sections. Doven Publications, Inc., New York, 1959.

2. Garabedian, P.R.: Transonic Airfoil Codes. NASA CP-2045, Advanced Technology Airfoil Research, Volume I, Part 1, 1979.

3. Stevens, W. A.; Goradia, S. H.; and Braden, J.A.: Mathematical Model for Two-Dimensional Multi-Component Airfoils in Viscous Flow. NASA CR-1843, 1971.

4. Eppler, Richard; and Somers, Dan M.: A Computer Program for the Design and Analysis of Low-Speed Airfoils. NASA TM 80210, 1980.

5. Reinsch, Christian H.: "Smoothing by Spline Functions" Numerische Mathematik, Vol. 10, no. 3, 1967, pp. 177-183.

6. Hildebrand, Francis B.: Methods of Applied Mathematics. Second Edition, Prentice-Hall, Inc., 1965.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE I. - TRANSFORMATION FUNCTION AND FIRST AND SECOND DERIVATIVES

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

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE I. - CONTINUED

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

TABLE I. - CONTINUED

ORIGINAL PAGE IS
OF POOR QUALITY

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

TABLE I. - CONCLUDED

ORIGINAL PAGE IS
OF POOR QUALITY

θ, deg (+/-)	x/c	d(x/c)/dθ	$d^2(x/c)/d\theta^2$
136	.79983	.34535	-.22988
137	.80582	.34134	-.23049
138	.81175	.33732	-.23093
139	.81760	.33328	-.23123
140	.82338	.32925	-.23137
141	.82909	.32521	-.23137
142	.83473	.32117	-.23123
143	.84030	.31714	-.23096
144	.84580	.31311	-.23056
145	.85123	.30909	-.23004
146	.85659	.30508	-.22940
147	.86188	.30108	-.22865
148	.86710	.29710	-.22779
149	.87225	.29313	-.22683
150	.87733	.28918	-.22577
151	.88235	.28525	-.22462
152	.88729	.28134	-.22338
153	.89217	.27746	-.22206
154	.89698	.27359	-.22066
155	.90172	.26975	-.21919
156	.90639	.26594	-.21764
157	.91100	.26216	-.21604
158	.91554	.25840	-.21437
159	.92002	.25467	-.21264
160	.92443	.25098	-.21086
161	.92878	.24731	-.20904
162	.93307	.24368	-.20716
163	.93729	.24008	-.20525
164	.94145	.23652	-.20329
165	.94554	.23299	-.20131
166	.94958	.22949	-.19928
167	.95356	.22603	-.19724
168	.95747	.22261	-.19516
169	.96133	.21922	-.19306
170	.96512	.21587	-.19094
171	.96886	.21255	-.18881
172	.97254	.20928	-.18666
173	.97617	.20604	-.18449
174	.97973	.20284	-.18231
175	.98325	.19967	-.18013
176	.98670	.19655	-.17794
177	.99011	.19346	-.17575
178	.99346	.19041	-.17355
179	.99676	.18740	-.17135
180	1.00000	.18443	-.16915

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE II. - SAMPLE OUTPUT FOR AIRFOIL SMOOTHING PROGRAM

PAGE 1 OUTPUT

```
                                        --INPUT DATA--

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

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

NU =  40

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

YU=   0.            .130000E-01   .204000E-01   .307000E-01   .617000E-01   .496500E-01   .558900E-01   .655100E-01
      .730000E-01   .790000E-01   .840000E-01   .884000E+00   .920000E-01   .977000E-01   .101600E+00   .104000E+00
      .104910E+00   .104450E+00   .102580E+00   .991000E-01   .966800E-01   .937100E-01   .900600E-01   .859900E-01
      .813600E-01   .763400E-01   .709200E-01   .651300E-01   .590700E-01   .529600E-01   .464600E-01   .398800E-01
      .331500E-01   .263900E-01   .196100E-01   .128700E-01   .609000E-02   .200000E-02   .700000E-03  -.700000E-03

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

NL =  40

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

YL=   0.           -.930000E-02  -.138000E-01  -.205000E-01  -.269000E-01  -.319000E-01  -.358000E-01  -.421000E-01
     -.470000E-01  -.510000E-01  -.543000E-01  -.570000E-01  -.593000E-01  -.627000E-01  -.645000E-01  -.652000E-01
     -.649000E-01  -.635000E-01  -.610000E-01  -.570000E-01  -.540000E-01  -.508000E-01  -.469000E-01  -.426000E-01
     -.384000E-01  -.340000E-01  -.294000E-01  -.249000E-01  -.204000E-01  -.160000E-01  -.120000E-01  -.860000E-02
     -.580000E-02  -.360000E-02  -.250000E-02  -.260000E-02  -.400000E-02  -.570000E-02  -.670000E-02  -.800000E-02

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

NINT =  17

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

CNEW =    10.000
```

PAGE 2 OUTPUT

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

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

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

TABLE II. - CONTINUED

ORIGINAL PAGE IS OF POOR QUALITY

PAGE 3 OUTPUT

--INPUT COORDINATES--

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

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

PAGE 4 OUTPUT

--TRANSLATED AND ROTATED COORDINATES--

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

I	XU	YU	XL	YL
1	0.000000	0.000000	0.000000	0.000000
2	.001943	.013009	.002040	-.009291
3	.004911	.020422	.005060	-.013778
4	.012366	.030754	.012589	-.020445
5	.024818	.041828	.025117	-.026791
6	.037294	.049813	.037638	-.031737
7	.049756	.056107	.050155	-.035582
8	.074714	.065836	.075182	-.041773
9	.099682	.073434	.100204	-.046565
10	.124655	.079543	.125221	-.050456
11	.149633	.084652	.150235	-.053647
12	.174661	.089070	.175246	-.056238
13	.199598	.092869	.200256	-.058429
14	.249573	.098787	.250270	-.061612
15	.299555	.102904	.300278	-.063194
16	.349544	.105522	.350280	-.063477
17	.399540	.106649	.400279	-.063159
18	.449541	.106406	.450272	-.061542
19	.499549	.104754	.500261	-.058824
20	.549564	.101492	.550243	-.054607
21	.574574	.099180	.575229	-.051498
22	.599587	.096319	.600215	-.048190
23	.624602	.092778	.625198	-.044181
24	.649620	.088817	.650180	-.039972
25	.674640	.084295	.675161	-.035463
26	.699661	.079384	.700141	-.030955
27	.724685	.074073	.725121	-.026246
28	.749710	.068392	.750101	-.021637
29	.774736	.062441	.775081	-.017029
30	.799762	.056339	.800062	-.012520
31	.824790	.050048	.825044	-.008411
32	.849818	.043577	.850029	-.004902
33	.874848	.036956	.875017	-.001994
34	.899877	.030305	.900007	.000315
35	.924906	.023634	.925002	.001524
36	.949935	.017002	.950002	.001532
37	.974964	.010331	.975008	.000241
38	.989982	.006306	.990015	-.001393
39	.994997	.005028	.995020	-.002372
40	.999994	.003650	1.000025	-.003650

