DEVELOPMENT OF THEORETICAL METHOD FOR TWO-DIMENSIONAL MULTI-ELEMENT AIRFOIL ANALYSIS AND DESIGN

Part II: Leading-Edge Slat Design Method

O. Wayne McGregor
Jack W. McWhirter

FOREWORD

This final report summarizes the work accomplished by the Fort Worth operation of the Convair Aerospace Division of General Dynamics Corporation, Fort Worth, Texas, under USAF Contract F33615-71-C-1597, Project 1366. The work was conducted under the direction of the Air Force Program Manager, Mr. Russell F. Osborn, AFFDL/FXM.

The results of this development are documented in two parts which constitute the Final Technical Report under the contract. In Part I, a method for the viscous flow analysis of multi-element airfoil systems is presented. In Part II, a semi-inverse method for the design of leading-edge slats is presented.

Mr. Ishwar C. Bhateley served as the General Dynamics Program Manager for this study and developed the viscous flow method. Dr. O. Wayne McGregor developed the slat design method. The computer programming task for both methods was performed by Jack W. McWhirter. Dr. R. G. Bradley served as a special consultant and counselor throughout the study.

The research and technical development for this report was accomplished between 24 May 1971 and 12 June 1972. This report was released by the authors in July 1972.

This technical report has been reviewed and is approved.

Phillip P. Antonatos
PHILLIP P. ANTONATOS
Chief, Flight Mechanics Division
AF Flight Dynamics Laboratory

ABSTRACT

A method has been developed for the design of leading-edge slats that produce a specified pressure distribution on the main airfoil. The method of distributed singularities is applied in a unique manner. The airfoil is represented in the conventional manner by a vortex sheet having the same shape as the airfoil. The slat is represented by a vortex sheet and a source line. The source line provides the slat thickness; the vortex sheet provides the camber. A closed slat shape is guaranteed by requiring that the net mass added to the system be zero and that the stream function at the slat leading-edge stagnation point have the same value as at the trailing edge. It was found that valid solutions are possible only when the source line at least approximates a streamline generated by the airfoil without the slat. The slat shape is computed by locating the body streamline of the slat. A constrained least-square analysis provides this definition. Several sample designs are discussed. Detailed instructions for application of the method are provided. Details regarding the associated computer program are included.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

NOMENCLATURE

A, B	Velocity influence coefficients (Eq. 1)
C_p	Pressure coefficient
D	Stream function influence coefficient (Eq. 22)
d	Distance between two points
G	Strength of vortex sheet (Eq. 1)
M	Strength of source sheet (Eq. 10)
Q	Velocity discontinuity on vortex sheet (Eq. 17)
U_∞	Freestream velocity
u, v	Horizontal and vertical velocity components
x, y	Horizontal and vertical coordinates
α	Geometric angle of attack
ψ	Local stream function (Eq. 22)
θ	Local slope of geometry

1. INTRODUCTION

The application of leading-edge slats on modern high-performance aircraft is now commonplace. The vast majority of these slats are retractable, which means that the low-drag qualities of the cruise airfoil are maintained for high-speed flight, but the high-lift capabilities provided by leading-edge slats are available during takeoff and landing.

Until recently the number of design options available to the slat designer was very restricted. The slat was basically a portion of the cruise airfoil, with the upper surface being the nose section of the airfoil. The designer would design the lower-surface shape of the slat and the upper-surface shape of the airfoil nose. He also had some freedom in the choice of the position of the extended slat. However, as performance requirements increased, designs have necessarily become more sophisticated. A rotating-nose slat was developed for the Convair F-111. A variable-camber slat was developed for the Boeing 747. With the advent of suitable deformable materials, the designer can now obtain almost any leading-edge slat shape he needs. The subject of this present study is a new design tool, which will allow the designer to take full advantage of these new capabilities.

Methods for the design of leading-edge devices can be divided into two main classes. The first, the older and most widely used, is the "cut-and-try" method. The second, just in its infancy but showing great promise, is the inverse method.

In the cut-and-try method, brute force is employed to select a leading-edge device shape and orientation. Slat shape and size are selected, and, although some guidelines are available, selection is made primarily on the basis of the intuition and experience of the designer. The optimum slat location is then determined by measuring its performance at a great variety of locations relative to the airfoil. Slat performance is usually measured by a combination of wind tunnel tests and analytical evaluations. The position that provides the most desirable performance is then selected as the optimum position.

Although this procedure has been of value in the past, it has several shortcomings. For example, no assurance can be made that the desired performance will be attained with the selected slat shape simply by varying its position. If the chosen shape is found to provide inadequate performance, a new shape must be selected, and one must again repeat the expensive process of determining the optimum orientation. It is not difficult to systematize the search for the optimum (largest $C_{L_{max}}$) slat orientation; however, selection of the slat shape is not so straightforward. Guidelines for shape selection are generally either statements of the obvious or premises difficult to translate into physical shapes. As a result, slat design, by this method, tends to be more of an art than a science.

The alternate to the cut-and-try method is the inverse method. In this method, the designer specifies conditions on the flow field, such as surface pressures and freestream conditions, and then computes the required shapes. Techniques of this type have been available for single-element airfoil design for several years (Ref. 1). Such procedures are especially useful in designing specialized airfoil shapes such as "laminar" airfoils (Ref. 2) or maximum-lift airfoils (Ref. 3). Since the development of this technique has been directed primarily toward single-element airfoil design, the extension to multi-element airfoil design has been neglected. Even if such an extension were made, the utility of the procedure for the design of high-lift devices is questionable. If a Lighthill-type inverse procedure (Ref. 4) is utilized, the entire airfoil geometry is designed, not just the leading- and trailing-edge regions. The result is that the main airfoil could have a substantially different shape from the cruise airfoil shape. Generally, this would not be an acceptable design solution.

A recent addition to the list of inverse methods, made by Chen (Ref. 5), utilizes a nonlinear approach to design maximum-lift multiple-element airfoils. This is a full inverse procedure in that both the airfoil and the high-lift devices are designed. For each element the method assumes an optimum pressure distribution, consisting of a uniform stagnation pressure on the lower surface, a uniform minimum pressure on the upper surface immediately downstream of the front stagnation point, followed by a Stratford zero skin-friction pressure rise. The contours of the various elements are adjusted in a systematic manner to produce the desired pressure distributions. The method presents a novel approach to the airfoil design problem, but it exhibits the same difficulty as the Lighthill method: the entire airfoil geometry is designed, not just the slat.

A compromise approach has been formulated by O'Pray (Ref. 6). By permitting the airfoil shape to be specified, he computes the shape of the leading-edge device through a semi-inverse procedure. Through a series of conformal transformations, the airfoil is mapped into an ellipse and then into the real axis of a complex half-plane with the airfoil nose at the origin. The slat is represented by a set of distributed singularities along a straight line in this plane. This depictment of the slat requires the designer to specify both the location and the approximate chord of the slat.

O'Pray defines the slat shape by requiring the slat to produce a specified pressure distribution on the main airfoil nose. He uses a seventh-order least-square description of the pressures to describe the distribution on the average. Using the coefficients of the least-square fit, he defines the slat. The solution at this point does not account for the circulation generated by the slat. In order to satisfy the Kutta condition on the ellipse, an iteration must be made. The net slat circulation is added to the circulation about the ellipse, and then a new slat is computed. This is repeated until convergence is attained. The actual slat-shape definition is accomplished by tracing streamlines from the leading edge to the trailing edge through the use of linearized flow theory.

After thorough examination of the O'Pray procedure, it is felt that its intricate mathematics and its approximations are not necessary. An alternate, more generalized method has been developed that applies the method of distributed singularities (Ref. 7 or 8, for example) to represent the airfoil and the leading-edge slat. The resulting method is very simple, direct, and versatile. Also, the entire analysis is carried out in the physical plane. This is especially useful in interpreting the effects of the design parameters. The method is described in detail in Section 2. (Derivation of the pertinent potential-flow equations is given in Appendix I.) Some representative slat designs are presented in Section 3. Step-by-step instructions on the utilization of the design technique are provided in Section 4. A detailed description of the computer program developed during the study is given in Appendix II.

Program utilization and program listing are given in Appendixes III and IV, respectively.

2. DESCRIPTION OF METHOD

The method of distributed singularities provides an efficient tool for the solution of complex potential-flow problems. The method is based on a fundamental characteristic of incompressible, inviscid flows. The linear nature of the governing partial differential equation (Laplace's equation) permits simple solutions to be superimposed to form more complex solutions. In this problem, the solutions for simple source singularities and for simple vortex singularities are superimposed. A description of how these singularities are superimposed and how the results are interpreted is the subject of this section.

In this design problem, two types of bodies are considered: those whose shapes are specified and those whose shapes are to be computed. The technique is general and can be extended to an arbitrary number of bodies, provided that at least one body shape is specified. For example, the shapes of the slat and the three flaps of a triple-slotted flap system could conceivably be computed with this technique. Only the two-body problem is considered in this study. The specified body is the main airfoil; the computed body is the slat.

The main airfoil is defined by a vortex sheet in the shape of an inscribed polygon, exactly as described by Bhateley in the companion volume to this present study (Ref. 7). Note that the capability of designing leading-edge slats for airfoil-flap combinations and blunt trailing-edge airfoils, as well as considering ground effects, could be included in the method in the same manner as discussed in Reference 7. The boundary condition of zero flow normal to the local surface is applied on the airfoil. For the purpose of the slat design, the pressure coefficients at selected points on the airfoil are specified. Thus, at these points both the normal and the tangential velocity components are specified; at the remaining points only the normal velocity component is specified. A Kutta condition is applied at the trailing edge of the main airfoil.

The slat is represented by a vortex sheet and by a source sheet in the vicinity of the airfoil leading edge. The source distribution provides the slat thickness; the vorticity distribution provides the slat camber. So that maximum versatility is provided, the locations of the slat vortex and source sheets are specified by the user. This permits the user to exercise control over the relative location of the slat and its shape. The singularity strengths at the vortex and source control points are the unknown parameters that will be fixed by the specified airfoil pressure conditions.

Two types of flow conditions are associated with the slat design. The first type satisfies the requirement that a specified pressure distribution be produced on the airfoil. The second type represents the closure conditions. These conditions ensure that the slat has a closed shape. The justification for each of the conditions is discussed in Subsection 2.2. Briefly, the conditions are as follows: (1) the net mass emitted by the source line is zero, (2) the source and vortex lines approximate a portion of some streamline around the airfoil without the slat, and (3) the stream function has the same value at the forward and rear stagnation points on the slat. The last condition implies an additional set of conditions, namely, that the locations of the forward and rear slat stagnation points must be specified.

The equations describing the boundary conditions on the wing and the conditions on the slat, listed above, form a linear system of algebraic equations. This system is solved by the Gauss-Jordan method to determine the unknown singularity strengths at the control points. This set of singularity strengths defines the flow field and provides sufficient information to determine the required slat shape.

The shape of the slat is determined by tracing the body streamlines from the forward stagnation point around the slat to the rear stagnation point. Since the initial slope of the streamlines is multi-valued, a special streamline integration routine is applied. The routine fits a segmented line between the two stagnation points. This line is required to have minimum directional deviation from the local flow direction. After the shape of the slat is determined, the pressure distributions on the slat and the wing are computed. The net forces on the individual elements, as well as on the entire system, are also computed.

The discussion above provides a brief description of the operation of the slat design method. Much of the method represents modifications of the airfoil potential-flow analysis procedure of Reference 7. Other portions of the slat design method were developed specifically for this program. A more detailed discussion of the method follows in the remainder of this section.

2.1 Airfoil Representation

The potential flow about an arbitrary airfoil can be simulated exactly by a vortex sheet with a continuously varying strength along the surface of the airfoil. The condition of tangential flow along the airfoil surface and the imposition of the Kutta condition establish the local strength of the vortex sheet.

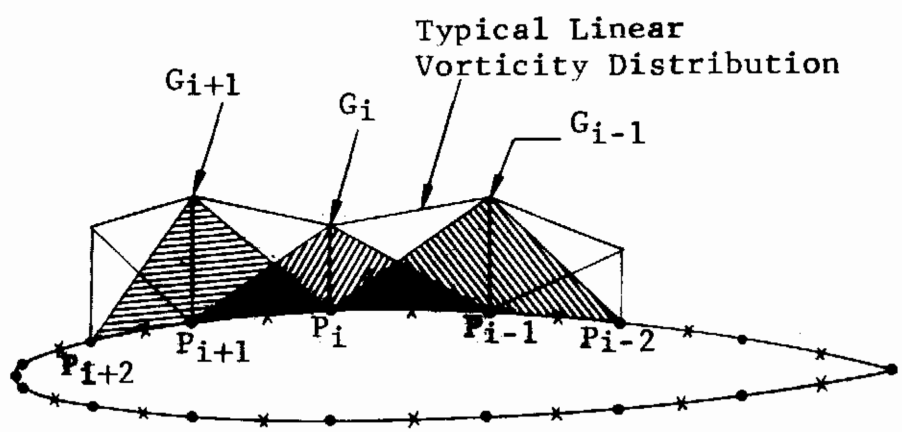
A reasonable approximation of this representation can be obtained by assuming that

1. The vortex sheet can be represented by a polygon inscribed within the airfoil. The vortex strength is allowed to vary linearly along each side of the polygon.

2. The tangential flow condition is imposed only at the mid-points of the polygon segments.

The vertices of the polygon are referred to as control or corner points, since it is at these points that the strength of the vortex sheet is controlled. The mid-points are boundary points, in reference to tangential-flow boundary condition. The general arrangement of this approximation is depicted in Figure 1. The locations of the corner points are arbitrary, but the resulting polygon should provide a good representation of the airfoil shape. Thus, short segments should be used in the leading-edge region, while longer segments are tolerable near the trailing edge.

The linear variation of the vortex strength over a typical segment is the sum of two triangular distributions (see Figure 1). One has the strength G_i and the other G_{i+1}. Thus, the influence of the vortex sheet can be represented by the summation (i.e., superposition) of the effects of a series of triangular distributions. The influence of a single triangular distribution is derived in Appendix I as a superposition of discrete vortex singularities. The velocity induced at the k<u>th</u> boundary point by the i<u>th</u> such distribution has the form



o Corner Points
* Boundary Points

Figure 1. General Arrangement for Airfoil Representation

$$u_{ik} = A_{ik} G_i$$
$$v_{ik} = B_{ik} G_i \qquad (1)$$

where u and v represent the horizontal and vertical velocity components, respectively. The coefficients A and B are the result of the singularity length and the relative locations of the boundary point and the singularity.

Normally, each G_i is associated with two triangular distributions, one on each side of the corner point. Thus, the coefficients are composed of

$$A_{ik} = \overset{+}{A}_{ik} + \overset{-}{A}_{ik}$$
$$B_{ik} = \overset{+}{B}_{ik} + \overset{-}{B}_{ik} \qquad (2)$$

A special case occurs at the airfoil trailing edge. The vortex sheet is assumed to begin at the trailing edge, pass over the upper surface of the airfoil to the nose, and then returning to the trailing edge along the lower airfoil surface. Thus, the first and last vortex control points only contribute to one distribution each. For this case Equation 2 is not needed, only Equation 1.

Another special case occurs when the boundary point is located on the influencing segment of the polygon. When this occurs, the distributed singularity is replaced by a discrete singularity at the corner point having the same net strength. The strength of this equivalent vortex, G_E, is

$$G_E = \tfrac{1}{2} d_k G_i \qquad (3)$$

where d_k is the length of the polygon segment. The coefficients for the induced velocity in this case are

$$A_{ik} = - \frac{d_k}{4\pi} \frac{y_i - y'_k}{(x_i-x'_k)^2 + (y_i-y'_k)^2} \qquad (4a)$$

$$B_{ik} = \frac{d_k}{4\pi} \frac{x_i - x'_k}{(x_i - x'_k)^2 + (y_i - y'_k)^2} \tag{4b}$$

The velocity at any point within the flow field is given by the summation of the velocities induced at that point by all of the singularities in the field plus the freestream velocity. Note that the influence of the source and vortex control points on the slat must be included in the summation. Letting G_i represent both the source and the vortex singularity strengths yields the velocity at any point (x'_k, y'_k):

$$u_k = \sum_{i=1}^{N} A_{ik} G_i + U_\infty \cos\alpha$$

$$v_k = \sum_{i=1}^{N} B_{ik} G_i + U_\infty \sin\alpha \tag{5}$$

where N is the total number of singularities.

The tangential flow condition requires that the component of the velocity normal to the airfoil surface at a boundary point be zero. If θ_k is the local slope at the k<u>th</u> boundary point, the condition is

$$\sum_{i=1}^{N} (A_{ik} \sin\theta_k - B_{ik} \cos\theta_k) G_i =$$

$$U_\infty (\sin\alpha \cos\theta_k - \cos\alpha \sin\theta_k) \tag{6}$$

The slope θ_k is understood to be the slope of the line joining the k<u>th</u> and the (k+1)<u>th</u> corner points; hence,

$$\theta_k = \arctan \frac{y_k - y_{k+1}}{x_k - x_{k+1}} \tag{7}$$

Equation 6 can be written in the form

$$\sum_{i=1}^{N} C_{ik} G_i = b_k \tag{8}$$

where C_{ik} is referred to as the influence coefficient of the ith singularity at the kth boundary point. An equation of this type is generated for each boundary point on the airfoil.