XNOSE = 0.000000 YNOSE = 0.000000 ANGLE = -.249

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE II. - CONTINUED

PAGE 5 OUTPUT

--SUMMARY OF INPUT DATA--

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

I	X/C	Y/C	THETA	YPS	YPPS	W
1	1.000000	-.003650	-180.00			1.00
2	.994995	-.002372	-178.46			1.00
3	.989990	-.001393	-176.97			1.00
4	.974983	.000241	-172.67			1.00
5	.949978	.001532	-166.10			1.00
6	.924979	.001524	-160.12			1.00
7	.899984	.000315	-154.63			1.00
8	.874995	-.001994	-149.54			1.00
9	.850008	-.004902	-144.77			1.00
10	.825023	-.008411	-140.29			1.00
11	.800042	-.012520	-136.03			1.00
12	.775062	-.017028	-131.98			1.00
13	.750082	-.021637	-128.10			1.00
14	.725103	-.026245	-124.37			1.00
15	.700124	-.030954	-120.77			1.00
16	.675144	-.035463	-117.27			1.00
17	.650164	-.039971	-113.86			1.00
18	.625182	-.044180	-110.53			1.00
19	.600200	-.048188	-107.27			1.00
20	.575215	-.051497	-104.06			1.00
21	.550229	-.054606	-100.89			1.00
22	.500248	-.058823	-94.64			1.00
23	.450261	-.061540	-88.45			1.00
24	.400268	-.063158	-82.24			1.00
25	.350271	-.063675	-75.93			1.00
26	.300278	-.063193	-69.44			1.00
27	.250264	-.061610	-62.66			1.00
28	.200251	-.058428	-55.44			1.00
29	.175242	-.056237	-51.59			1.00
30	.150231	-.053646	-47.52			1.00
31	.125217	-.050454	-43.16			1.00
32	.100201	-.046563	-38.42			1.00
33	.075181	-.041772	-33.12			1.00
34	.050154	-.035581	-26.92			1.00
35	.037637	-.031736	-23.27			1.00
36	.025116	-.026790	-18.96			1.00
37	.012589	-.020445	-13.39			1.00
38	.005060	-.013778	-8.48			1.00
39	.002040	-.009291	-5.38			1.00
40	0.000000	0.000000	0.00			1.00
41	.001943	.013008	5.25			1.00
42	.004911	.020421	8.35			1.00
43	.012366	.030753	13.28			1.00
44	.024818	.041827	18.85			1.00
45	.037283	.049811	23.16			1.00
46	.049755	.056106	26.81			1.00
47	.074712	.065834	33.01			1.00
48	.099679	.073432	38.32			1.00
49	.124652	.079541	43.06			1.00
50	.149629	.084650	47.42			1.00
51	.174610	.089068	51.49			1.00
52	.199593	.092667	55.34			1.00
53	.249566	.098784	62.57			1.00
54	.299548	.102901	69.35			1.00
55	.349535	.105519	75.84			1.00
56	.399530	.106646	82.15			1.00
57	.449530	.106404	88.26			1.00
58	.499536	.104751	94.56			1.00
59	.549550	.101489	100.81			1.00
60	.574559	.099178	103.98			1.00
61	.599571	.096317	107.19			1.00
62	.624587	.092776	110.45			1.00
63	.649603	.088814	113.79			1.00
64	.674623	.084293	117.20			1.00
65	.699644	.079382	120.70			1.00
66	.724666	.074071	124.31			1.00
67	.749691	.068390	128.04			1.00
68	.774716	.062439	131.93			1.00
69	.799742	.056338	135.99			1.00
70	.824769	.050047	140.24			1.00
71	.849797	.043576	144.73			1.00
72	.874825	.036955	149.50			1.00
73	.899854	.030304	154.61			1.00
74	.924883	.023633	160.10			1.00
75	.949911	.017002	166.08			1.00
76	.974940	.010331	172.66			1.00
77	.989957	.006306	176.96			1.00
78	.994942	.005028	178.45			1.00
79	.999968	.003650	179.99			1.00

CHORD = 1.000025

TABLE II. - CONTINUED

PAGE 6 OUTPUT

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

                          --SUM OF SQUARES GENERATED DURING SMOOTHING PROCESS--

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

 SMOOTHING PROCESS CONVERGED AFTER  84 ITERATIONS

 SUM OF SQUARES FROM LEAST SQUARES CUR      LINE SMOOTHING =    .7900E-06

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

ORIGINAL PAGE IS OF POOR QUALITY

TABLE II .- CONTINUED

ORIGINAL PAGE IS
OF POOR QUALITY

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

--SMOOTHING OUTPUT SUMMARY--

I	THETA	X/C	Y/C	YT/C	YSMO/C	DELTA	YPS	YPPS	DY/DX	D(DY/DX)/DX	CURVATURE
1	-180.00	1.000000	-.003650	-.003634	-.003634	-.000016	.046662	-.251722	-.253004E+00	-.865835E+01	.788889E+01
2	-178.46	.994995	-.002372	-.002482	-.002471	.000100	.040183	-.231661	-.212594E+00	-.751107E+01	.702919E+01
3	-176.97	.989990	-.001393	-.001518	-.001497	.000104	.034363	-.213279	-.177523E+00	-.652514E+01	.622841E+01
4	-172.67	.974983	.000241	.000472	.000520	-.000279	.020044	-.168921	-.967836E-01	-.435650E+01	.429599E+01
5	-166.10	.949978	.001532	.001736	.001805	-.000273	.003201	-.124731	-.139704E-01	-.242847E+01	.242776E+01
6	-160.12	.924979	.001524	.001428	.001491	.000033	-.008931	-.107946	.356487E-01	-.160033E+01	.159720E+01
7	-154.63	.899984	.000315	.000093	.000136	.000179	-.019363	-.109732	.714075E-01	-.127899E+01	.126927E+01
8	-149.54	.874995	-.001994	-.002056	-.002034	.000040	-.029613	-.120810	.101760E+00	-.115470E+01	.113699E+01
9	-144.77	.850008	-.004982	-.004937	-.004924	.000022	-.040022	-.129549	.129104E+00	-.103885E+01	.101341E+01
10	-140.29	.825023	-.008411	-.008471	-.008452	.000041	-.050038	-.126252	.152515E+00	-.845039E+00	.816389E+00
11	-136.03	.800042	-.012528	-.012534	-.012496	-.000023	-.058748	-.108496	.170175E+00	-.582889E+00	.557689E+00
12	-131.98	.775062	-.017028	-.016960	-.016899	-.000129	-.065483	-.082023	.181220E+00	-.314867E+00	.299970E+00
13	-128.10	.750082	-.021637	-.021582	-.021501	-.000136	-.070152	-.055875	.186367E+00	-.106339E+00	.101030E+00
14	-124.37	.725103	-.026245	-.026270	-.026175	-.000070	-.073120	-.035247	.187297E+00	.263583E-01	.250297E-01
15	-120.77	.700124	-.030954	-.030943	-.030835	-.000119	-.074771	-.017226	.185429E+00	.119704E+00	.113785E+00
16	-117.27	.675144	-.035463	-.035542	-.035417	-.000045	-.075079	.007125	.180959E+00	.234592E+00	.223523E+00
17	-113.86	.650164	-.039971	-.039996	-.039845	-.000126	-.073515	.045501	.172834E+00	.411565E+00	.393790E+00
18	-110.53	.625182	-.044180	-.044195	-.044012	-.000167	-.069500	.092741	.159929E+00	.617598E+00	.594639E+00
19	-107.27	.600200	-.048188	-.048010	-.047801	-.000387	-.063212	.128078	.142845E+00	.748502E+00	.726164E+00
20	-104.06	.575215	-.051497	-.051353	-.051136	-.000361	-.055751	.138268	.124115E+00	.751767E+00	.734725E+00
21	-100.89	.550229	-.054606	-.054222	-.054008	-.000598	-.048130	.137405	.105889E+00	.708829E+00	.697072E+00
22	-94.64	.500248	-.058823	-.058651	-.058462	-.000361	-.033768	.126015	.732075E-01	.605127E+00	.600295E+00
23	-88.45	.450261	-.061548	-.061557	-.061407	-.000133	-.021018	.109849	.454342E-01	.510646E+00	.509069E+00
24	-82.24	.400268	-.063158	-.063179	-.063068	-.000089	-.009874	.095697	.215331E-01	.448747E+00	.448435E+00
25	-75.93	.350271	-.063675	-.063676	-.063596	-.000080	.000110	.085695	-.246031E-03	.425414E+00	.425414E+00
26	-69.44	.300270	-.063193	-.063102	-.063045	-.000147	.009507	.080277	-.219416E-01	.446560E+00	.446238E+00
27	-62.66	.250264	-.061610	-.061402	-.061364	-.000246	.018876	.078109	-.459168E-01	.519921E+00	.518281E+00
28	-55.44	.200251	-.058428	-.058381	-.058366	-.000062	.028654	.076969	-.751952E-01	.665834E+00	.660228E+00
29	-51.59	.175242	-.056237	-.056267	-.056267	.000030	.033772	.075308	-.931369E-01	.776423E+00	.766429E+00
30	-47.52	.150231	-.053646	-.053662	-.053681	.000036	.038969	.071015	-.114183E+00	.916881E+00	.898453E+00
31	-43.16	.125217	-.050456	-.050480	-.050519	.000065	.044162	.065608	-.139501E+00	.112455E+01	.109251E+01
32	-38.42	.100201	-.046543	-.046587	-.046646	.000083	.049352	.059800	-.171611E+00	.147545E+01	.141260E+01
33	-33.12	.075181	-.041772	-.041755	-.041836	.000063	.054498	.051449	-.215540E+00	.211150E+01	.197266E+01
34	-26.92	.050154	-.035581	-.035558	-.035667	.000086	.059347	.038221	-.283233E+00	.353240E+01	.314630E+01
35	-23.27	.037637	-.031736	-.031683	-.031809	.000073	.061555	.031007	-.336718E+00	.521141E+01	.443599E+01
36	-19.96	.025116	-.026790	-.026958	-.027087	.000296	.064298	.042033	-.427560E+00	.101328E+02	.787697E+01
37	-13.39	.012589	-.020445	-.020457	-.020541	.000096	.071415	.104413	-.666148E+00	.351781E+02	.202784E+02
38	-8.48	.005060	-.013778	-.013915	-.013927	.000149	.084022	.189553	-.123115E+01	.161688E+03	.405210E+02
39	-5.38	.002040	-.009291	-.009112	-.009095	-.000206	.095459	.233481	-.219967E+01	.661604E+03	.469282E+02
40	0.00	0.000000	0.000000	.000878	.000878	-.000878	.116289	.210007	.100000E+99	.100000E+99	.342209E+02
41	5.25	.001943	.013008	.012368	.012230	.000778	.129279	.073366	.305143E+01	-.742557E+03	.224266E+02
42	8.35	.004911	.020421	.019521	.019288	.001133	.130542	-.026692	.194141E+01	-.202505E+03	.194448E+02
43	13.28	.012366	.030753	.030559	.030278	.000475	.123997	-.125730	.116686E+01	-.576750E+02	.158925E+02
44	18.85	.024818	.041827	.042046	.041716	.000111	.110840	-.144752	.741341E+00	-.209996E+02	.108865E+02
45	23.16	.037283	.049511	.049968	.049664	.000147	.100957	-.118213	.554764E+00	-.106972E+02	.715283E+01
46	26.81	.049755	.056106	.056174	.055883	.000223	.094163	-.094693	.451085E+00	-.644561E+01	.488438E+01
47	33.01	.074712	.065834	.065850	.065548	.000286	.084785	-.078656	.336277E+00	-.329020E+01	.280177E+01
48	38.32	.099679	.073432	.073382	.073060	.000372	.077550	-.077664	.270285E+00	-.213556E+01	.192123E+01
49	43.06	.124652	.079541	.079549	.079212	.000329	.070990	-.080795	.224676E+00	-.157925E+01	.145843E+01
50	47.42	.149629	.084450	.084720	.084372	.000277	.064710	-.084374	.189914E+00	-.123896E+01	.117483E+01
51	51.49	.174610	.08906A	.089111	.088754	.000314	.058594	-.087787	.161814E+00	-.102513E+01	.986145E+00
52	55.34	.199593	.092867	.092858	.092493	.000373	.052571	-.091364	.138106E+00	-.881396E+00	.856768E+00
53	62.57	.249566	.098784	.098759	.098380	.000404	.040645	-.097742	.989568E-01	-.704434E+00	.694212E+00
54	69.35	.299548	.102901	.102893	.102493	.000408	.028761	-.103114	.664176E-01	-.607705E+00	.603706E+00
55	75.84	.349535	.105519	.105469	.105077	.000442	.016758	-.108843	.373463E-01	-.581603E+00	.580430E+00
56	82.15	.399530	.106646	.106647	.106248	.000396	.004382	-.115939	.955842E-02	-.554542E+00	.554466E+00
57	88.36	.449530	.106404	.106428	.106023	.000381	-.008696	-.125198	-.187988E-01	-.583912E+00	.583603E+00
58	94.56	.499536	.104751	.104741	.104329	.000423	-.022836	-.136308	-.495027E-01	-.649024E+00	.646646E+00
59	100.81	.549550	.101489	.101418	.101003	.000486	-.0383[illegible]	-.148059	-.843443E-01	-.750814E+00	.742873E+00
60	103.98	.574559	.099178	.099065	.098652	.000525	-.046684	-.153318	-.103895E+00	-.814662E+00	.801647E+00
61	107.19	.599571	.096317	.096200	.095793	.000524	-.055350	-.155818	-.125027E+00	-.877314E+00	.857137E+00
62	110.45	.624587	.092776	.092782	.092390	.000386	-.063944	-.145695	-.147073E+00	-.886642E+00	.858633E+00
63	113.79	.649603	.088814	.088809	.088439	.000375	-.071706	-.121296	-.169492E+00	-.825262E+00	.791326E+00
64	117.20	.674623	.084293	.084319	.083976	.000317	-.077998	-.090198	-.187090E+00	-.723432E+00	.686747E+00
65	120.70	.699644	.079382	.079375	.079059	.000323	-.082563	-.059161	-.204633E+00	-.611839E+00	.575326E+00
66	124.31	.724668	.074071	.074047	.073758	.000313	-.085430	-.031832	-.218698E+00	-.508800E+00	.474363E+00
67	128.04	.749691	.068390	.068399	.068134	.000256	-.086777	-.009485	-.230394E+00	-.422272E+00	.390751E+00
68	131.93	.774718	.062439	.062463	.062243	.000196	-.086819	.008247	-.240122E+00	-.351437E+00	.323894E+00
69	135.99	.799742	.056338	.056346	.056129	.000209	-.085739	.022266	-.248216E+00	-.291593E+00	.266581E+00
70	140.24	.824769	.050047	.050024	.049831	.000216	-.083657	.033773	-.254843E+00	-.233796E+00	.212738E+00
71	144.73	.849797	.043576	.043554	.043385	.000190	-.080635	.043305	-.259983E+00	-.171947E+00	.155879E+00
72	149.50	.874825	.036955	.036973	.036831	.000124	-.076723	.050607	-.263529E+00	-.105605E+00	.954877E-01
73	154.61	.899856	.030304	.030323	.030208	.000096	-.071994	.055559	-.265400E+00	-.376401E-01	.339867E-01
74	160.10	.924883	.023633	.023646	.023559	.000074	-.066576	.057374	-.265666E+00	.221672E-01	.201922E-01
75	166.08	.949911	.017002	.016977	.016919	.000083	-.060694	.055334	-.264805E+00	.496267E-01	.448297E-01
76	172.66	.974940	.010331	.010329	.010301	.000030	-.054758	.048085	-.264360E+00	-.205638E-01	.185822E-01
77	176.96	.989957	.006306	.006335	.006323	-.000017	-.051450	.040186	-.265752E+00	-.174626E+00	.157636E+00
78	178.45	.994962	.005028	.004996	.004990	.000038	-.050442	.038839	-.266829E+00	-.257499E+00	.232258E+00
79	179.99	.999968	.003650	.003651	.003651	-.000001	-.049504	.033162	-.268368E+00	-.359699E+00	.324066E+00