The number of boundary points on the airfoil is one less than the number of corner points. The Kutta condition provides the additional equation to give a balanced system of equations and unknowns for the airfoil. The condition requires that the net vorticity at the trailing edge be zero. Since two corner points are coincident at the trailing edge, the condition is

$$G_F + G_L = 0 \tag{9}$$

where G_F is the vortex strength at the first corner point and G_L is the strength at the last corner point.

Equations 8 and 9 combine to form a system of N_c linear algebraic equations for the singularity strengths at the N_c airfoil control points. This system can be solved to define the flow about the airfoil in the absence of a leading-edge slat.

2.2 Slat Representation

The leading-edge slat is represented by a set of distributed singularities located in the vicinity of a leading-edge slat, i.e., external to the wing. The singularity strengths vary linearly along straight lines between fixed control points. The singularities are distributed vortices and sources. These external vortices induce velocities in the same manner as the vortex distribution on the wing. Thus, the same methods can be used to compute the influence coefficients related to these external vortices. The velocity induced by the source distribution can be derived very simply by comparing the velocities induced by a single discrete vortex and a source. The velocity induced at (x,y) by a discrete source at (x',y') is

$$u_M = \frac{M}{2\pi} \frac{x - x'}{(x-x')^2 + (y-y')^2} \tag{10}$$

$$v_M = \frac{M}{2\pi} \frac{y - y'}{(x-x')^2 + (y-y')^2} \tag{11}$$

Similarly, for a vortex,

$$u_G = -\frac{G}{2\pi}\;\frac{y - y'}{(x-x')^2 + (y-y')^2} \qquad (12)$$

$$v_G = \frac{G}{2\pi}\;\frac{x - x'}{(x-x')^2 + (y-y')^2} \qquad (13)$$

Eliminating the geometric terms yields

$$u_M = \frac{M}{G}\, v_G \qquad v_M = -\frac{M}{G}\, u_G \qquad (14)$$

Thus, the velocity induced at (x,y) by a source at (x',y') can be related to the velocity induced by a vortex at the same location. Since the effect of distribution of the singularity strengths is a geometric relation, Equation 14 remains valid for distributed sources and vortices.

The stated purpose of the slat design is to produce a specified pressure distribution on the wing. Thus, a set of conditions is required that specifies the pressure on the wing at a selected interval of boundary points. The velocity at the kth point is given by

$$u_k = \sum_{i=1}^{N} A_{ik}\, G_i + U_\infty \cos\alpha - Q_k \cos\theta_k \qquad (15)$$

$$v_k = \sum_{i=1}^{N} B_{ik}\, G_i + U_\infty \sin\alpha - Q_k \sin\theta_k \qquad (16)$$

The term Q_k accounts for the influence of the vortex sheet on which the boundary point occurs. It is given by

$$Q_k = \frac{G_k + G_{k+1}}{4} \qquad (17)$$

The velocity component tangential to the surface is

$$\vec{V}\cdot\hat{s} = u_k \cos\theta_k + v_k \sin\theta_k = U_\infty\sqrt{1-C_p} \qquad (18)$$

Substituting Equations 15 and 16 into 17 yields

$$\sum_{i=1}^{N} A_{ik}\, G_i \cos\theta_k + \sum_{i=1}^{N} B_{ik}\, G_i \sin\theta_k - Q_k =$$

$$U_\infty\left(\sqrt{1-C_p} - \cos\alpha \cos\theta_k - \sin\alpha \sin\theta_k\right) \qquad (19)$$

This equation specifies the relationship between the singularity strengths and the local pressure, C_p. It provides the set of linear equations, one for each pressure condition, that are included in the system to be solved for the singularity strengths.

A necessary, but not sufficient, condition for the slat being a closed body is that the net strength of the slat source line is zero. Thus, no net mass is added to or removed from the system. Mathematically, this condition is expressed

$$M_1\, d_1 + \sum_{i=2}^{N_p-1} M_i(d_i + d_{i-1}) + M_{N_p}\, d_{N_p-1} = 0 \qquad (20)$$

where M_i represents the source strengths at the N_p control points, and d_i represents the distance from control point M_i to control point M_{i+1}. Equation 20 is linear in M_i and, as such, is included in the system to be solved for the singularity strengths.

One additional requirement must be met to assure a closed-slat shape. The streamlines that emanate from the forward stagnation point must also pass through the rear stagnation point. One method of satisfying this requirement is to force the stream function to have the same value at the forward and the rear stagnation points. Thus,

$$\psi_{LE} - \psi_{TE} = 0 \qquad (21)$$

The value of the stream function at any point (x,y) is given by

$$\psi = \sum_{i=1}^{N} D_i\, G_i + (y \cos\alpha - x \sin\alpha)\, U_\infty \qquad (22)$$

where G_i represents the strength of the <u>ith</u> singularity (source or vortex). The term D_i represents the geometric influence coefficient of the ith control point on the stream function at (x,y). The derivation of the form of D_i is given in Appendix I.

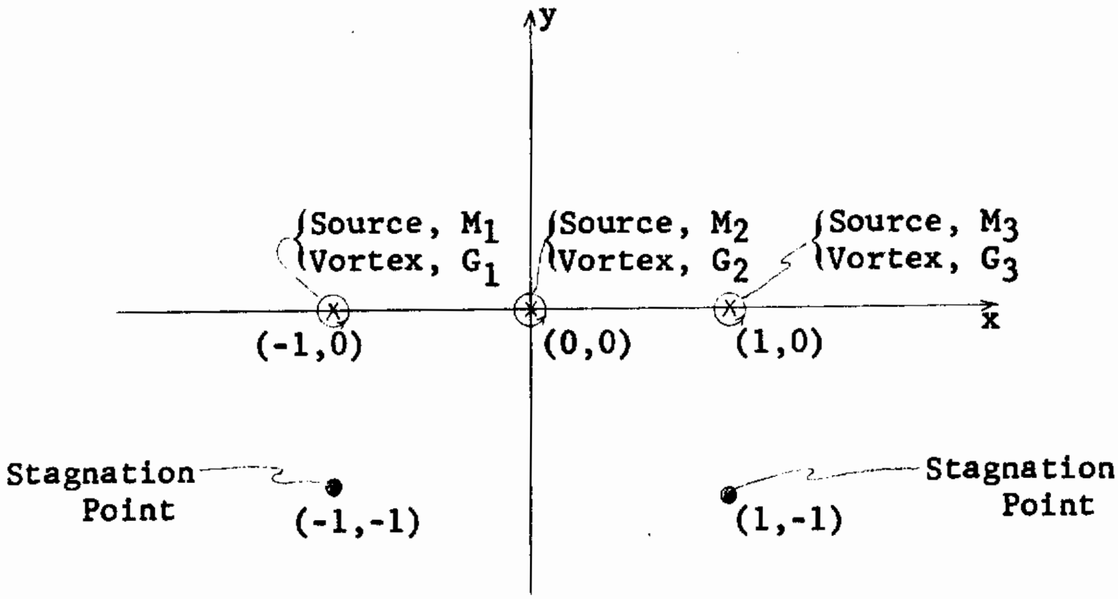
Equation 21, in conjunction with Equation 22, represents another linear algebraic equation governing the strengths of the singularities, if the locations of the stagnation points are known. This property can be used only if the stagnation-point locations are known a priori. Thus, additional conditions must be placed on the flow-field solution. The conditions are given by Equations 15 and 16 with u and v set equal to zero. Since the stagnation points do not lie on a vortex sheet, Q = 0 and the conditions are

$$\sum_{i=1}^{N} A_i \, G_i = - U_\infty \cos\alpha$$
$$\sum_{i=1}^{N} B_i \, G_i = - U_\infty \sin\alpha \qquad (23)$$

where the A_i and B_i are the influence coefficients evaluated at the assumed stagnation points.

The application of the above conditions is still not sufficient. When these conditions were utilized on a sample problem, the influence-coefficient matrix was found to be singular. This implied that no singularity strength distribution along the specified singularity line would produce a close-slat configuration. An examination of several very simple problems demonstrated why the matrix was singular and permitted a resolution of the difficulty.

Of the several problems studied, one provides a particularly clear demonstration of the closure constraint. The case consists of three discrete sources (or sinks) and three discrete vortices placed in a uniform flow field, as shown below. Stagnation points are specified to occur at (-1,-1) and (1,-1). The summation of the source strengths is required to be zero. These conditions define the singularity strengths of five of the six singularities. The sixth one, arbitrarily chosen to be G_2, is left unspecified with the others defined in terms of G_2. The resulting configuration for G_2 = 0 and a zero angle of attack is shown in Figure 2.

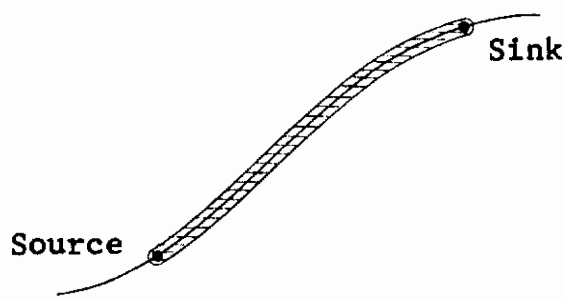


As mentioned above, the configuration will be closed only if the stream functions at the two stagnation points are equal. Thus, the value of G_2 should be selected such that the difference between the two stream function values be zero. For this problem the difference is

$$\Delta\psi = -\ (\ln 25 + 6\pi + 4\ \tan^{-1}2 + 2)\ \sin\alpha \tag{24}$$

The terms within the parentheses are a result of the geometric arrangement of the problem. They are independent of G_2 and all of the other singularity strengths. Thus, the configuration will be closed if and only if $\sin\alpha = 0$, i.e., a zero angle of attack.

Another simple problem provides a physical interpretation of this result. Consider a source-sink pair of equal strengths located on the same undisturbed streamline, as shown below. Let their strengths be small enough so that



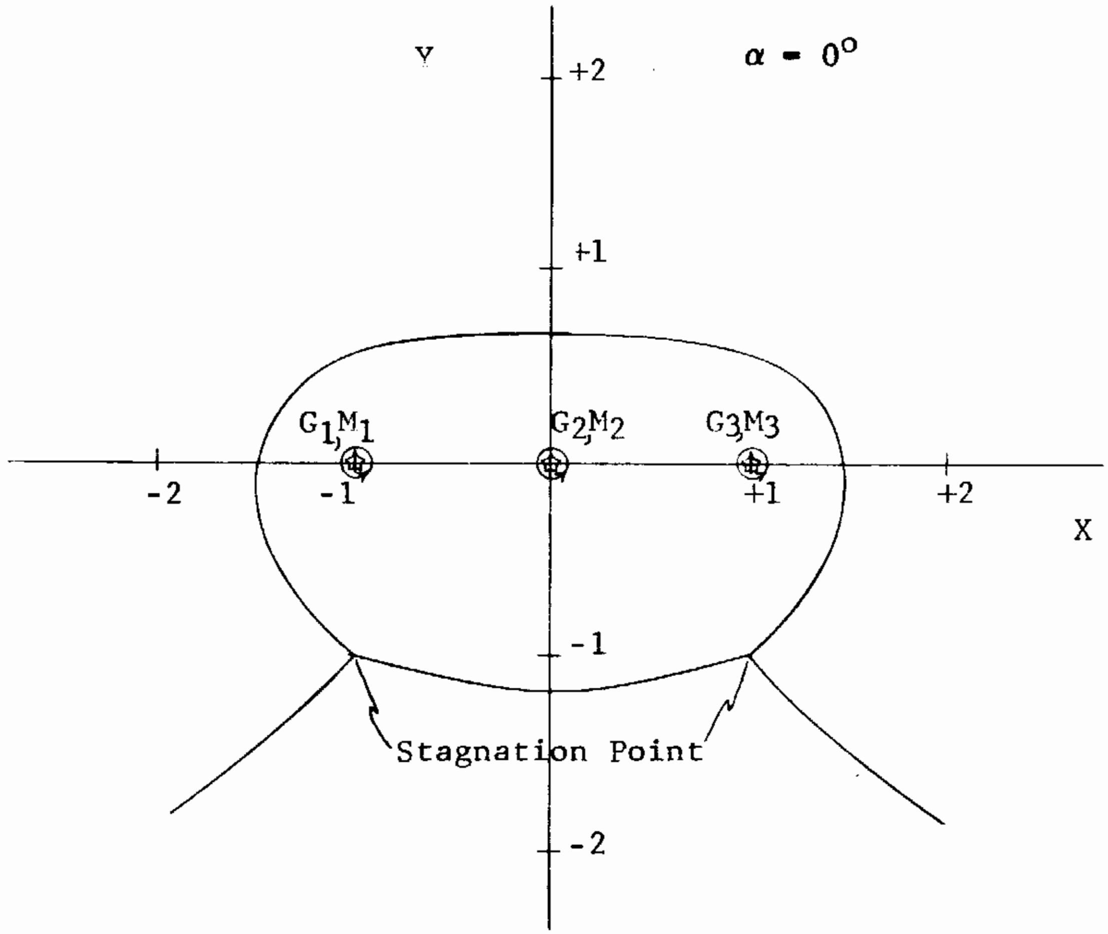
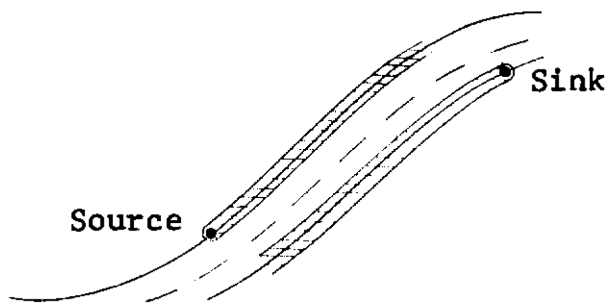


Figure 2. Simple Test Problem

their effects are only small perturbations on the surrounding flow field. Because of the small net flow from the source, the mass it emits will very nearly follow the undisturbed streamline. When this mass reaches the sink, it is completely absorbed. As the strengths of the singularities are increased, the flow pattern remains essentially the same, except that the perturbation on the external flow field becomes larger. Now consider the same problem, only with the sink placed on a different streamline, as shown below. The relative displacement of the sink permits a



streamline to pass between it and the source. This prevents the sink from absorbing the mass emitted by the source. As the singularity strengths are increased, the local perturbation becomes large enough that some of the source mass is absorbed by the sink, but not all of it. Thus, a closed configuration is possible only if the singularities are located on a single streamline in the undisturbed flow field. For the slat design problem, the slat singularities must be placed on a streamline generated by the airfoil in the absence of the slat.

2.3 Flow-Field Solution

Each of the flow-field conditions discussed above can be written in the form of a linear algebraic equation in terms of the unknown singularity strengths. Seven of the conditions, or equations, are present in all problems. They are (1) airfoil Kutta condition, Eq. 9; (2) mass conservation on the slat, Eq. 20; (3) equalization of slat stream function, Eq. 21; and (4) through (7) specification of slat leading- and trailing-edge locations, Eq. 23. In addition, there are N_c-1 equations which assure tangential flow over the airfoil. The remaining M_p equations specify the pressure at M_p points on the airfoil. Thus, a total of $N_c + M_p + 6$ equations are required for each problem.

Now consider the number of unknown singularity strengths. The airfoil vortex sheet has N_c corners; hence, N_c singularity strengths need to be specified. Since $N_c + M_p + 6$ equations are required, $M_p + 6$ slat singularity control points must be available. Thus, seven slat control points are needed to specify the pressure at a single point on the airfoil.

The linear form of these equations allows efficient solution by means of several standard methods. The Gauss-Jordan method was utilized in this study. The solution defines the singularity strengths at each of the control points on the airfoil and on the slat. From this data the conditions at any non-singular point in the flow field can be computed. Thus, the slat shape is defined but still unknown at this point.

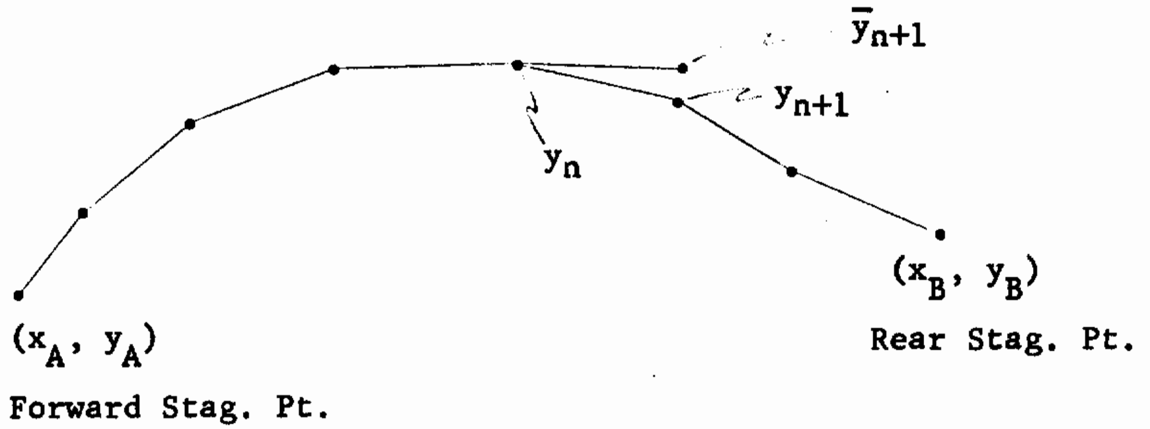
2.4 Determination of Slat Shape

The slat shape is determined by calculating the shapes of its body streamlines. These are the two streamlines that emanate from the forward stagnation point on the slat and pass to the rear stagnation point. In the computerized version of the technique, the flow-field properties are calculable at nearly all points in the field. The primary exception are points coinciding with distributed singularities. Also, the locations of the forward and rear stagnation points are known. Thus, determination of the slat shape appears to be simply a problem of computing the shapes of two streamlines. Unfortunately, the streamline slopes become indeterminate at the stagnation points, so that the initial step in the integration is not reliable. The result is an erroneous shape definition. In the discussion below, an alternate, more accurate method for determining the slat shape is described.

The basic idea of the method is to determine a segmented line which emanates from the forward stagnation point and terminates at the rear stagnation point and which minimizes the difference between its slopes and that of the local flow velocity vectors. Consider the typical segmented line shown below. The end points are given by

$$y_B = y_A + \sum_{n=1}^{N} d \sin \theta_n \qquad (25a)$$

$$x_B = x_A + \sum_{n=1}^{N} d \cos \theta_n \tag{25b}$$



where d is the length of all segments and N is the total number of segments. Each successive corner point is defined by

$$y_{n+1} = y_n + \sin \theta_n \, d \tag{26}$$

$$x_{n+1} = x_n + \cos \theta_n \, d \tag{27}$$

If N is made infinitely large, then the line should approach a streamline, or

$$\theta_n \rightarrow \bar{\theta}_n = \tan^{-1} \frac{v}{u} \tag{28}$$

The streamline projection point is given by

$$\begin{aligned} \bar{y}_{n+1} &= y_n + \sin \bar{\theta}_n \, d \\ \bar{x}_{n+1} &= x_n + \cos \bar{\theta}_n \, d \end{aligned} \tag{29}$$

The velocities are computed at the midpoint of the segment, which is

$$\begin{aligned} y_c &= (y_n + y_{n+1})/2 \\ x_c &= (x_n + x_{n+1})/2 \end{aligned} \tag{30}$$

The difference between this slope and the corresponding slope on the segmented line is a measure of the error in segmented-line approximation. It is

$$e_n = (\theta_n - \bar{\theta}_n)^2 \tag{31}$$

The total error is the sum of all of these, or

$$e = \sum_{n=1}^{N} (\theta_n - \bar{\theta}_n)^2 \tag{32}$$

The task is to minimize e subject to the constraint that Equation 25 remains valid.

The method of Lagrangian undetermined multipliers is applied to determine the θ_n and d such that e is a minimum and Equation 25 is satisfied. Equation 32 may be written

$$e = \sum_{n=1}^{N} (\theta_n - \bar{\theta}_n)^2 + H\left[y_A - y_B + \sum_{n=1}^{N} \sin\theta_n\, d\right]$$

$$+ I\left[x_A - x_B + \sum_{n=1}^{N} \cos\theta_n\, d\right] \tag{33}$$

where H and I are the undetermined multipliers. Holding v_n and u_n temporarily constant, e may be differentiated as

$$\frac{\partial e}{\partial \theta_n} = 2(\theta_n - \bar{\theta}_n) + Id\sin\theta_n - Hd\cos\theta_n = 0 \tag{34}$$

$$\frac{\partial \theta_n}{\partial d} = H\sum_{n=1}^{N} \sin\theta_n + I\sum_{n=1}^{N} \cos\theta_n = 0 \tag{35}$$

Equation 34 defines the set of θ_n in terms of the unknowns H, I, and d and the set of $\bar{\theta}_n$. In theory it could be used to eliminate the θ_n from Equations 25 and 35. The result would be a set of three simultaneous equations in H, I, and d, namely

$$F_1(H, I, d) = \sum_{n=1}^{N} \sin\theta_n + \sum_{n=1}^{N} \cos\theta_n = 0 \tag{36a}$$

$$F_2(H, I, d) = y_B - y_A - d \sum_{n=1}^{N} \sin \theta_n = 0 \qquad (36b)$$

$$F_3(H, I, d) = x_B - x_A - d \sum_{n=1}^{N} \cos \theta_n = 0 \qquad (36c)$$

Unfortunately, the transcendental form of Equation 34 prevents it from being solved for θ_n in closed form. A simple Newton-Raphson iteration provides an efficient numerical solution to the equation. The iteration is defined by

$$\theta_n^{(v+1)} = \theta_n^{(v)} - (F/F')^{(v)} \qquad (37)$$

where

$$\begin{aligned} F &= 2(\theta_n - \bar{\theta}_n) + Id \sin \theta_n - Hd \cos \theta_n \\ F' &= 2 + Id \cos \theta_n + Hd \sin \theta_n \end{aligned} \qquad (38)$$

Thus, for a given set of I, d, and H, a set of θ_n can be determined. These three parameters can be computed by solving Equations 36. A numerical approximation of the solution can be obtained by application of a three-dimensional form of the Newton-Raphson iteration method. In vector form, the iteration scheme is

$$\underline{X}^{(v+1)} = \underline{X}^{(v)} - \underline{\underline{A}}^{-1} \cdot \underline{F}^{(v)} \qquad (39)$$

where

$$\underline{F} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} \qquad \underline{X} = \begin{pmatrix} H \\ I \\ d \end{pmatrix} \qquad 40)$$

and

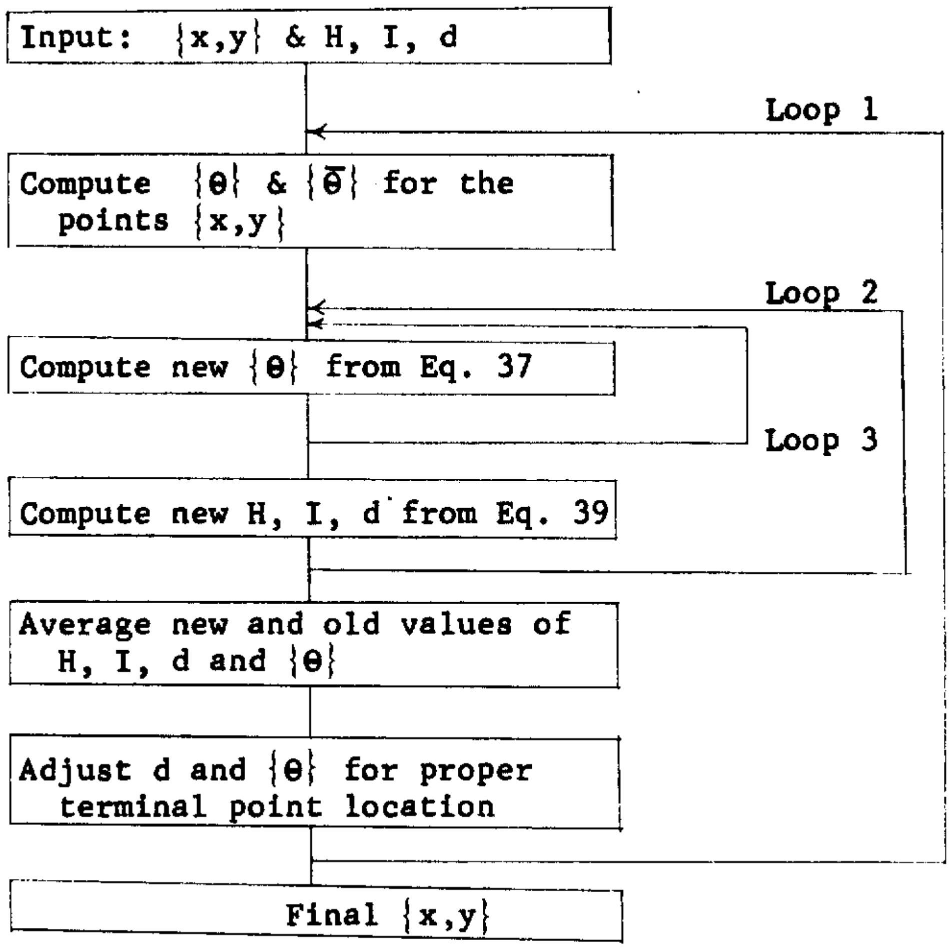
$$\underline{\underline{A}} = \begin{pmatrix} F_{1H} & F_{1I} & F_{1d} \\ F_{2H} & F_{2I} & F_{2d} \\ F_{3H} & F_{3I} & F_{3d} \end{pmatrix} \qquad (41)$$

where

$$F_{1H} = \frac{\partial F_1}{\partial H}\ ,\ F_{2I} = \frac{\partial F_2}{\partial I},\ \text{etc.}$$

These derivatives, as derived from Equations 34 and 36, are listed in Table I.

The computation of the slat shape through the use of these equations is accomplished by a three-level iteration scheme. The diagram below describes the general scheme of how the different levels interact. (A more detailed description is provided in Appendix II.) As with most nonlinear



iteration schemes, this scheme is not the only workable method. Of the methods examined, it has the fastest convergence rate and the most uniform error distribution. The integration (Loop 1) is continued until the error e (Eq. 32) is negligibly small. The resulting segmented line represents an approximation of the slat body streamline.

Table I

CROSS DERIVATIVES FOR NEWTON-RAPHSON ITERATION

$$F_{1H} = \sum_{n=1}^{N} \sin \theta_n + H \sum_{n=1}^{N} \cos \theta_n \frac{\partial \theta_n}{\partial H} - I \sum_{n=1}^{N} \sin \theta_n \frac{\partial \theta_n}{\partial H}$$

$$F_{2H} = -d \sum_{n=1}^{N} \cos \theta_n \frac{\partial \theta_n}{\partial H}$$

$$F_{3H} = d \sum_{n=1}^{N} \sin \theta_n \frac{\partial \theta_n}{\partial H}$$

$$F_{1I} = \sum_{n=1}^{N} \cos \theta_n + H \sum_{n=1}^{N} \cos \theta_n \frac{\partial \theta_n}{\partial I} - I \sum_{n=1}^{N} \sin \theta_n \frac{\partial \theta_n}{\partial I}$$

$$F_{2I} = -d \sum_{n=1}^{N} \cos \theta_n \frac{\partial \theta_n}{\partial I}$$

$$F_{3I} = d \sum_{n=1}^{N} \sin \theta_n \frac{\partial \theta_n}{\partial I}$$

$$F_{1d} = H \sum_{n=1}^{N} \cos \theta_n \frac{\partial \theta_n}{\partial d} - I \sum_{n=1}^{N} \sin \theta_n \frac{\partial \theta_n}{\partial d}$$

$$F_{2d} = - \sum_{n=1}^{N} \sin \theta_n - d \sum_{n=1}^{N} \cos \theta_n \frac{\partial \theta_n}{\partial d}$$

$$F_{3d} = - \sum_{n=1}^{N} \cos \theta_n + d \sum_{n=1}^{N} \sin \theta_n \frac{\partial \theta_n}{\partial d}$$

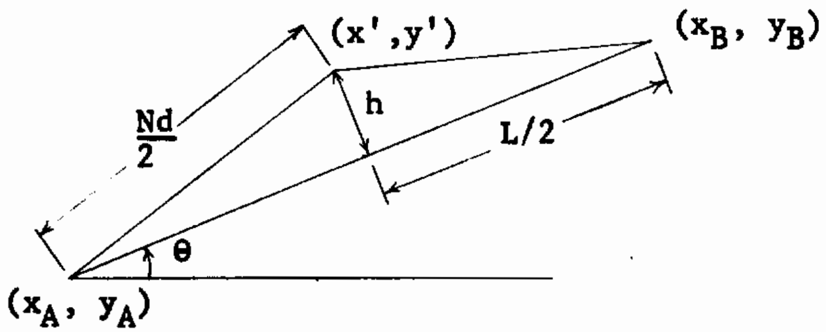
$$\frac{\partial \theta_n}{\partial H} = d \cos \theta_n / D_n \ ; \quad \frac{\partial \theta_n}{\partial I} = -d \sin \theta_n / D_n \ ;$$

$$\frac{\partial \theta_n}{\partial d} = (H \cos \theta_n - I \sin \theta_n) / D_n$$

$$D_n = 2 + Hd \sin \theta_n + Id \cos \theta_n$$

The initial line provides the first guess of the slat shape for the iteration procedure. By modifying the particular initial shape, either the upper or lower slat surface can be determined.

The initial line is composed of two linear segments, as shown below. Each of these segments is divided into N/2



parts (N must be even). If L is the distance from (x_A, y_A) to (x_B, y_B), then the length of the small segments is

$$d = \frac{L}{N}\sqrt{1 + 4h^2/L^2} \tag{42}$$

The coordinates of the midpoint on the line are

$$\begin{aligned} x' &= \frac{x_A+x_B}{2} \cos\theta - (y_A \pm h) \sin\theta \\ y' &= \frac{x_A+x_B}{2} \sin\theta + (y_A \pm h) \cos\theta \end{aligned} \tag{43}$$

where the upper sign corresponds to the upper surface of the slat and vice versa. The coefficients for Equation 25 are given by

$$\sin\theta_n = \begin{cases} \dfrac{y'-y_A}{Nd/2} \\ \\ \dfrac{y_B-y'}{Nd/2} \end{cases} \quad \cos\theta_n = \begin{cases} \dfrac{x'-x_A}{Nd/2} & n < N/2 \\ \\ \dfrac{x_B-x'}{Nd/2} & n \geq N/2 \end{cases} \tag{44}$$

Equations 44 in combination with Equations 42 and 43 provide the definition of both the upper and lower initial lines.

The accuracy of the slat shape definition routine was examined by a simple test case. A velocity field was specified, which corresponded to the flow about a circular cylinder. The routine computed the coordinates of the cylinder to an accuracy of 0.03% of the radius. The mean local deviation between the velocity direction and the line slope was only 0.14 degrees.

2.5 Pressure and Force Computation

The velocity at each airfoil boundary point is given by Equations 15 and 16. The velocity at each slat definition point is also given by Equations 15 and 16, except with Q $Q_k = 0$, since the points are not on a vortex sheet. The pressure coefficients at these locations are given by the incompressible relation

$$C_{p_k} = 1 - \frac{u_k{}^2 + v_k{}^2}{U_\infty{}^2} \qquad (45)$$

The pressures are numerically integrated to give the force and moment coefficients of both elements and of the total system.

3. APPLICATION OF METHOD

A study was conducted to examine the performance of this slat design method. The purpose of the study was not so much to design perfect slats as it was to determine the effects of three design parameters: (1) the location of the leading-edge stagnation point, (2) the location of the trailing-edge stagnation point, and (3) the specified design pressure on the wing. The effects of each of these parameters on the slat design are discussed below.

A very simple problem was chosen for this study so that the effects of the various slat design parameters could be readily identified. The airfoil was a simple symmetric airfoil operating at an angle of attack of five degrees. The slat singularities were placed on a straight line. The line was oriented such that it approximated a streamline in the undisturbed airfoil flow field. The simplicity of the problem was maintained by specifying only one pressure on the airfoil. Thus, only seven singularity control points were required, four sources and three vortices. This basic configuration is shown in Figure 3.

The effect of modifying the leading-edge location is shown in Figure 4. Each design shown has the same trailing-edge location and produces the same pressure at the airfoil pressure control point. In the figure, d_{LE} is the distance from the leading-edge stagnation point to the first control point. Two trends are indicated by this comparison. First, an increase in the distance between the first source control point and the leading-edge point increases the radius of curvature of the leading edge. The second trend is actually a result of the first: the larger the separation distance, the thicker the slat design. The same trend is valid for the trailing-edge location. The slight indentation at the leading edge was also noted on other configurations not presented here.

The resulting pressure distributions on the slat and the airfoil are shown in Figure 5. Both designs create a rather sharp peak at the design pressure point, as compared to the solution without the slat.

The location of the trailing edge affects the apparent slat trailing-edge angle (see Figure 6). The closer the stagnation point is to the last source control point, the smaller the included angle. Note the relative sizes of the distance from the source point to the trailing edge, d_{TE}. Only when that distance is very small does the slat become slender. This implies that the trailing edge should coincide with the last source control point. Presumably, this would eliminate the rounded trailing edge that seems to be characteristic of these slats. Such a formulation has not been attempted.

Both slat designs produce essentially the same pressure distributions on the airfoil (see Figure 7). The thicker slat reduces the load on the airfoil slightly more than does the thinner slat. Note that both slats reduce the suction peak on the airfoil significantly.

The final comparison shows the effect of changing the design pressure. The effect on the slat shape is shown in Figure 8. As would be expected, the case with the higher suction ($C_p = -2$) has a lesser gap. The pressure increase (to $C_p = -1$) is gained by increasing the camber of the slat with only a slight change in slat thickness.

The two designs result in significantly different airfoil pressure distributions near the slat (Fig. 9). The high-suction case ($C_p = -2$) produces a very sharp spike; the low-suction case ($C_p = -1$) produces a relatively flat-topped pressure distribution. On the basis of this comparison a reasonable pressure distribution with reduced suction can be obtained by use of a single pressure control point; multiple control points are required to produce a smooth pressure distribution when airfoil suction is increased.