LEADING-EDGE RADIUS/C= .029222

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

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

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

TABLE II. - CONTINUED

ORIGINAL PAGE IS
OF POOR QUALITY

PAGE 8 OUTPUT

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

CHECK OF SMOOTHED COORDINATES DF= .000100

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

TABLE II. - CONTINUED

ORIGINAL PAGE IS
OF POOR QUALITY

PAGE 9 OUTPUT

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

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

IOP =   0

NU =  40

DX =  0.000000   .001943   .004911   .012366   .024818   .037254   .049756   .074714
      .099682   .124655   .149633   .174614   .199596   .249573   .299555   .349544
      .399540   .449541   .499549   .549564   .574574   .599587   .624602   .649620
      .674640   .699661   .724685   .749710   .774736   .799762   .824790   .849818
      .874848   .899877   .924906   .949935   .974964   .989982   .994988   .999994

DY =   .000878   .012230   .019288   .030279   .041717   .049666   .055864   .065550
      .073062   .079214   .084374   .088757   .092496   .095333   .102496   .105080
      .106251   .106026   .104331   .101005   .098655   .095795   .092392   .088442
      .083978   .079061   .073760   .068136   .062245   .056131   .049832   .043387
      .036832   .030209   .023560   .016920   .010301   .006323   .004991   .003651