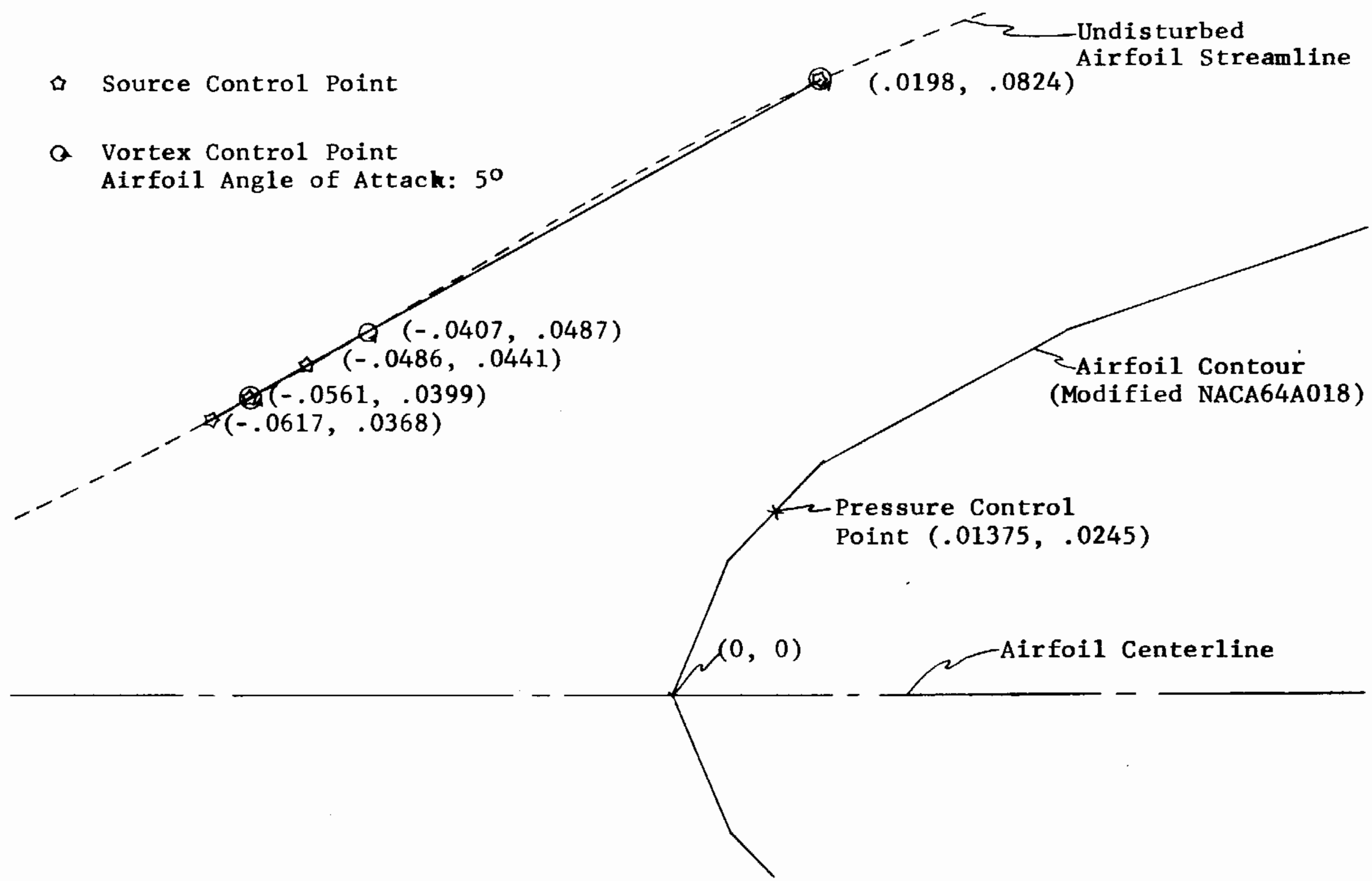


Figure 3. Slat Singularity Control-Point Geometry for Design Study

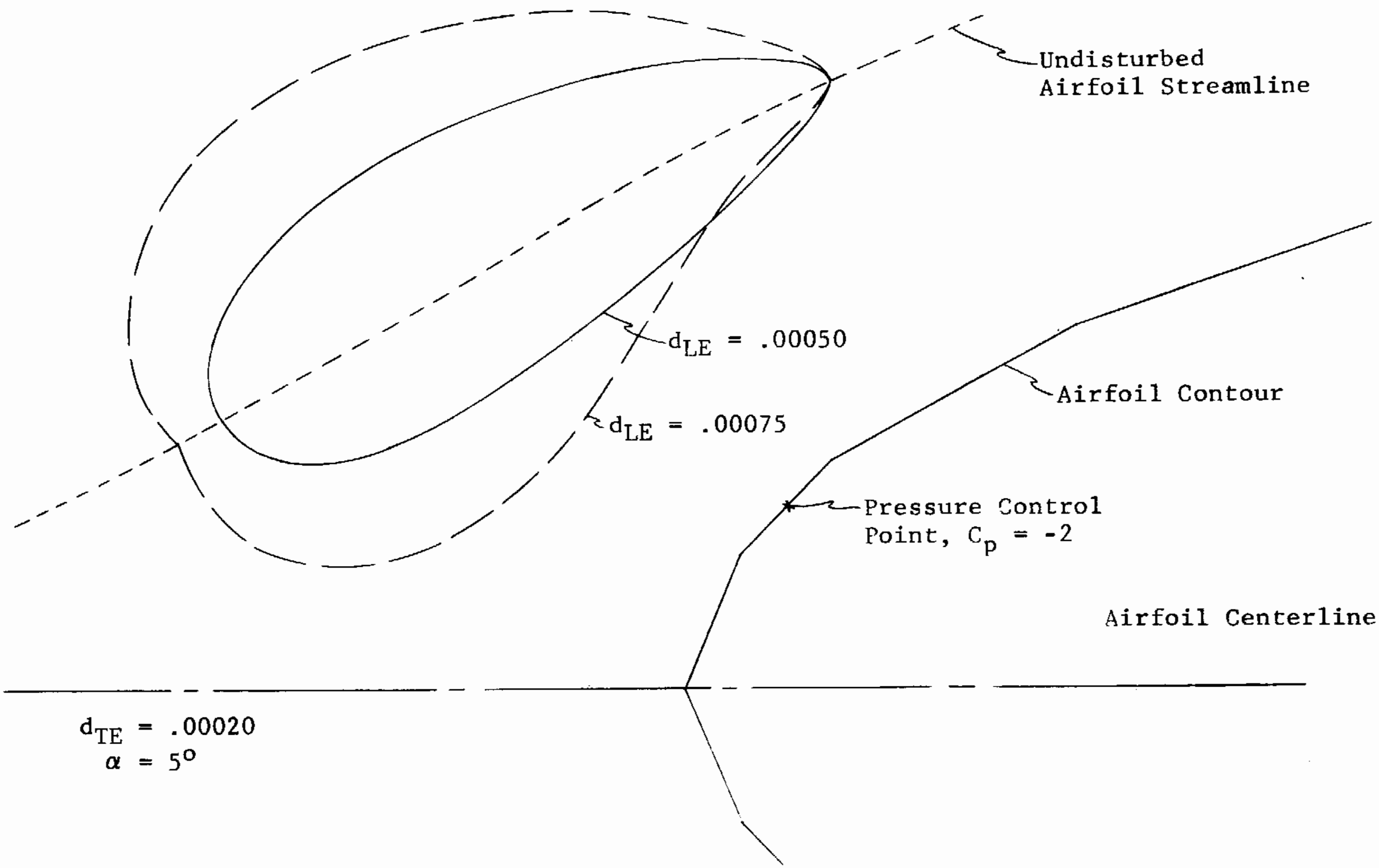


Figure 4. Effect of Slat Leading-Edge Location on Slat Geometry

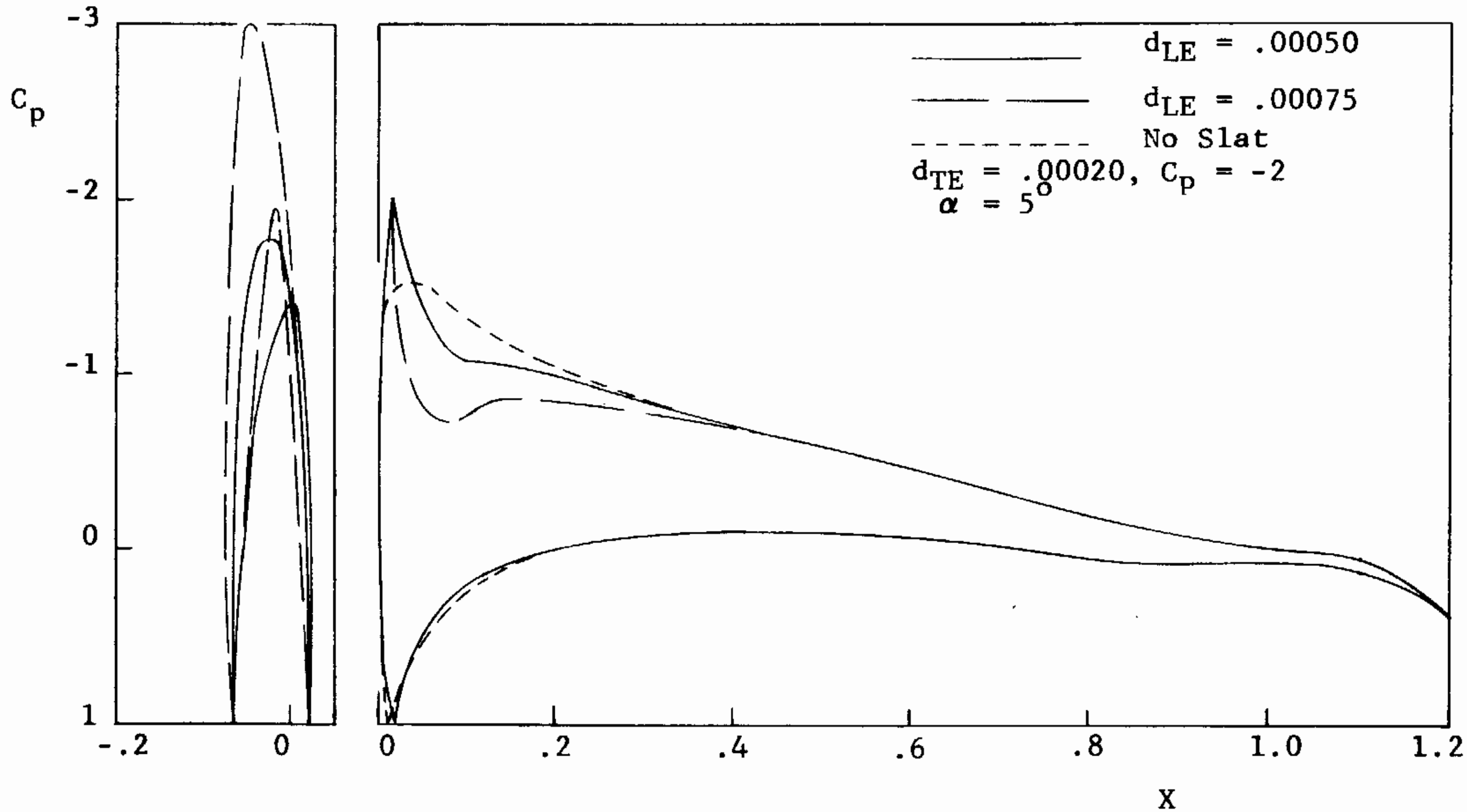


Figure 5. Effect of Slat Leading-Edge Location on Slat and Airfoil Pressure Distributions

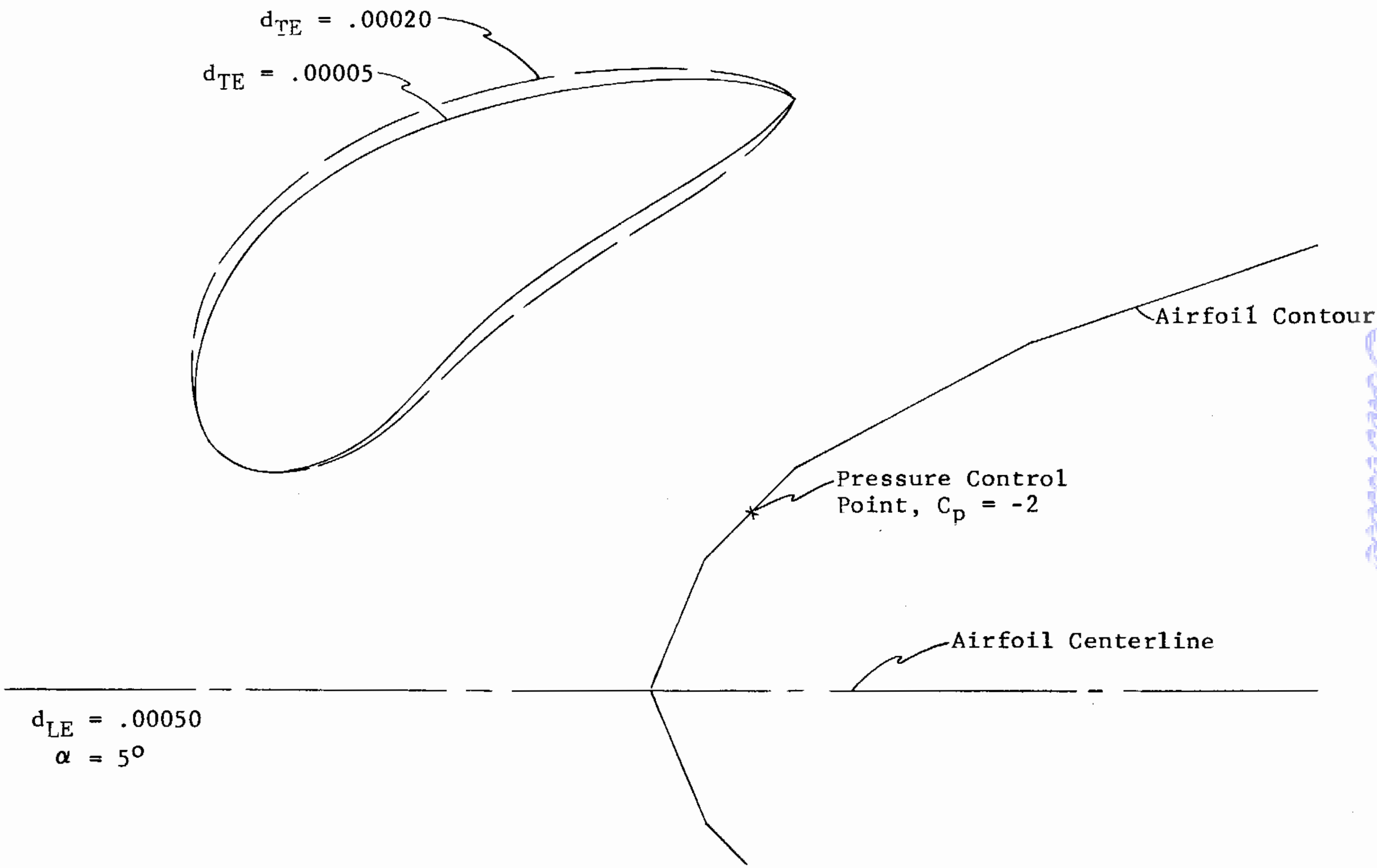


Figure 6. Effect of Slat Trailing-Edge Location on Slat Geometry

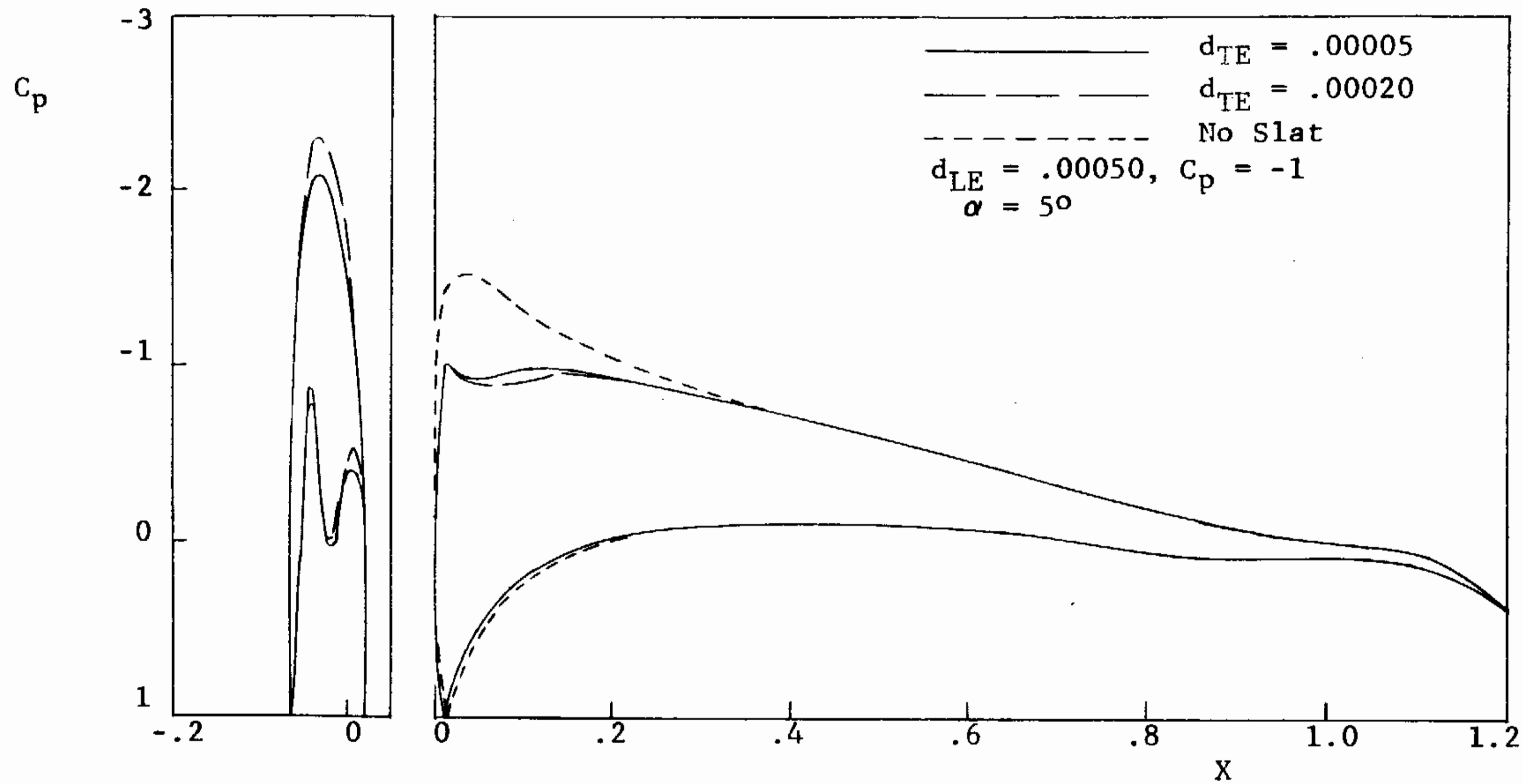


Figure 7. Effect of Slat Trailing-Edge Location on Slat and Airfoil Pressure Distributions

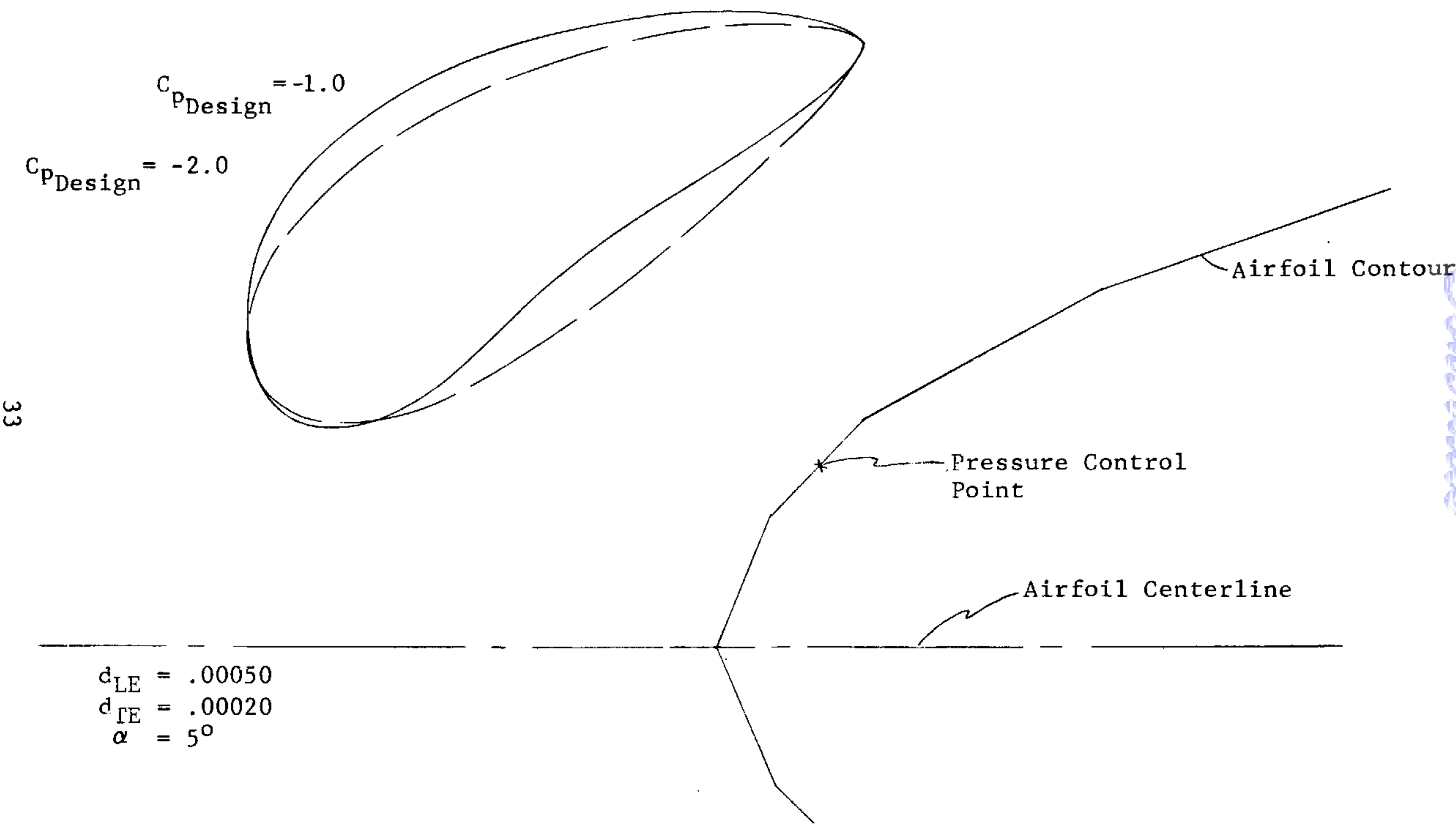


Figure 8. Effect of Design Pressure on Slat Geometry

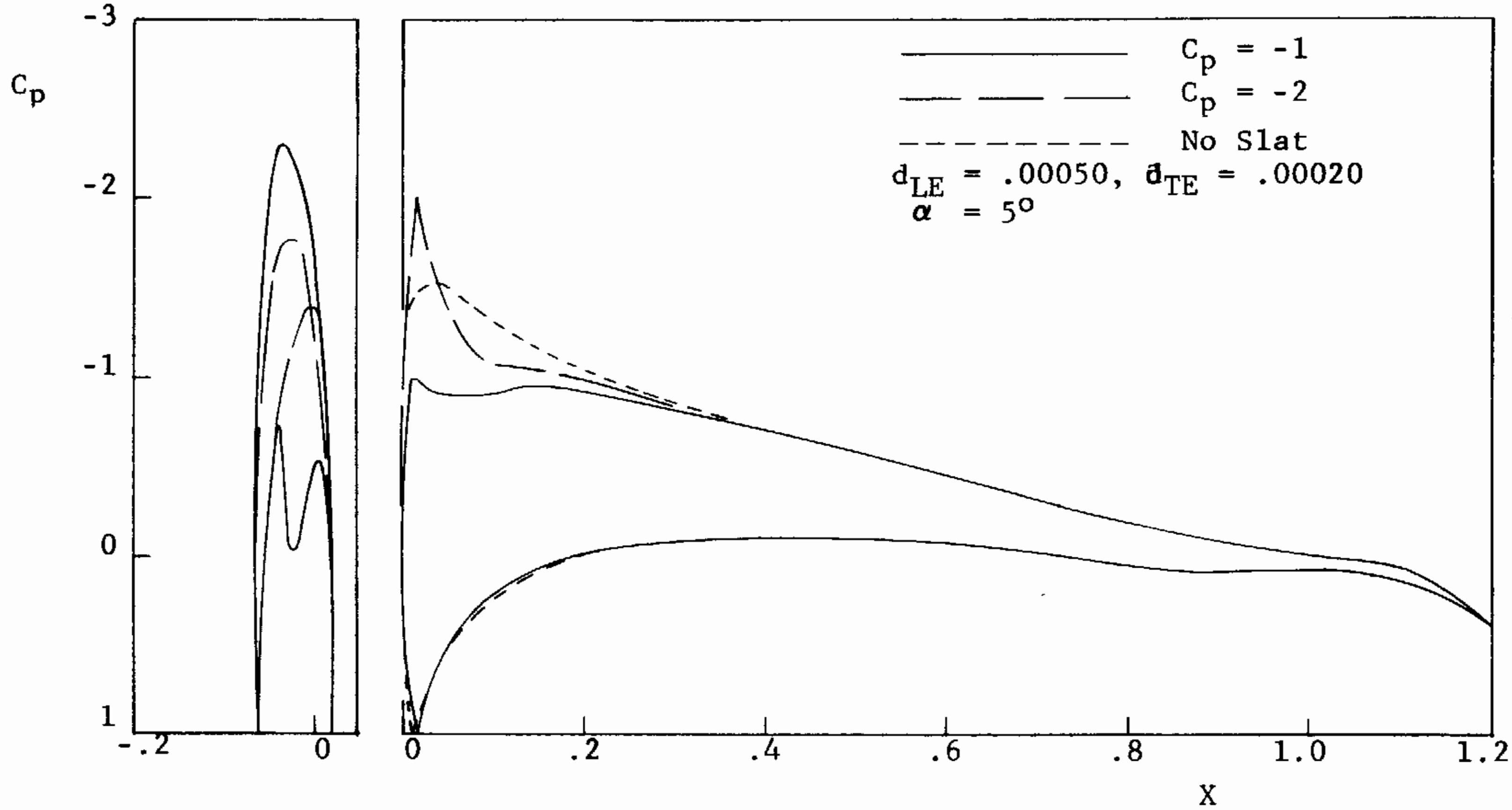


Figure 9. Effect of Design Pressure on Slat and Airfoil Pressure Distribution

4. UTILIZATION OF METHOD

One of the principal shortcomings of an inverse design technique is that it requires experience in order to obtain good designs. The amount of experience required is significantly reduced if a good set of design guidelines is available. In this section, such a set of guidelines is provided through step-by-step instructions on the utilization of the design technique. In particular, the instructions apply to the utilization of the computer program developed in this study (see Appendix II).

Step 1. The first task is to obtain a potential-flow solution for the airfoil alone at the angle of attack of interest. This solution serves a two-fold purpose: (1) it provides the baseline airfoil pressure distribution, and (2) it permits a set of streamlines (about five, normally) to be traced in the proposed vicinity of a leading-edge slat.

Step 2. The design pressure distribution is specified next. Note that the slat will generally only affect the pressures in the leading edge. For preliminary work, only one or two pressures need be specified. All pressures control points should be on the upper surface of the airfoil.

Step 3. Next, the streamlines traced in Step 1 are plotted on an expanded scale (see Fig. 3) along with the airfoil and the pressure control points. At this point some engineering judgment must be used. Select a portion of one of the streamlines that is relatively straight and near the pressure control points, yet has the location of a reasonable slat configuration. This line will form the vortex and source sheets for the slat. It will also form the approximate chord line of the slat. The program will permit curved singularity lines, but their utilization is much more complex and they do not provide significantly better results.

Step 4. A straight line is drawn along the selected streamline. Its length should be approximately 90 percent of the desired slat chord length. The orientation and location of the line should provide a reasonable approximation of the streamline. This line will be the slat singularity line.

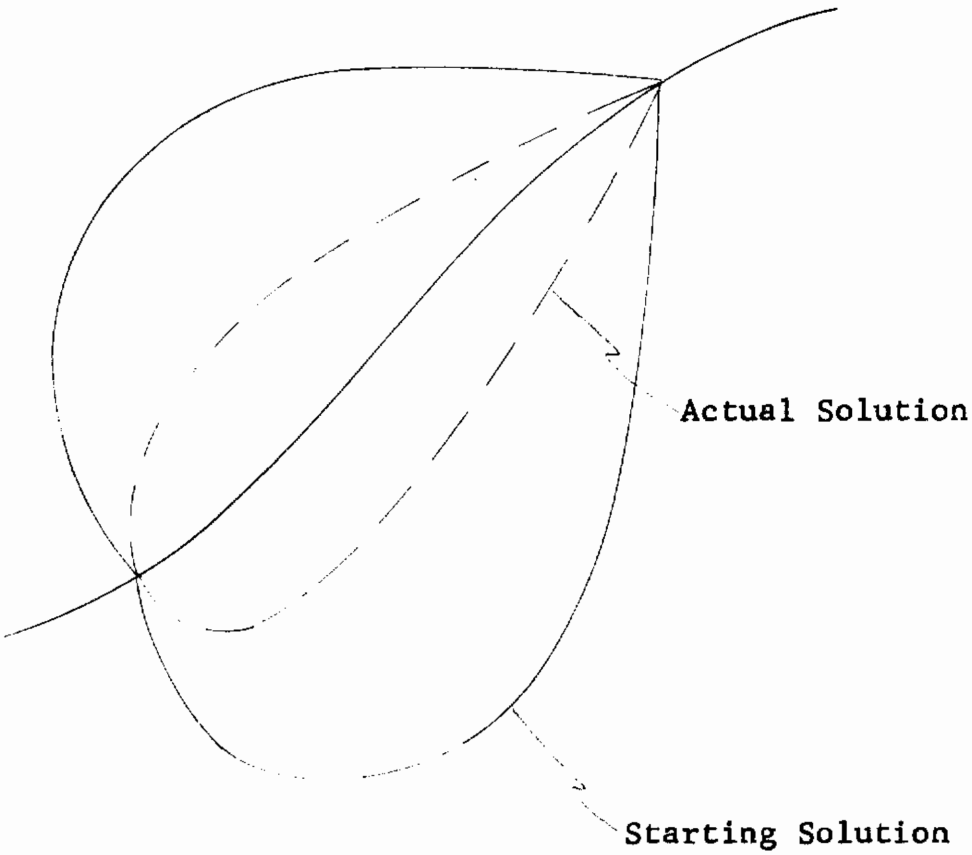
Step 5. The number and location of vortex and source control points should be selected next. $N_p + 6$ control points are needed (N_p is the number of pressure control points). The singularities should be divided evenly between vortices and sources, with the odd point being a source point. The distribution along the singularity line is not critical, but a source control point should be placed at each end of the line and a vortex control point at the aft end. The best results are obtained when the sources are concentrated near the leading-edge region. Care should be taken that the control-point locations are defined accurately. Misaligned source control points can result in the design of unclosed slat shapes (see Appendix I-2 for discussion of source branch cuts).

Step 6. The trailing-edge stagnation point should be located very near the end of the singularity line and very slightly below its extension. The separation distance should be on the order of .01 percent of the slat chord. Note that this minute distance places a very tight tolerance on the location of the source control points.

Step 7. The final parameter to be specified is the location of the slat leading-edge stagnation point. Two items are of importance here. First, the larger the distance from the stagnation point to the singularity line, the "fatter" the slat. Second, the slat camber can be increased (hence the load it carries) by moving the stagnation point normal to the singularity line. With these guidelines, a set of candidate leading-edge locations should be selected.

Step 8. Approximate slat shapes for each of the leading-edge locations are obtained next. Two points should be selected near the trailing edge, one slightly above and one slightly below. From these two points, streamlines can be traced upstream to the leading-edge region. If reasonable starting locations are selected, the streamlines will provide an approximation of the slat shape. Although the representation in the nose region is usually poor, the general shape is sufficiently accurate to select a good leading-edge location. Airfoil pressures should also be examined at this point.

<u>Step 9</u>. For the selected configuration, i.e., leading-edge location, the program is rerun, requesting the slat shape to be computed. If the slat shape routine is unable to converge on the proper shape, an alternate starting solution should be provided. As a general rule the starting solution should lie well outside (see sketch below) the actual slat, especially in the leading- and trailing-edge regions.



5. CONCLUDING REMARKS

This study has proven the feasibility of an inverse design method based on the method of distributed singularities. The method has been demonstrated to be workable and able to produce reasonable design solutions. Such solutions can be obtained with minimum input requirements but with a good degree of design flexibility.

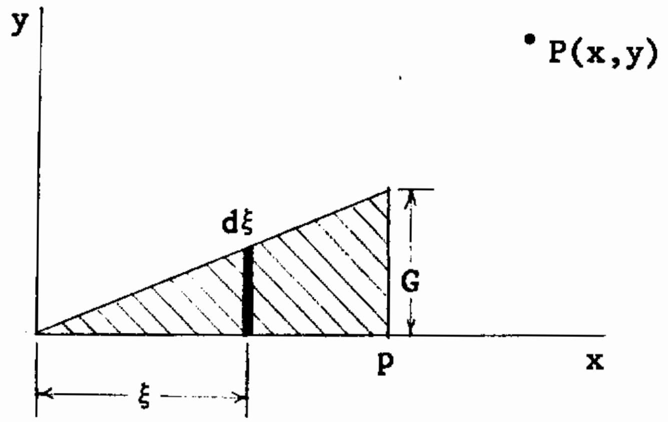
Also, the utility of this type of approach is enhanced by the possibilities of further developments. A relatively simple improvement would be to allow the airfoil to have trailing-edge flaps. Ground effects and blunt-trailing-edge effects could also be included without much difficulty. A somewhat more extensive extension would allow the simultaneous design of both the leading-edge slats and an arbitrary number of trailing-edge flaps.