NL =  40

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

DY =   .000878  -.009085  -.013927  -.020541  -.027087  -.031809  -.035668  -.041837
     -.046647  -.050521  -.053683  -.056268  -.058367  -.061366  -.063047  -.063597
     -.063070  -.061409  -.058463  -.054809  -.051138  -.047802  -.044014  -.039846
     -.035418  -.030836  -.026176  -.021502  -.016900  -.012497  -.008452  -.004924
     -.002034   .000136   .001491   .001605   .000520  -.001497  -.002472  -.003634
```

PAGE 10 OUTPUT

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

                          --THICKNESS AND CAMBER DISTRIBUTION--

  I     XU/C      YU/C      XL/C      YL/C      X/C       Y/C       T/C/2     SLOPE       ERROR

  1   .999968   .003651   .998234  -.003201   .999101   .000225   .003534  -14.2073     .000005
  2   .994962   .004990   .993303  -.002122   .994133   .001434   .003652  -13.1308     .000026
  3   .989957   .006323   .988331  -.001212   .989144   .002556   .003854  -12.1802     .000023
  4   .974940   .010301   .973251   .000681   .974095   .005491   .004883   -9.9589     .000039
  5   .949911   .019019   .947878   .001829   .948894   .009374   .007613   -7.6733     .000028
  6   .924883   .023559   .922434   .001395   .923658   .012477   .011149   -6.3035     .000019
  7   .899854   .030208   .897053  -.000078   .898453   .015065   .015208   -5.2838     .000014
  8   .874825   .036831   .871828  -.002362   .873326   .017234   .019653   -4.3739     .000012
  9   .849797   .043385   .846803  -.005343   .848300   .019021   .024410   -3.5160     .000001
 10   .824769   .049831   .821955  -.008924   .823362   .020454   .029411   -2.7419     .000005
 11   .799742   .056129   .797212  -.012990   .798477   .021574   .034578   -2.0968     .000006
 12   .774716   .062243   .772501  -.017364   .773609   .022439   .039819   -1.5938     .000000
 13   .749691   .068134   .747800  -.021927   .748745   .023104   .045040   -1.2025     .000005
 14   .724666   .073758   .723151  -.026541   .723909   .023669   .050155    -.8656     .000003
 15   .699644   .079059   .698614  -.031115   .699129   .023972   .055090    -.5352     .000004
 16   .674623   .083976   .674209  -.035586   .674416   .024195   .059781    -.1983     .000003
 17   .649603   .088439   .649866  -.039897   .649735   .024271   .064168     .1174     .000003
 18   .624587   .092390   .625454  -.043969   .625020   .024210   .068181     .3644     .000003
 19   .599571   .095793   .600862  -.047706   .600217   .024043   .071753     .5153     .000001
 20   .574559   .098652   .576102  -.051026   .575331   .023813   .074843     .5906     .000001
 21   .549550   .101003   .551259  -.053898   .550405   .023552   .077455     .6323     .000001
 22   .499536   .104329   .501520  -.058368   .500528   .022980   .081355     .6985     .000001
 23   .449530   .106023   .451823  -.061336   .450677   .022344   .083687     .7851     .000000
 24   .399530   .106248   .402236  -.063025   .400883   .021611   .084647     .9159     .000000
 25   .349535   .105077   .352752  -.063595   .351144   .020741   .084351    1.0926     .000000
 26   .299548   .102493   .303336  -.063111   .301442   .019691   .082823    1.3106     .000000
 27   .249566   .098380   .253936  -.061529   .251751   .018426   .079985    1.5654     .000000
 28   .199593   .092493   .204506  -.058680   .202049   .016907   .075626    1.8615     .000000
 29   .174610   .088754   .17[illegible]774  -.056681   .177192   .016037   .072763    2.0338     .000000
 30   .149629   .084372   .155057  -.054222   .152343   .015075   .069350    2.2425     .000000
 31   .124652   .079212   .130394  -.051227   .127523   .013992   .065282    2.5205     .000000
 32   .099679   .073060   .105827  -.047589   .102753   .012736   .060403    2.9170     .000001
 33   .074712   .065548   .081471  -.043151   .078092   .011198   .054454    3.5578     .000002
 34   .049755   .055883   .057757  -.037724   .053756   .009079   .046974    4.8859     .000002
 35   .037283   .049664   .046504  -.034609   .041894   .007528   .042388    6.2449     .000001
 36   .024818   .041716   .036053  -.031268   .030435   .005224   .036922    8.7511     .000000
 37   .012366   .030278   .026086  -.027496   .019226   .001391   .029691   13.3584     .000003
 38   .004911   .019288   .018942  -.024216   .011927  -.002464   .022855   17.8758     .000003
 39   .001943   .012230   .014553  -.021786   .008248  -.004778   .018139   20.3388     .000002
 40  0.000000   .000878   .007551  -.016593   .003776  -.007858   .009517   23.3746     .000005
 41   .002056  -.009120   .002056  -.009120   .002056  -.009120  0.000000   36.7380     .000000
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE II. - CONCLUDED

PAGE 11 OUTPUT

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

--UPPER SURFACE INTERPOLATED COORDINATES--

I	XU	YU	DY/DX	D(DY/DX)/DX	CURVATURE
1	0.000000	.008778	.100000E+99	.100000E+99	.342209E+02
2	.010000	.089074	.417266E+01	-.196293E+04	.248476E+02
3	.020000	.124016	.301025E+01	-.712541E+03	.223259E+02
4	.050000	.194594	.192363E+01	-.197447E+03	.193756E+02
5	.100000	.273409	.132338E+01	-.762275E+02	.167032E+02
6	.500000	.559929	.449512E+00	-.639751E+01	.485424E+01
7	.800000	.672815	.319754E+00	-.296890E+01	.256555E+01
8	1.000000	.731472	.269601E+00	-.212599E+01	.191361E+01
9	2.000000	.925495	.137748E+00	-.879432E+00	.854982E+00
10	3.000000	1.025230	.661428E-01	-.607111E+00	.603149E+00
11	4.000000	1.062524	.929760E-02	-.554703E+00	.554631E+00
12	5.000000	1.043058	-.498038E-01	-.649796E+00	.647386E+00
13	6.000000	.957391	-.125403E+00	-.877407E+00	.857109E+00
14	7.000000	.789865	-.204851E+00	-.610511E+00	.574003E+00
15	8.000000	.560652	-.248291E+00	-.291119E+00	.266134E+00
16	9.000000	.301692	-.265405E+00	-.373856E-01	.337568E-01
17	9.999600	.036510	-.268368E+00	-.359699E+00	.324066E+00

CHORD = 10.000000

PAGE 12 OUTPUT

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

--LOWER SURFACE INTERPOLATED COORDINATES--

I	XL	YL	DY/DX	D(DY/DX)/DX	CURVATURE
1	0.000000	.008778	.100000E+99	.100000E+99	.342209E+02
2	.010000	-.063026	-.335261E+01	.191941E+04	.448226E+02
3	.020000	-.089958	-.222619E+01	.631620E+03	.468941E+02
4	.050000	-.138530	-.124092E+01	.164756E+03	.407024E+02
5	.100000	-.186816	-.777581E+00	.526254E+02	.258900E+02
6	.500000	-.356231	-.283778E+00	.354689E+01	.315786E+01
7	.800000	-.428505	-.205762E+00	.195000E+01	.183240E+01
8	1.000000	-.466116	-.171908E+00	.147902E+01	.141580E+01
9	2.000000	-.583469	-.753524E-01	.666760E+00	.661121E+00
10	3.000000	-.630395	-.220622E-01	.446844E+00	.446518E+00
11	4.000000	-.630743	.214127E-01	.448574E+00	.448266E+00
12	5.000000	-.584799	.730575E-01	.604586E+00	.599778E+00
13	6.000000	-.478298	.142695E+00	.748504E+00	.726211E+00
14	7.000000	-.308580	.185414E+00	.120328E+00	.114379E+00
15	8.000000	-.125036	.170199E+00	-.581575E+00	.557190E+00
16	9.000000	.001376	.713875E-01	-.127916E+01	.126944E+01
17	10.000000	-.036343	-.253004E+00	-.865835E+01	.788889E+01

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE III. - SAMPLE OUTPUT FOR AIRFOIL SCALING PROGRAM

PAGE 1 OUTPUT

```
                                                    --INPUT DATA--

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

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

  X/C=     .182000E-02   .338100E-02   .791900E-02   .114860E-01   .192200E-01   .306300E-01   .420950E-01   .538920E-01
           .781350E-01   .102793E+00   .127608E+00   .152464E+00   .177313E+00   .202154E+00   .251836E+00   .301530E+00
           .351251E+00   .401010E+00   .450814E+00   .500657E+00   .550503E+00   .575409E+00   .600279E+00   .625084E+00
           .649822E+00   .674533E+00   .699266E+00   .724045E+00   .748863E+00   .773699E+00   .798548E+00   .823428E+00
           .848372E+00   .873407E+00   .898533E+00   .923728E+00   .948946E+00   .974128E+00   .989166E+00   .994153E+00
           .999118E+00

  Y/C=    -.843100E-02  -.739600E-02  -.455900E-02  -.239900E-02   .126700E-02   .502700E-02   .734200E-02   .889900E-02
           .109520E-01   .123760E-01   .135190E-01   .145050E-01   .153780E-01   .161540E-01   .174660E-01   .185130E-01
           .193480E-01   .200120E-01   .205450E-01   .209800E-01   .213350E-01   .214760E-01   .215800E-01   .216200E-01
           .215600E-01   .213760E-01   .210550E-01   .205940E-01   .199850E-01   .192040E-01   .182080E-01   .169490E-01
           .153830E-01   .134750E-01   .111980E-01   .850900E-02   .529300E-02   .127700E-02  -.174200E-02  -.288900E-02
          -.412100E-02

T/C/2=    0.             .902500E-02   .178730E-01   .226660E-01   .296660E-01   .370210E-01   .424880E-01   .470340E-01
           .544440E-01   .603680E-01   .652520E-01   .693270E-01   .727400E-01   .756000E-01   .799640E-01   .828210E-01
           .843640E-01   .846590E-01   .836790E-01   .813180E-01   .774050E-01   .748020E-01   .717340E-01   .681920E-01
           .642030E-01   .598240E-01   .551240E-01   .501760E-01   .450520E-01   .398330E-01   .346060E-01   .294590E-01
           .244720E-01   .197160E-01   .152540E-01   .111720E-01   .761800E-02   .488500E-02   .385800E-02   .365500E-02
           .353500E-02

SLOPE=    .592326E+00   .383967E+00   .337573E+00   .29[illegible]424E+00   .228491E+00   .152664E+00   .108353E+00   .828360E-01
           .576090E-01   .462850E-01   .399860E-01   .[illegible]55510E-01   .318050E-01   .284960E-01   .229570E-01   .184780E-01
           .148330E-01   .119300E-01   .972200E-02   .807400E-02   .650200E-02   .547000E-02   .388800E-02   .123000E-02
          -.277500E-02  -.780800E-02  -.133180E-01  -.190370E-01  -.252480E-01  -.326600E-01  -.419630E-01  -.533870E-01
          -.665680E-01  -.809090E-01  -.962880E-01  -.114138E+00  -.138752E+00  -.179363E+00  -.217802E+00  -.233989E+00
          -.252162E+00

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

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

TABLE III.- CONTINUED

ORIGINAL PAGE IS
OF POOR QUALITY

PAGE 2 OUTPUT

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

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE III. - CONCLUDED

PAGE 3 OUTPUT

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

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

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

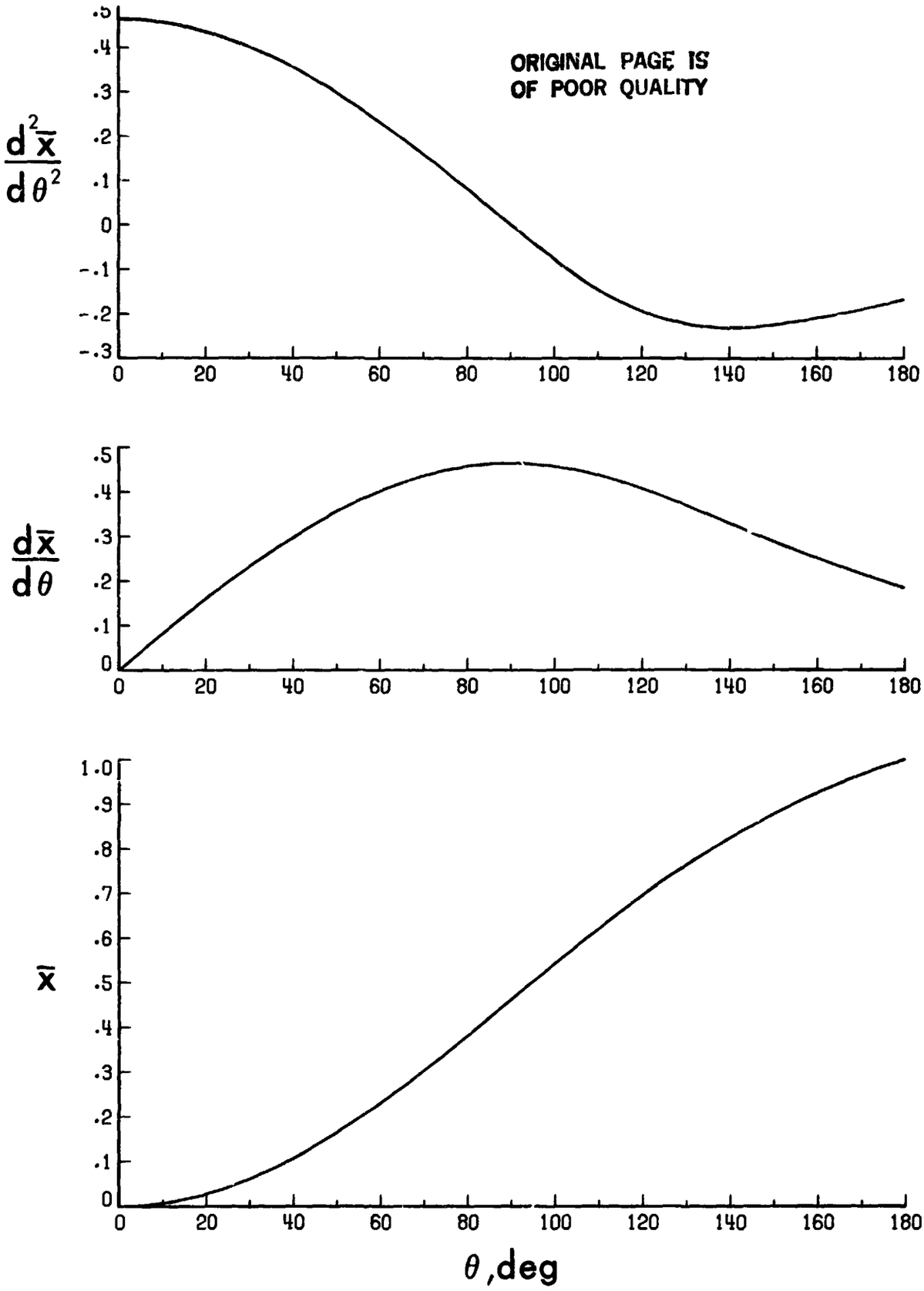


Figure 1. - Properties of θ-transformation function.

ORIGINAL PAGE IS
OF POOR QUALITY

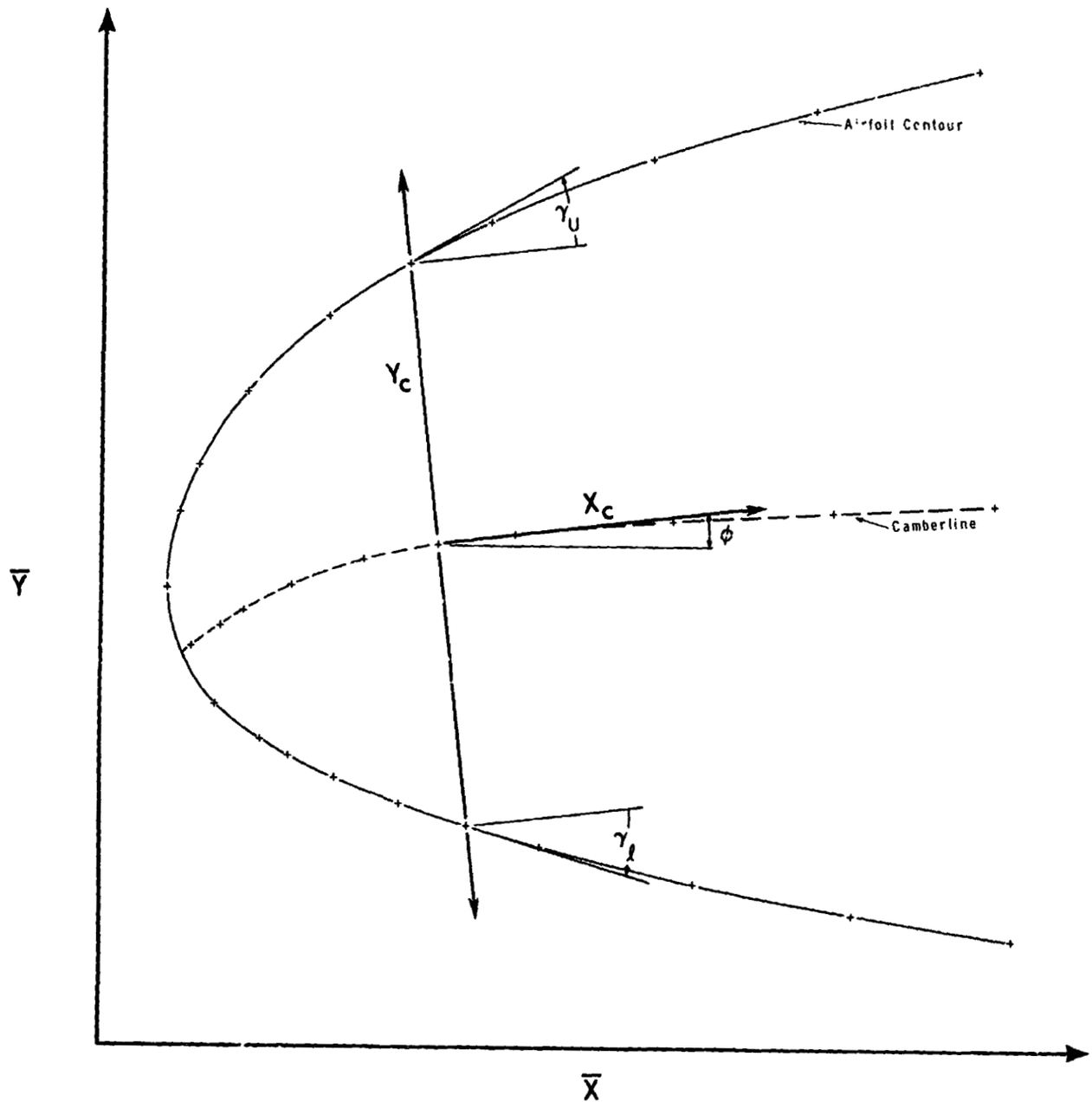


Figure 2. - Camberline axis system.