APPENDIX I

DERIVATION OF POTENTIAL-FLOW EQUATIONS

I-1 Flow Field Induced by Distributed Vorticity

Consider the stream function produced by a line vortex distribution whose strength increases linearly from zero. The stream function at any arbitrary point P at (x,y) is given by



$$\psi = \int_0^p \left\{ -\frac{1}{4\pi} \text{ Ln} \left[(x-\xi)^2 + y^2 \right] \right\} \left\{ (\xi/p) G \, d\xi \right\} \qquad \text{(I-1)}$$

where G is the maximum vortex strength of the distribution. Equation I-1 can be integrated by making use of the formulas

$$\int z \text{ Ln } (z^2 + y^2) dz = \frac{z^2+y^2}{2} \text{ Ln } (z^2 + y^2) - \frac{1}{2} z^2 \qquad \text{(I-2)}$$

$$\int \text{Ln } (z^2 + y^2) dz = z\text{Ln } (z^2 + y^2) + 2y \tan^{-1}(\frac{z}{y}) - 2z \qquad \text{(I-3)}$$

The integration of Equation I-1 gives

$$\psi = \frac{G}{8\pi p} \left\{ (x^2 - y^2)L + 4yxT - p^2 \text{ Ln } \left[(x-p)^2 + y^2 \right] \right\} \qquad \text{(I-4)}$$

where

$$L = Ln\left[\frac{(x-p)^2 + y^2}{x^2 + y^2}\right] \tag{I-5}$$

and

$$T = \tan^{-1}\frac{x-p}{y} - \tan^{-1}\frac{x}{y} \tag{I-6}$$

Equation I-6 can be rewritten as

$$T = \tan^{-1}\left\{\tan\left[\tan^{-1}\frac{x-p}{y} - \tan^{-1}\frac{x}{y}\right]\right\}$$

Then,

$$T = -\tan^{-1}\frac{py}{y^2 + x(x-p)} \tag{I-7}$$

Actually, either Equation I-6 or I-7 can be used, but Equation I-7 is more useful for computer work.

The velocity induced by a vortex sheet can be derived by differencing Equation I-4. Recall that

$$u = \frac{\partial\psi}{\partial y} \qquad v = -\frac{\partial\psi}{\partial x} \tag{I-8}$$

Thus,

$$u = \frac{G}{2\pi p}\left(xT - \frac{1}{2}yL\right) \tag{I-9}$$

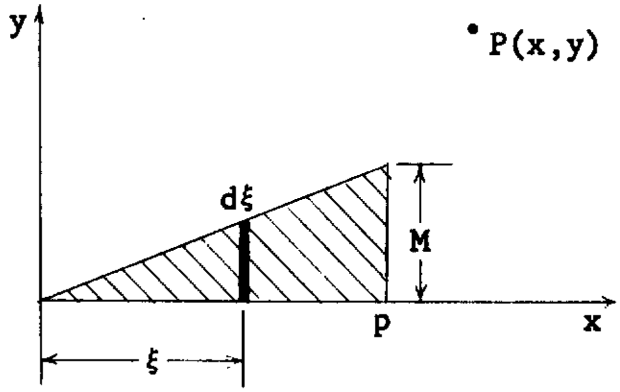
and

$$v = \frac{-G}{2\pi p}\left(\frac{1}{2}xL + yT + p\right) \tag{I-10}$$

where L and T are the same as in Equation I-4.

I-2 Flow Field Induced by Distributed Sources

The derivation for the flow induced by distributed sources follows the same pattern as the vortex derivation.



The stream function at any arbitrary point P at (x,y) is given by

$$\psi = \int_0^p \left\{ -\frac{1}{2\pi} \tan^{-1} \frac{x-\xi}{y} \right\} \left\{ (\xi/p) M \, d\xi \right\} \tag{I-11}$$

where M is the maximum source strength of the distribution. Equation I-11 can be integrated by making use of the formulas

$$\int z \tan^{-1} \frac{z}{y} \, dz = \frac{1}{2}(z^2 + y^2) \tan^{-1} \frac{z}{y} - \frac{1}{2} yz \tag{I-12}$$

$$\int \tan^{-1} \frac{z}{y} \, dz = z \tan^{-1} \frac{z}{y} - \frac{1}{2} y \, Ln(z^2 + y^2) \tag{I-13}$$

The integration of Equation I-11 gives

$$\psi = -\frac{M}{4\pi p} \left\{ (y^2 - x^2)T + yp(L+1) + p^2 \tan^{-1} \frac{x-p}{y} \right\} \tag{I-14}$$

The last term in this equation requires special attention. This term introduces a branch cut into the flow field due to the multi-valued nature of the inverse tangent function. The branch cut is the semi-infinite line defined by ($x > 0$, $y = 0$). Along this line, the term approaches zero; just below the line, the term approaches 2π. Because of this discontinuity, special care must be exercised when comparing the stream functions at two points in the presence of multiple source distributions. Both observation points must have the same relative orientation with respect to the branch cuts. For example, they must all be located above all of the branch cuts or they must all be located below the cuts. Otherwise, the numerical value of the difference between the stream functions at the two points will be incorrect.

The velocity field is given by

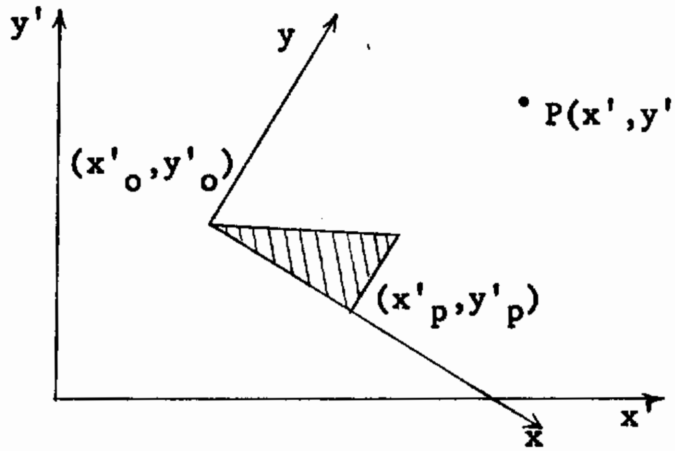
$$u = \frac{\partial \psi}{\partial y} = -\frac{M}{2\pi p}\left[yT + \frac{1}{2}xL + p\right] \qquad \text{(I-15)}$$

and

$$v = -\frac{\partial \psi}{\partial x} = -\frac{M}{2\pi p}\left[xT - \frac{1}{2}yL\right] \qquad \text{(I-16)}$$

-3 Coordinate Transformation

In general the singularity line is not aligned with the coordinate system as shown above. The line must be rotated and translated into that orientation. Consider a singularity line with an arbitrary orientation as shown below. For this case, the coordinates of the singularity



line and observation point are known with respect to the primed coordinate system. The problem is to determine the coordinates of the observation point in the unprimed coordinate system. The proper transformation is

$$\begin{aligned} x &= (x' - x'_p)\cos\phi + (y' - y'_p)\sin\phi \\ y &= -(x' - x'_p)\sin\phi + (y' - y'_p)\cos\phi \end{aligned} \qquad \text{(I-17)}$$

where

$$\tan\phi = \frac{y'_o - y'_p}{x'_o - x'_p} \qquad \text{(I-18)}$$

The application of the transformation, I-17, permits the flow-field functions defined in the earlier discussion to be applied for general singularity arrangements.

APPENDIX II

COMPUTER PROGRAM DESCRIPTION

II-1 Procedure Description

The procedure XOV is programmed in FORTRAN Extended Language for operation on a CDC 6600 or a compatible model and requires only the standard CDC-supplied operating system and utility programs. The main program XOV00 directs the calling of four overlays. In each overlay there is a program XOVyz where y is the primary level number and z is the secondary level number of the overlay. These programs bring the desired subroutines into the core and direct the calling of the subroutines to make the necessary computations. Figure 10 shows the important subroutines called from these programs, while Figure 11 shows the flow of information from the customers viewpoint.