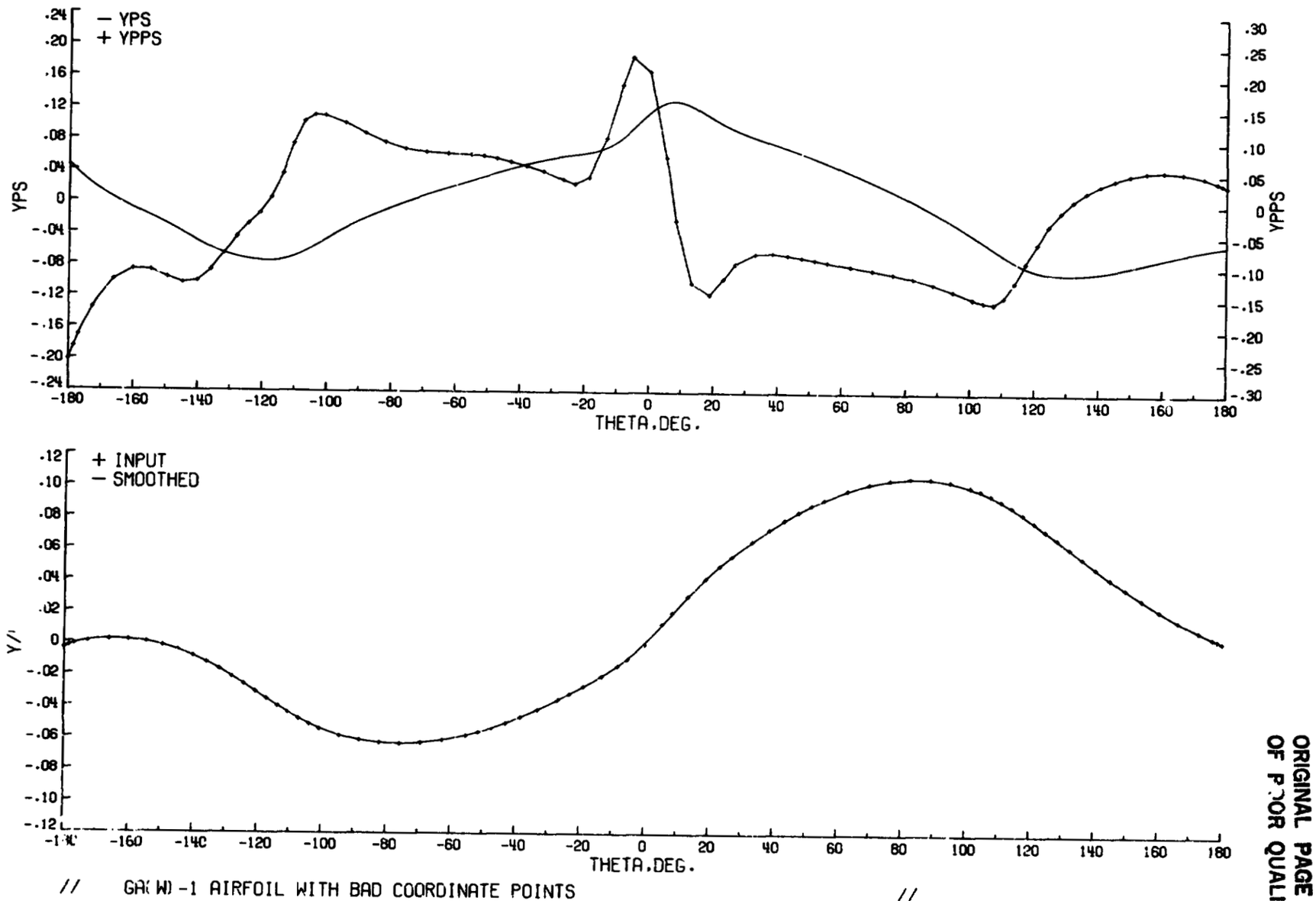


Figure 3.- Sample plot for airfoil smoothing program plotting option 1.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

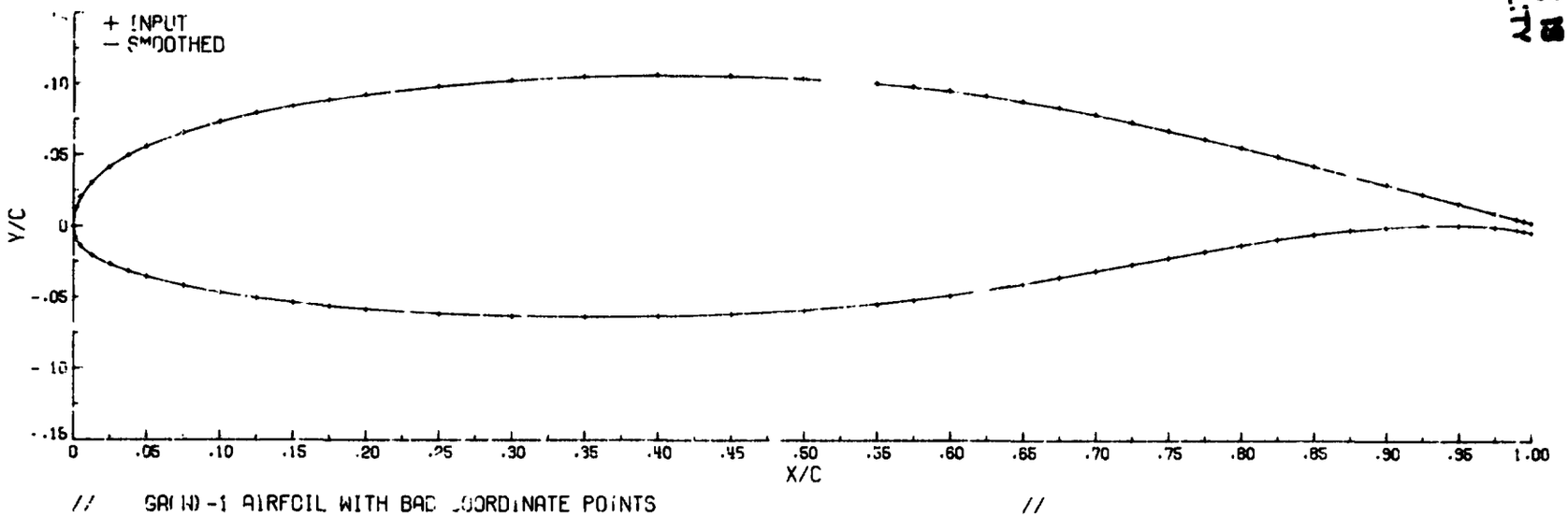
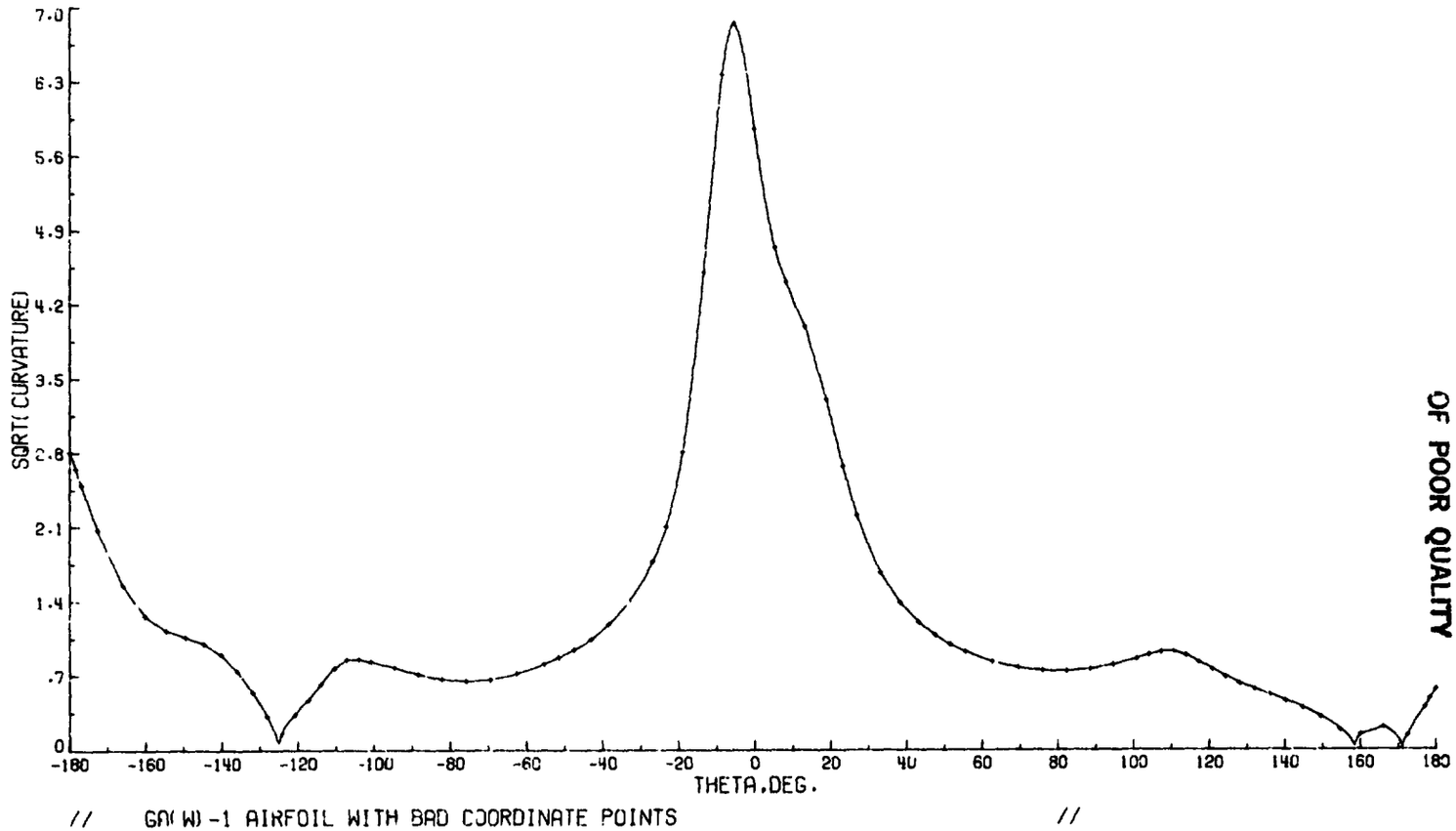


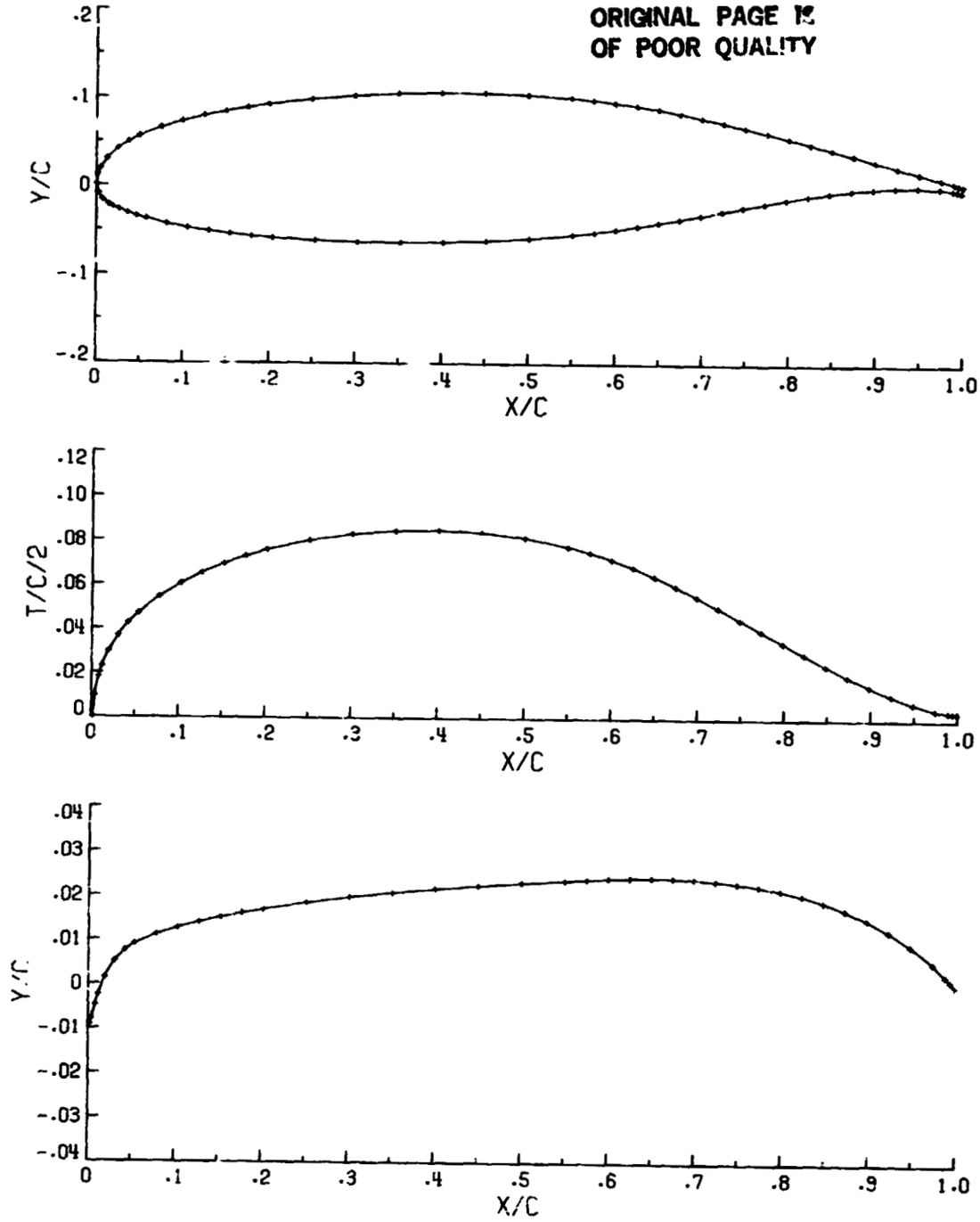
Figure 4.- Sample plot for airfoil smoothing program plotting option 2.