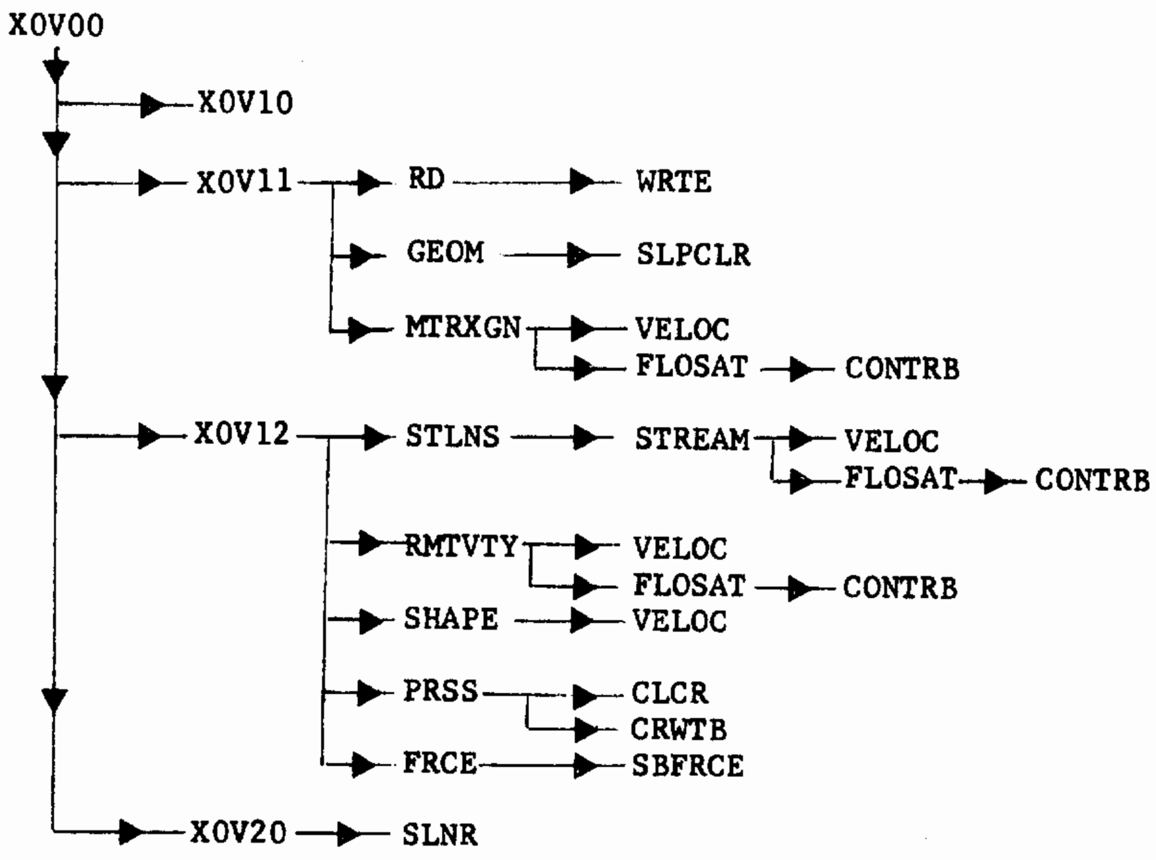


Figure 10 Flow Schematic of Procedure XOV

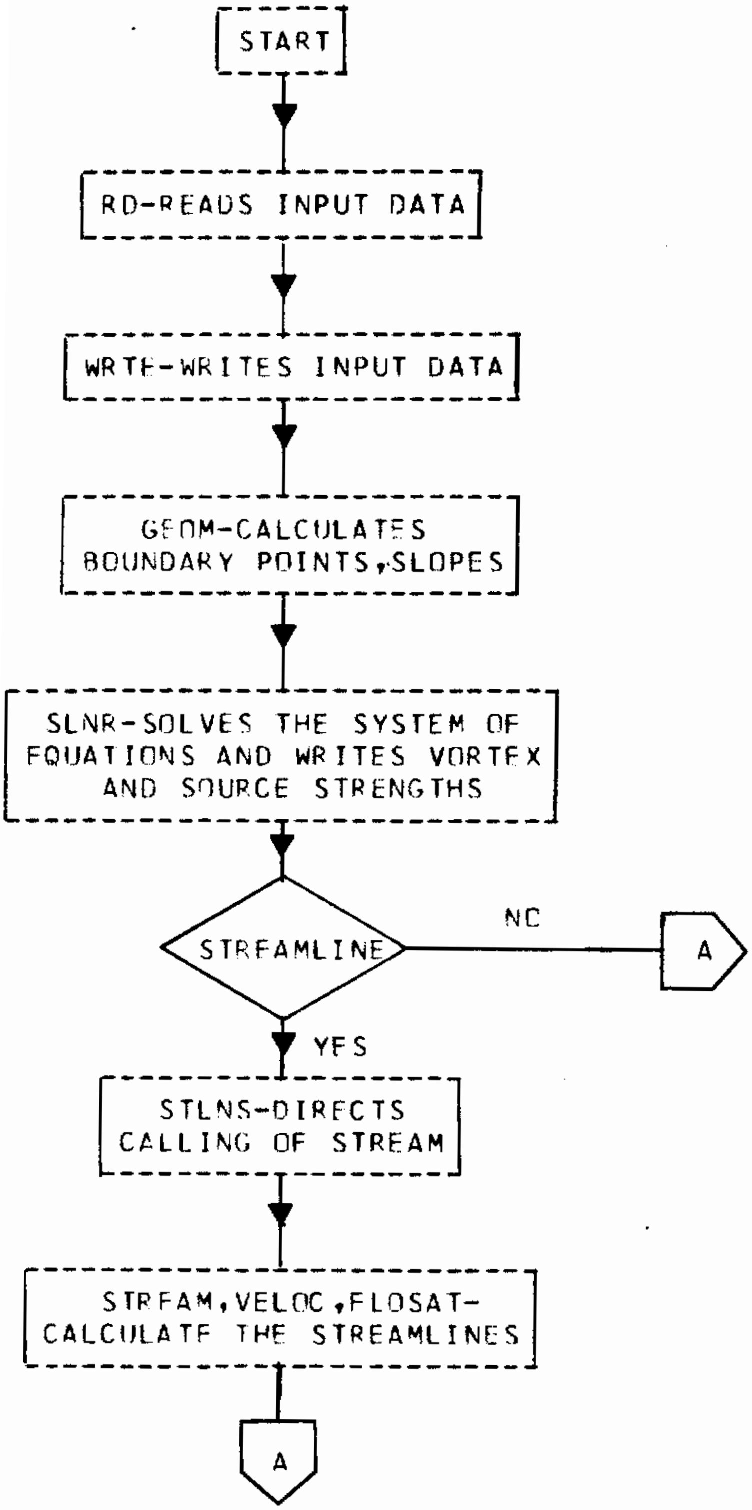


Figure 11 Flow Chart for Procedure XOV

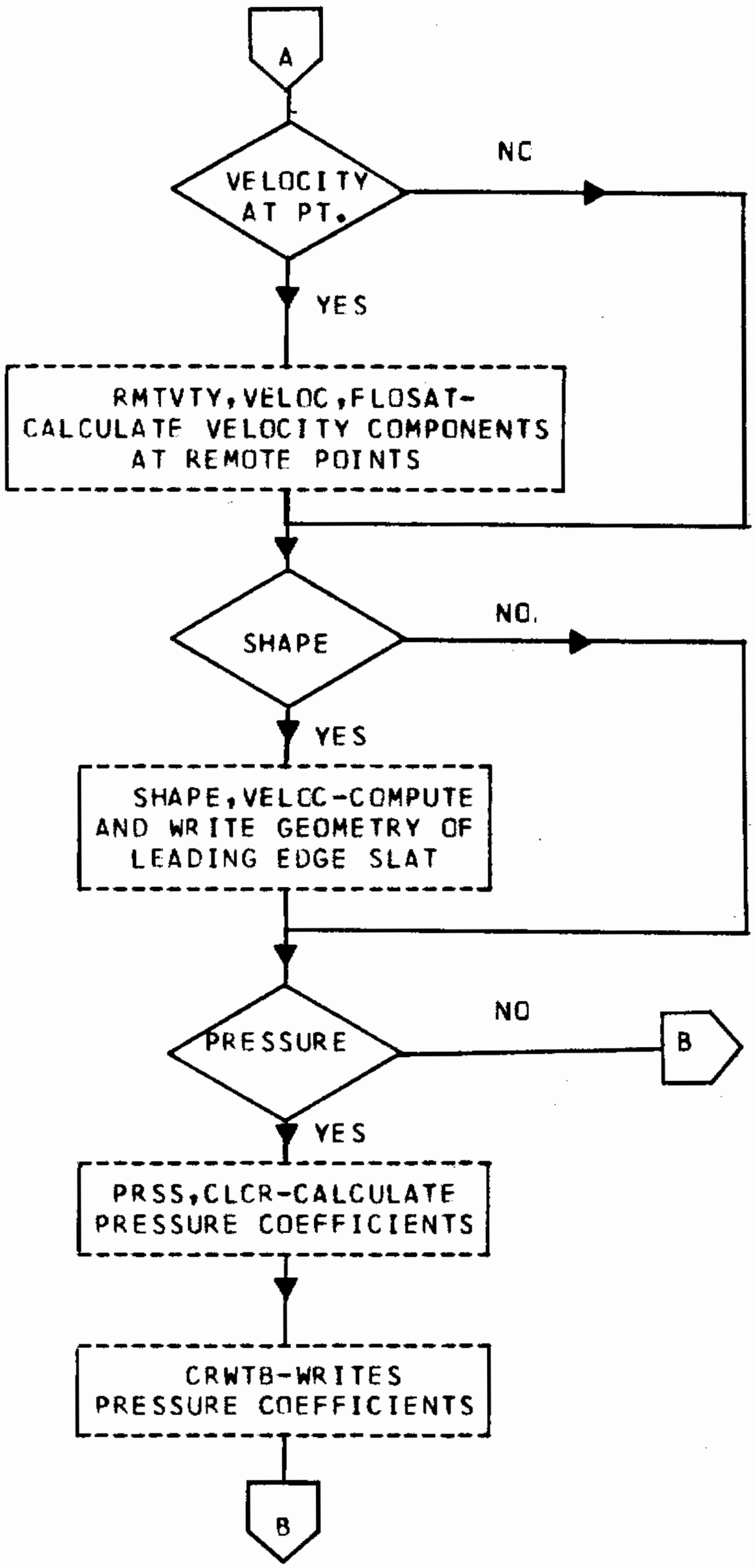


Figure 11 Continued

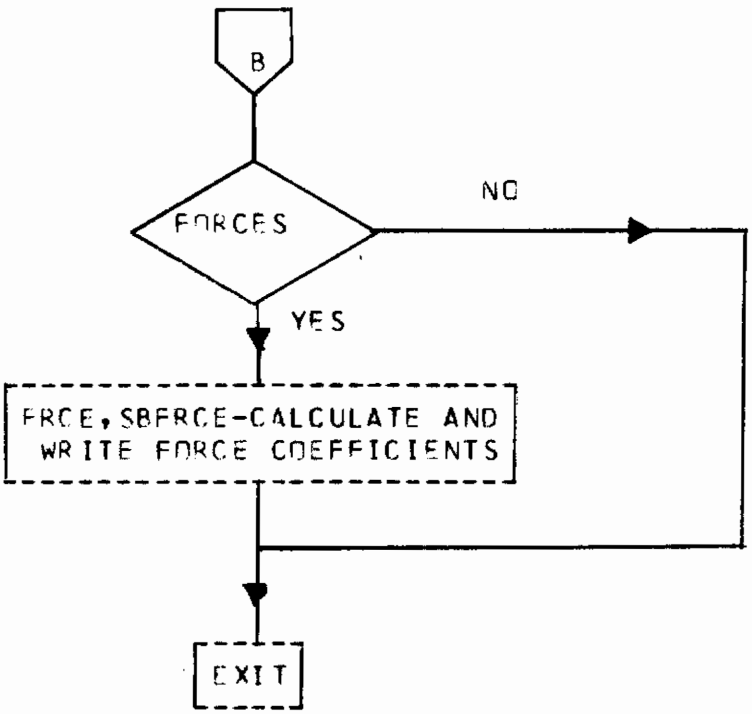


Figure 11 Continued

Subroutine SHAPE

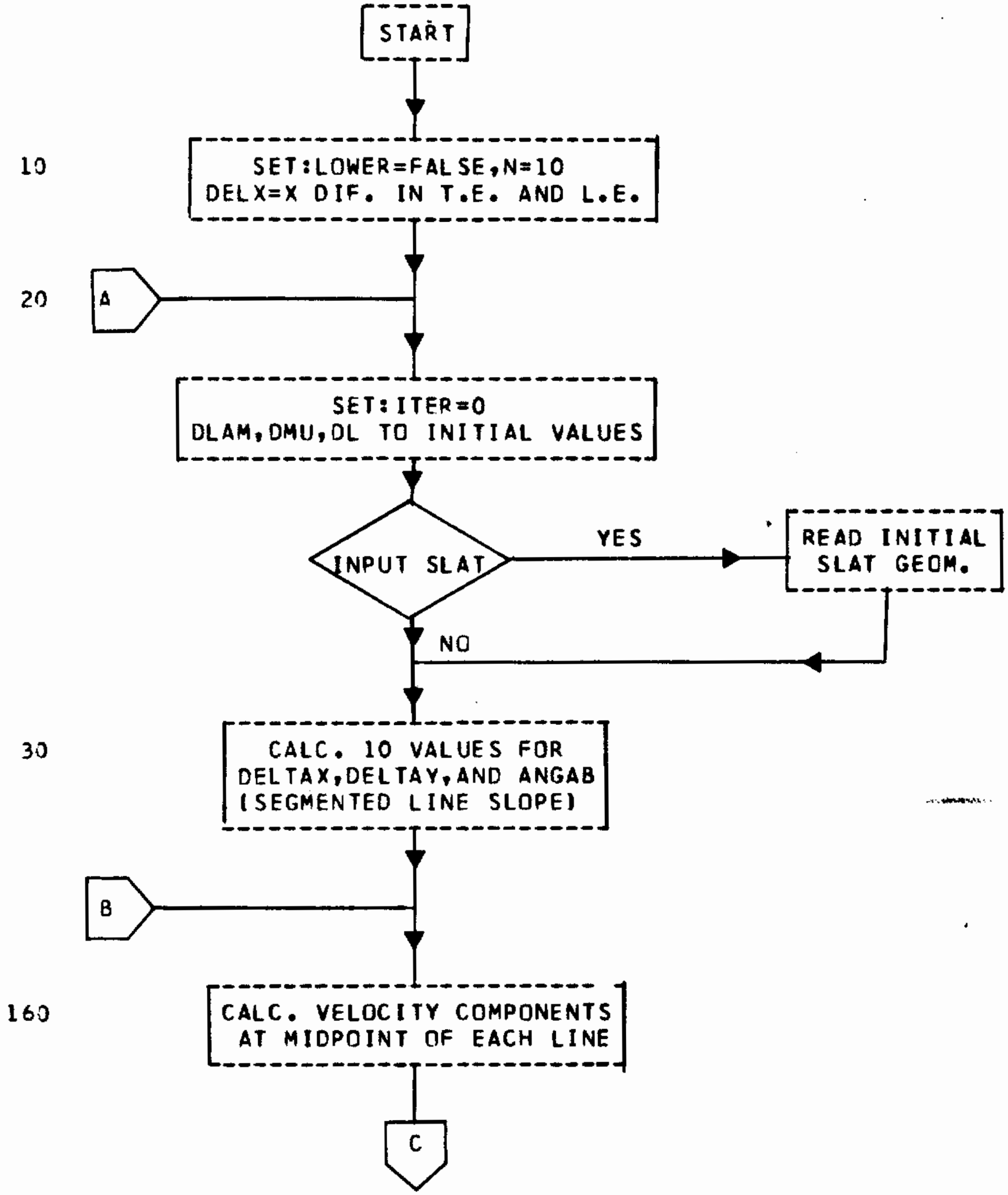


Figure 11 Continued

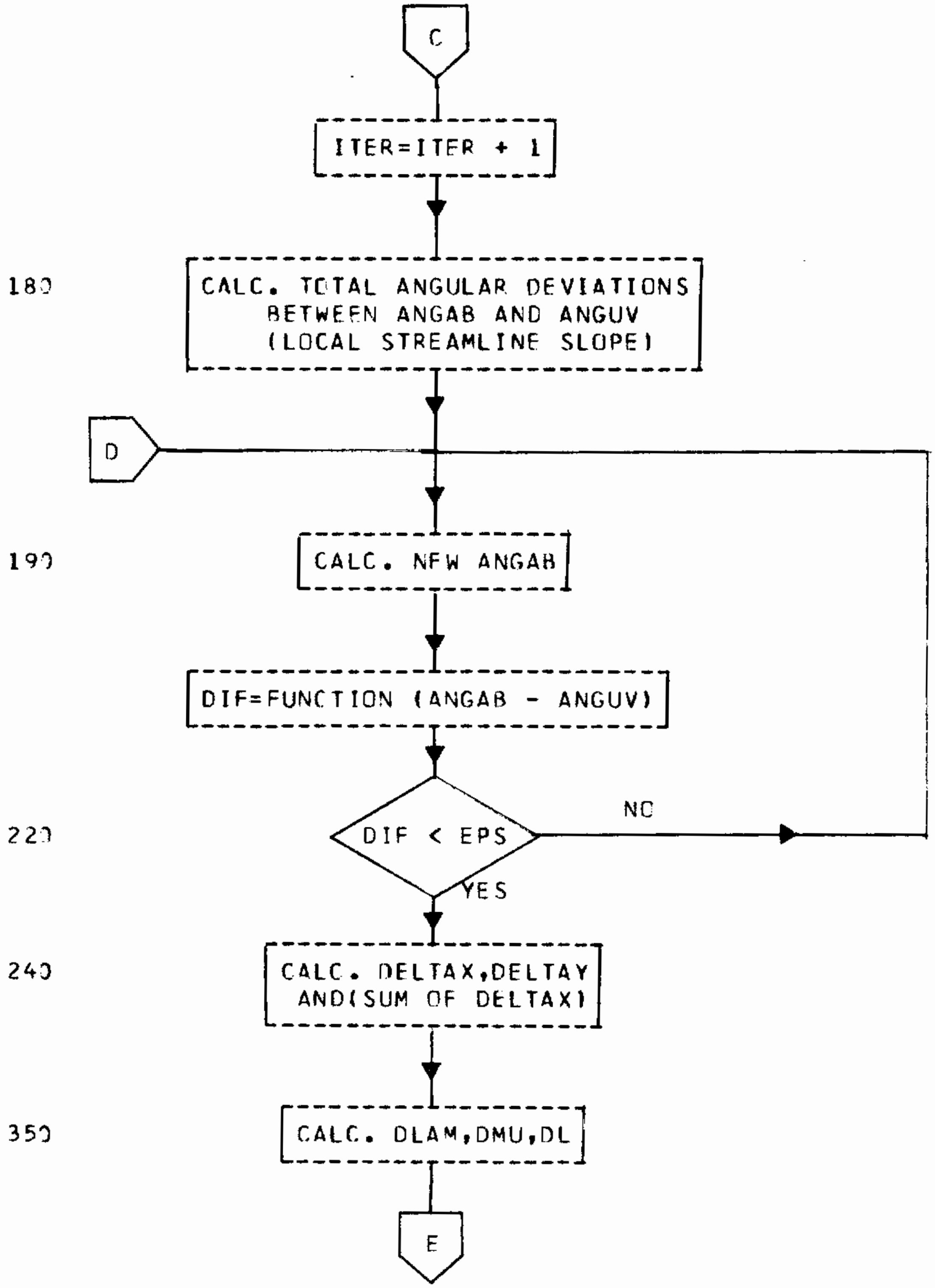


Figure 11 Continued

360

100

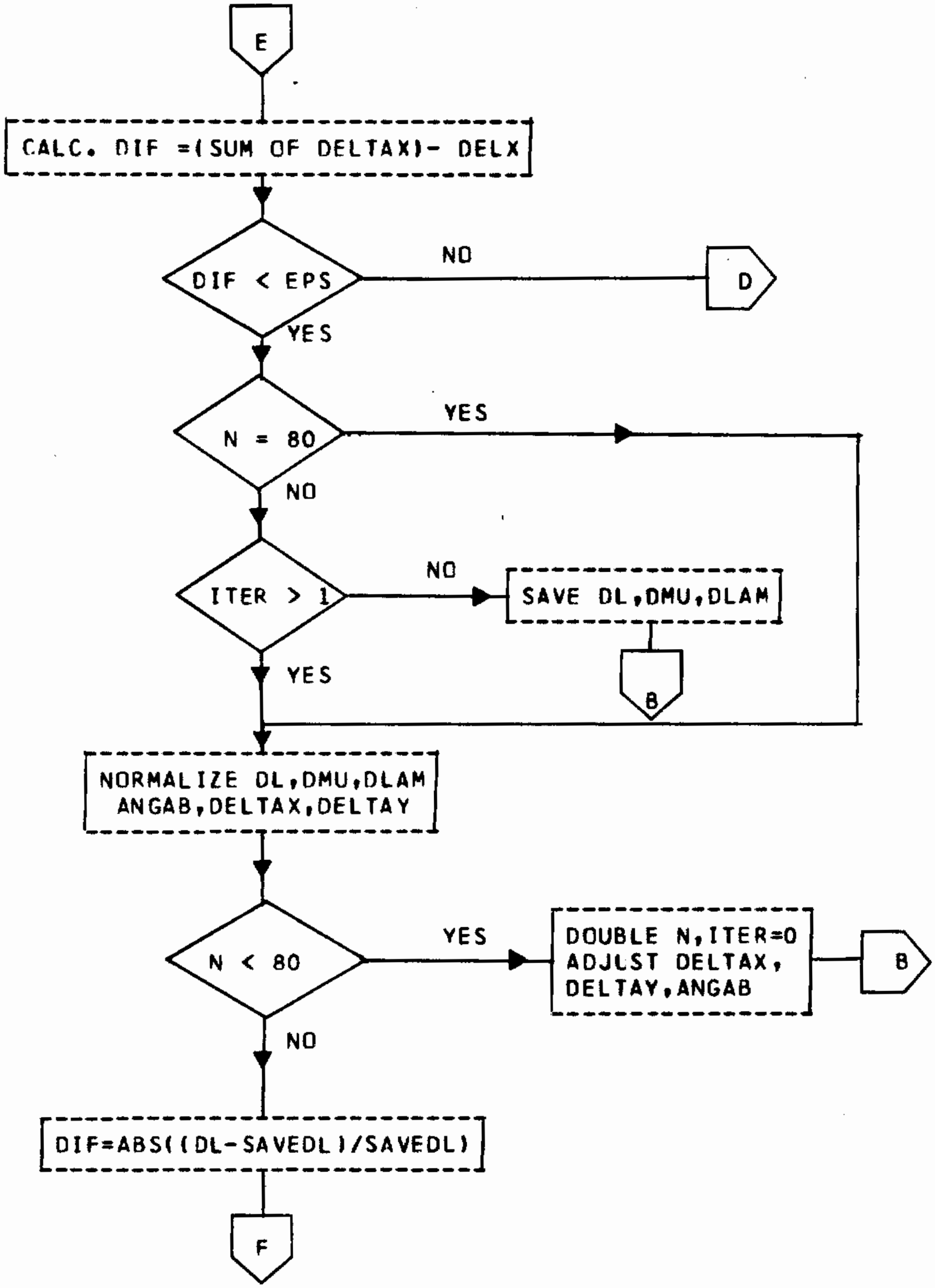


Figure 11 Continued

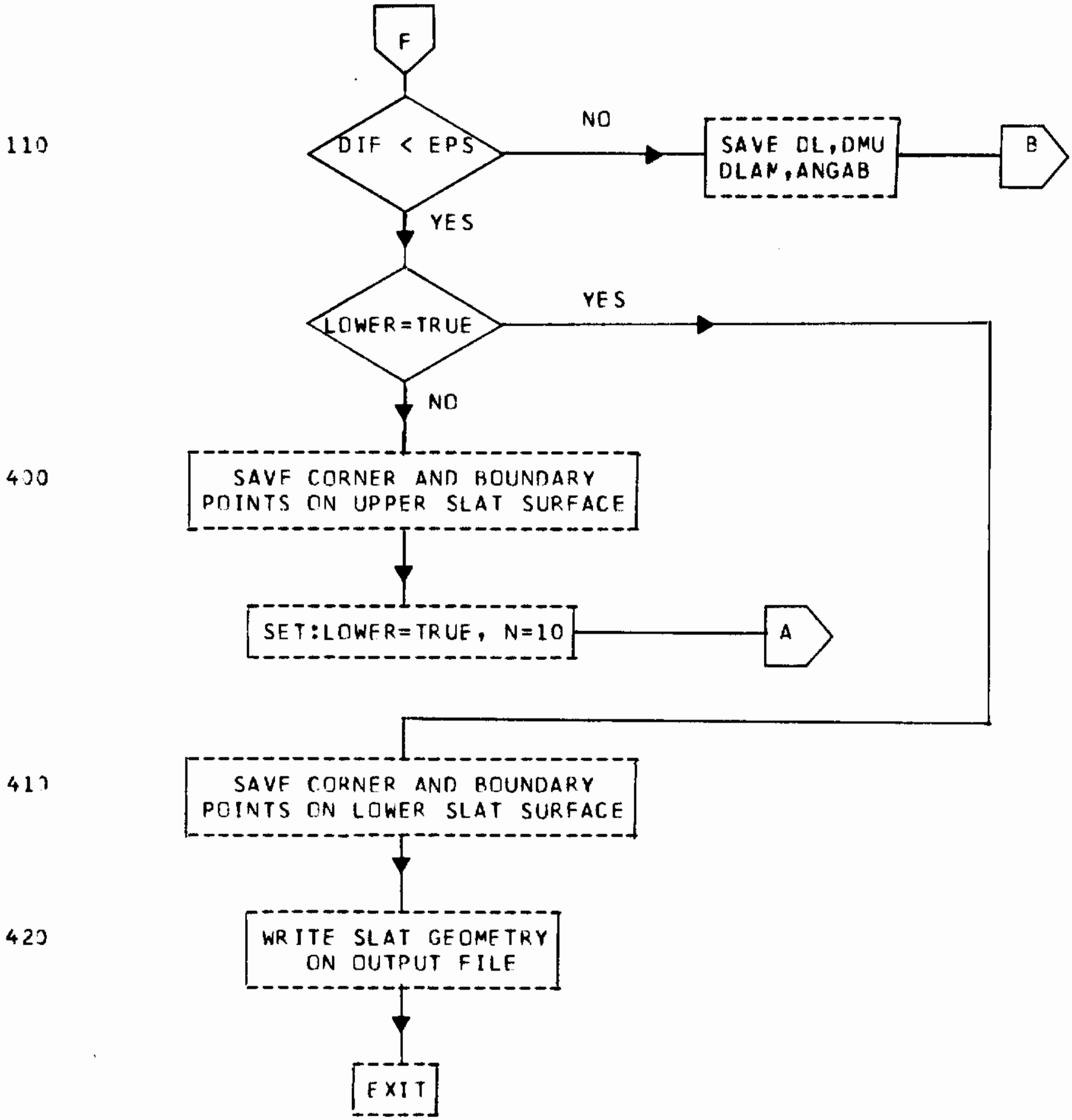


Figure 11 Concluded

II-2 I/O Requirements

Card input occurs on file name TAPE5, and data output occurs on file name TAPE6. In addition to these files there are two more file names, as listed below.