ORIGINAL PAGE IS
OF POOR QUALITY

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

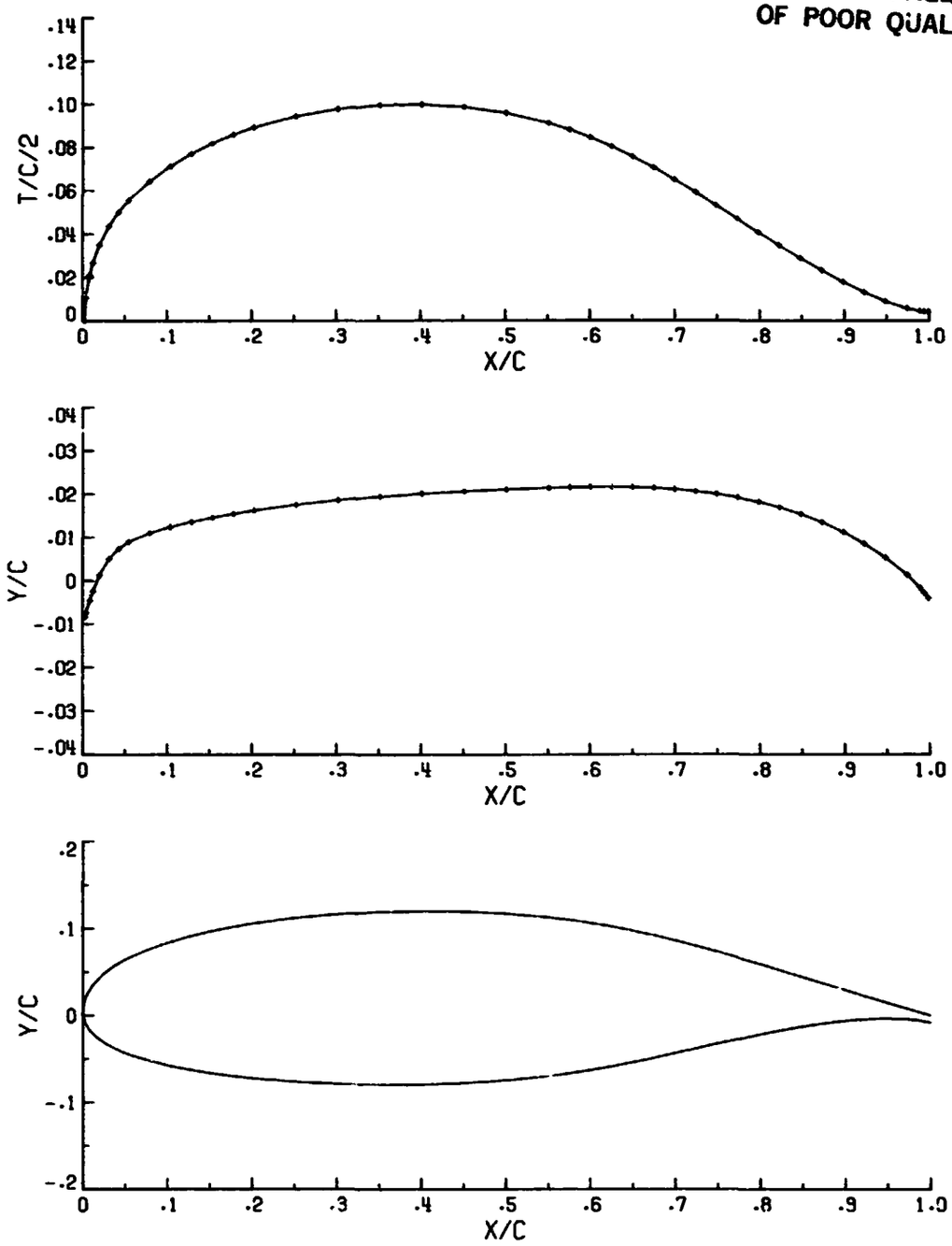
ORIGINAL PAGE IS
OF POOR QUALITY



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

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

ORIGINAL PAGE IS
OF POOR QUALITY

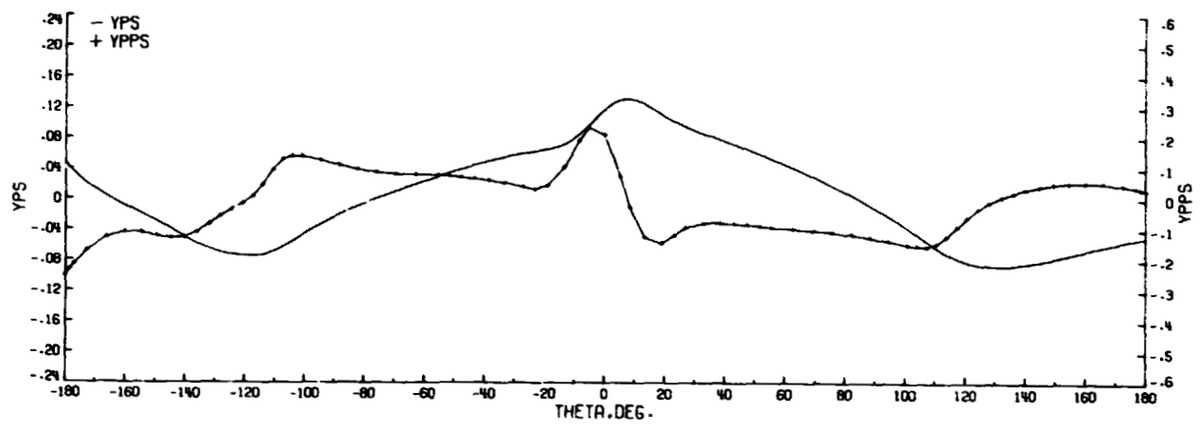


* GA(W)-1 SMOOTHED *

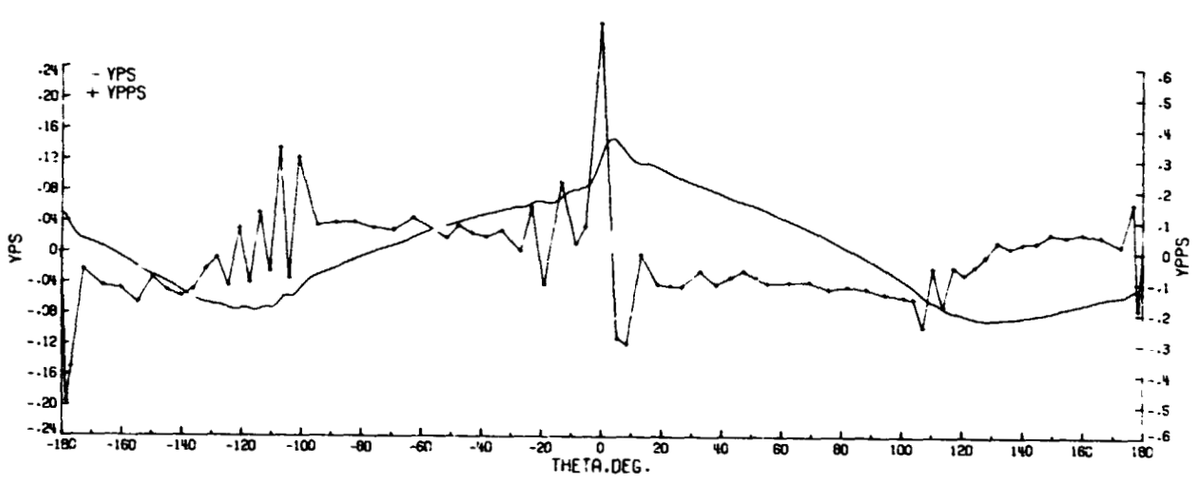
PLOT OF AIRFOIL GENERATED BY SCALING PROGRAM (T/C)MAX = .200

Figure 7.- Sample plot for airfoil scaling program

ORIGINAL PAGE IS
OF POOR QUALITY



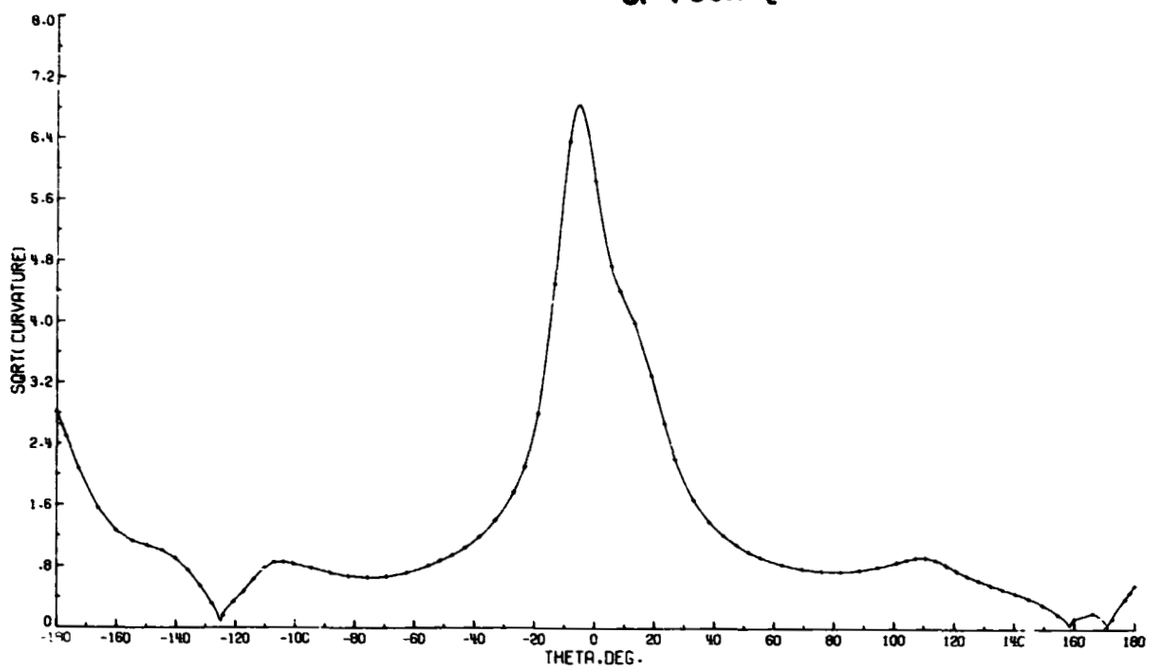
(b) Smoothed



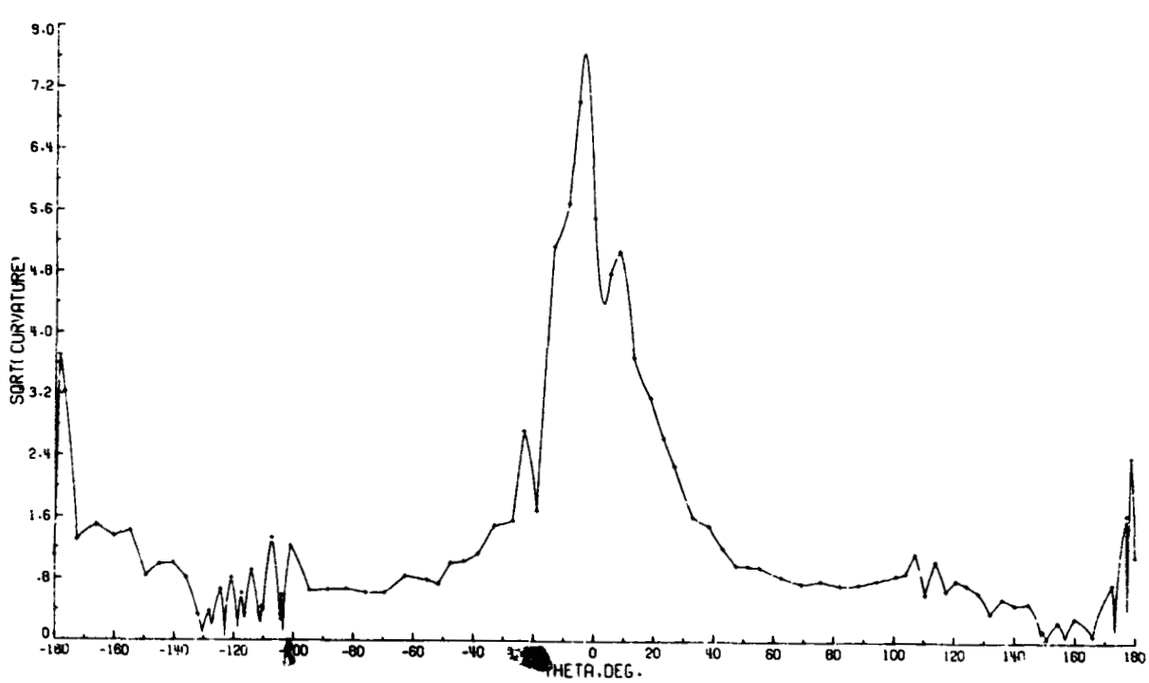
(a) Unsmoothed

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

ORIGINAL PAGE IS
OF POOR QUALITY



(b) Smoothed



(a) Unsmoothed

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

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-84666		

4. Title and Subtitle	5. Report Date
Computer Programs for Smoothing and Scaling Airfoil Coordinates	July 1983
	6. Performing Organization Code: 505-31-23-06
7. Author(s): Harry L. Morgan, Jr.	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address: NASA Langley Research Center, Hampton, VA 23665	11. Contract or Grant No.
	13. Type of Report and Period Covered: Technical Memorandum
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

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

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

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Airfoil Coordinate smoothing Computer code Scaling Camber and thickness distribution	[illegible]

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	186	