Subroutine	File Name	Operation
MTRXGN	TAPE1	Writes U, V velocity components
	TAPE9	Writes linear equation coefficients
PRSS	TAPE1	Reads U, V velocity components
SLNR	TAPE9	Reads linear equation coefficients

II-3 Deck Structure

The deck makeup for a run consists of a Job card, INPUT. card, 7/8/9 card, binary object decks as listed below, 7/8/9 card, problem data deck, and 6/7/8/9 card.

DECK STRUCTURE FOR OVERLAYS

```
OVERLAY(DUM,0,0)
    PROGRAM X0V00
        SUBROUTINE KEOF
OVERLAY(DUM,1,0)
    PROGRAM X0V10
        SUBROUTINE CONTRB
        SUBROUTINE DSTRBD
        SUBROUTINE DSTSQD
        SUBROUTINE FLOSAT
        SUBROUTINE VELOC
OVERLAY(DUM,1,1)
    PROGRAM X0V11
        SUBROUTINE GEOM
        SUBROUTINE MTRXGN
        SUBROUTINE RD
        SUBROUTINE SLPCLR
        SUBROUTINE WRTE
OVERLAY(DUM,1,2)
    PROGRAM X0V12
        SUBROUTINE CLCR
        SUBROUTINE CRWTB
        SUBROUTINE FRCE
        SUBROUTINE PRSS
        SUBROUTINE RMTVTY
        SUBROUTINE SBFRCE
        SUBROUTINE SHAPE
        SUBROUTINE STLNS
        SUBROUTINE STREAM
OVERLAY(DUM,2,0)
    PROGRAM X0V20
        SUBROUTINE SLNR
```

II-4 Function of Subroutines and Programs

SUBROUTINE CLCR calculates the incompressible pressure coefficients at any boundary point after the velocity components have been calculated.

SUBROUTINE CONTRB calculates the portion of the stream function induced by a single distributed singularity.

SUBROUTINE CRWTB writes the pressure coefficients on the output file.

SUBROUTINE DSTRBD calculates the components of velocity induced by a triangular distribution of vorticity at any point.

SUBROUTINE DSTSQD calculates the distance between two specified points.

SUBROUTINE FLOSAT directs the computation of the stream function at any point by subroutine CONTRB.

SUBROUTINE FRCE writes the force coefficients, drag coefficient and pitching moment coefficient.

SUBROUTINE GEOM calculates and writes the boundary point coordinates and the local surface slope associated with each boundary point.

SUBROUTINE KEOF writes the location of an unexpected end of file during a read operation.

SUBROUTINE MTRXGN generates the influence coefficient matrix to be solved.

SUBROUTINE PRSS directs the calculation and writing of the pressure coefficients.

SUBROUTINE RD reads the input data and checks for input errors.

SUBROUTINE RMTVTY directs the calculation and writing of the velocity at the remote points.

SUBROUTINE SBFRCE computes the force coefficients.

SUBROUTINE SHAPE computes and writes the geometry of the leading-edge slat.

SUBROUTINE SLNR solves the system of equations for the vortex and source strengths.

SUBROUTINE SLPCLR calculates an angle by the use of the arctangent function.

SUBROUTINE STLNS initializes the calculation of the streamlines.

SUBROUTINE STREAM calculates the streamlines for the airfoil system.

SUBROUTINE VELOC computes velocities induced at a point by singularities at the control points.

SUBROUTINE WRTE writes the input data.

PROGRAM XOV00 directs the calling of the overlays.

PROGRAM XOV10 permits the subroutines CONTRB, DSTRBD, DSTSQD, FLOSAT, and VELOC to be in the overlay (1,0).

PROGRAM XOV11 directs the calling of subroutines RD, GEOM, and MTRXGN.

PROGRAM XOV12 controls the calling of subroutines FRCE, PRSS, RMTVTY, SHAPE, and STLNS.

PROGRAM XOV20 directs the calling of subroutine SLNR.

II-5 Definition of FORTRAN Variables

COMMON BLKA

IND(30)	Option indicators, input
NCPTS	Number of control points
NSTLNS	Number of streamlines, input
NVLPTS	Number of remote points at which the velocity is required, input
NPNTS	Number of control points plus one
IB	Number of control points minus one

COMMON BLKB

XLE	X coordinate of main airfoil leading edge, input
YLE	Y coordinate of main airfoil leading edge, input
XTE	X coordinate of main airfoil trailing edge, input
YTE	Y coordinate of main airfoil trailing edge, input
XMBAR	X coordinate about which moments are calculated, input
YMBAR	Y coordinate about which moments are calculated, input
USUBI	Freestream velocity
CREF	Reference chord length
XSUBC	X value to which streamlines are calculated, input
DLTLMN	Minimum distance between two consecutive points on a streamline, input
DLTALN	Nominal starting distance between two consecutive points on a streamline, input
DLTALM	Maximum distance between two consecutive points on a streamline, input
TOLLMT	Tolerance used in calculating streamlines, input
CRT	Maximum number of points on a streamline, input
CRFSQ	Square of reference chord length

COMMON BLKC

ALPHA	Angle of attack in degrees, input
XS(10)	X coordinates of initial points on streamlines, input
YS(10)	Y coordinates of initial points on streamlines, input
XV(100)	X coordinates of remote points at which velocity is to be calculated, input
YV(100)	Y coordinates of remote points at which velocity is to be calculated, input
X(200)	X coordinates of main airfoil corner points, input
Y(200)	Y coordinates of main airfoil corner points, input

COMMON BLKD

CS	Cosine of the angle of attack
SN	Sine of the angle of attack
XBP(200)	X coordinates of the boundary points
YBP(200)	Y coordinates of the boundary points
COSSLP(200)	Cosine of the slope angle associated with a boundary point on main airfoil
SINSLP(200)	Sine of the slope angle associated with a boundary point on main airfoil

COMMON BLKE

U(200)	Velocity component in the X-direction
V(200)	Velocity component in the Y-direction
CPUP(200)	Pressure coefficients on upper side of airfoil vortex sheet

CPLR(200)	Pressure coefficients on lower side of airfoil vortex sheet
CPM(200)	Pressure coefficients on upper side of airfoil vortex sheet

<u>COMMON BLKG</u>

GAMMA(200)	Vortex and source strengths

<u>COMMON BLKJ</u>

TITLE(12)	Title of problem, input

<u>BLOCK BLKK</u>

SCOSLP(160)	Cosine of the slope angle associated with a boundary point on slat
CPSLAT(160)	Pressure coefficient on slat
SSINLP(160)	Sine of the slope angle associated with a boundary point on slat
XSLAT(161)	X-coordinate of corner point on slat
XXBP(160)	X-coordinate of boundary point on slat
YSLAT(161)	Y-coordinate of corner point on slat
YYBP(160)	Y-coordinate of boundary point on slat

<u>COMMON BLK1</u>

NVOR	Number of vortex control points on slat, input
NSOR	Number of source control points on slat, input
MCPS	Index of first pressure or singularity strength to be specified, input
MCPF	Index of final pressure or singularity strength to be specified

NMOD	Number of pressures or singularity strengths to be specified
NPTS	Number of control points on main airfoil, input
NB	Number of corner points on main airfoil minus one
NCTS	Number of corner points on main airfoil plus one
NSOR1	First source control point index
NVORT	Total number of vortex control points

<u>COMMON BLK2</u>

CPMOD(50)	If B1 = 0, slat singularity strength array, input If B1 = 1, pressure coefficient array, input

<u>COMMON BLK3</u>

XSLE	X coordinate of slat leading edge, input
YSLE	Y coordinate of slat leading edge, input
XSTE	X coordinate of slat trailing edge, input
YSTE	Y coordinate of slat trailing edge, input

APPENDIX III

PROGRAM UTILIZATION

III-1 Input

Card input is classified into three sections. The first section consists of the title for the problem and has two cards with alphanumeric information. The second section consists of the option indicators and problem dimensions and has three cards with integers. The last section has a variable number of cards with floating-point constants.

The section requiring integer input has a maximum of 15 input items per card. Columns 1 to 3 contain the first input item, columns 4 to 6 contain the second input item, ..., and columns 43-45 contain the 15th input item, if required. The numbers should be right-adjusted in each field since leading, trailing, and embedded blanks are treated as zeros. If additional input items are required, the 16th item appears in columns 1 to 3 on the next card with the additional items in the following three column fields.

The cards requiring floating-point constants have a maximum of six items per card. Each input item has a field length of 10 columns and must be right-adjusted or accompanied by a decimal point. If the decimal point is omitted, the decimal is assumed after the digit in the 10th column. The first input item is in columns 1 to 10; the second input item, if required, is in columns 11 to 20; ...; and the sixth input item, if required, is in columns 51-60. If more input items for a card type are required, the input items appear on additional cards.

III-1.1 Card Description

A problem deck should be assembled in the order given below.

A. Title Cards (2 required)

These cards contain the title in columns 1-60. Any combination of alphanumeric characters may be used (FORMAT 6A10/6A10)

B. <u>Option Indicator Cards</u> (2 required)

These cards have the following input items on Card 1:

B1, B2, B3, ..., B14, B15

and on Card 2:

B16, B17, ..., B30

with a FORMAT of 15I3/15I3, where

B1	= 0	Slat singularity strength input
	= 1	Airfoil pressure input
B2	= 0	Slat included and shape calculated
	= 1	Slat included and shape not calculated
B3	= 0	Not in use in present version
B4	= 0	Slat design option
	= 1	Single-element airfoil solution, used in defining a slat singularity streamline
B5	= 0	Airfoil pressures calculated
	= 1	Airfoil pressures not calculated
B6	= 0	Forces calculated with detailed output
	= 1	Forces calculated without detailed output
	= 2	Forces not calculated
B7	= 0	Not in use in present version
B8	= 0	No output of intermediate calculation
	= 1	Output of intermediate calculations, used in program checkout
B9	= 0	Program calculates initial slat shape
	= 1	Input of initial slat shape
B10	= 0	Not in use in present version
⋮		
B30	= 0	Not in use in present version

C. Problem Dimension Card (1 required)

This card has the following input items:

C1, C2, C3, C4, C5, C6 with a FORMAT of 6I3

where

C1 = Number of corner points on main airfoil, $C1 > 0$

C2 = Number of vortex control points on slat
If B4=0, C2 must be positive
If B4=1, set C2 to zero

C3 = Number of source control points on slat,
If B4=0, $C3 \geq 2$
If B4=1, set C3 to zero
$(C1+C2+C3) \leq 200$, $(C2+C3) \geq 7$

C4 = If B4=0 and if B1=0, the index of first control point whose strength is to be specified, $C4 > C1$
If B4=0 and if B1=1, the index of first airfoil boundary point at which pressure is to be specified, $C4 \leq C1$
If B4=1, set C4 to zero

C5 = Number of streamlines, $0 \leq C5 \leq 10$

C6 = Number of remote points at which the velocity is required, $0 \leq C6 \leq 100$

NOTE: The singularity control points for the slat are indexed in sequence with the airfoil singularities. The slat source control points follow, in sequence, the slat vortex control points, which follow the airfoil corner points. Boundary points are the midpoints of the lines connecting two adjacent corner points. They have the same index as that of the lower (numerically) corner point index.

D. Moments Card (1 Required)

This card has the following input items:

D1, D2 with a FORMAT of 2F10.0, where

D1 = X coordinate of the point about which moments are calculated

D2 = Y coordinate of the point about which moments are calculated

E. <u>Velocity and Edge Cards</u> (2 required)

These cards have the following input items on Card 1:

E1, E2, E3, E4, E5

and on Card 2:

E6, E7, E8, E9

with a FORMAT of 5F10.0/4F10.0, where

E1	=	Freestream velocity
	= 0.0	Freestream velocity is set equal to 1.0
E2	=	X-coordinate of airfoil leading edge
E3	=	Y-coordinate of airfoil leading edge
E4	=	X-coordinate of airfoil trailing edge
E5	=	Y-coordinate of airfoil trailing edge
E6	=	X-coordinate of slat leading edge
E7	=	Y-coordinate of slat leading edge
E8	=	X-coordinate of slat trailing edge
E9	=	Y-coordinate of slat trailing edge

F. <u>Incidence Card</u> (1 required)

This card has the following input item:

F1 with a FORMAT of 1F10.0 where

F1 = Angle of attack in degrees

G. <u>Airfoil Corner Point Cards</u> (C1/3 required)

These cards have the following input items:

G1, G2, G3, G4, ..., GA, with a FORMAT of 6F10.0 where

G1 = X-coordinate of corner point 1

G2 = Y-coordinate of corner point 1

G3 = X-coordinate of corner point 2

G4 = Y-coordinate of corner point 2

⋮

GA = Y-coordinate of corner point B with B=C1 and A=2*C1

The corner points should be listed starting from the trailing edge along the top surface to the leading edge and then back to the trailing edge along the lower surface. The trailing-edge point on the upper surface is always the first point listed, and the lower trailing-edge point is the last point listed. These points must be coincident.

H. <u>Slat Vortex Control Point Cards</u> (0 or C2/3 required)

These cards are required when B4 is zero and have the following input items:

H1, H2, H3, ..., HA with a FORMAT of 6F10.0 where

H1 = X-coordinate of slat vortex control point 1

H2 = Y-coordinate of slat vortex control point 1

H3 = X-coordinate of slat vortex control point 2

H4 = Y-coordinate of slat vortex control point 2

⋮

HA = Y-coordinate of slat vortex control point B with B=C2 and A=2*C2

The slat vortex control points should be listed in sequence starting with the point nearest the trailing edge and ending with the point nearest the leading edge. The total number of slat vortex control points is given by C2.

I. <u>Slat Source Control Point Cards</u> (0 or C3/3 required)

These cards are required when B4 is zero and have the following input items:

I1, I2, I3, I4, ..., IA with a FORMAT of 6F10.0 where

I1 = X-coordinate of slat source control point 1

I2 = Y-coordinate of slat source control point 1

I3 = X-coordinate of slat source control point 2

I4 = Y-coordinate of slat source control point 2

⋮

IA = Y-coordinate of slat source control point B with B=C3 and A=2*C3

The slat vortex control points should be listed in sequence starting with the point nearest the trailing edge and ending with the point nearest the leading edge. The total number of slat source control points is given by C3.

J. <u>Specified Singularity Strength Cards</u> (0 or (C2+C3-6)/3 required)

These cards are required when B1 is zero and B4 is zero and have the following input items:

J1, J2, ..., JA with a FORMAT of 6F10.0 where

J1 = Singularity strength at slat control point index C4

J2 = Singularity strength at slat control point index C4+1

⋮

JA = Singularity strength at slat control point with an index of C4+(A-1) where A=C2+C3-6

The singularity strengths should be listed in sequence (ascending order) starting with the point specified by C4. The total number of singularity strengths to be specified is given by (C2+C3-6), and the number must not exceed 50.

K. Specified Pressure Coefficient Cards (0 or (C2+C3-6)/3 required)

These cards are given when B1 is one and B4 is zero and have the following input items:

K1, K2, ..., KA with a FORMAT of 6F10.0 where

K1 = Pressure coefficient at boundary point index C4

K2 = Pressure coefficient at boundary point index C4+1

⋮

KA = Pressure coefficient at boundary point index C4+(A-1) with A=C2+C3-6

The pressure coefficients should be listed in sequence (ascending order) starting with the boundary point specified by C4. The total number of pressures coefficients to be specified is given by (C2+C3-6), and this number must not exceed 50.

L. Streamline Card (0 or 1 required)

This card is required only when C5≠0 and has the following input items:

L1, L2, L3, L4, L5, L6 with a FORMAT of 6F10.0 where

L1 = Maximum X value to which the streamlines are calculated; a typical value is two chord lengths downstream of the trailing edge.

L2 = Minimum distance between two consecutive points on a streamline; a typical value is 10^{-3} times the reference chord.

L3 = Nominal value which is the starting distance between two consecutive points on a streamline; a typical value is 10^{-2} times reference chord.

L4 = Maximum distance between two consecutive points on a streamline; a typical value is 10^{-1} times reference chord.

L5 = Maximum error allowed in calculating streamlines; a typical value is 10^{-5} times the reference chord.

L6 = Maximum number of points on each streamline, typically 500.

$L2 < L3 < L4$

M. Streamline Coordinate Cards (0 or C5/3 required)

These cards are required only when C5≠0 and have the following input items:

M1, M2, M3, M4, ..., MA with a FORMAT of 6F10.0 where

M1 = X-coordinate of the starting point for the first streamline

M2 = Y-coordinate of the starting point for the first streamline

M3 = X-coordinate of the starting point for the second streamline

M4 = Y-coordinate of the starting point for the second streamline

⋮

MA = Y-coordinate of the starting point for the Bth streamline with B=C5 and A=2*C5

N. Remote Points Cards (0 or C6/3 required)

These cards are required only when C6≠0 and have the following input items:

N1, N2, N3, N4, ..., NA with a FORMAT of 6F10.0 where

N1 = X-coordinate of remote point 1 at which the velocity is required

N2 = Y-coordinate of remote point 1 at which the velocity is required

N3 = X-coordinate of remote point 2 at which the velocity is required

N4 = Y-coordinate of remote point 2 at which the velocity is required

⋮

NA = Y-coordinate of remote point B at which the velocity is required with B=C6 and A=2*C6

O. Initial Slat Control Points Cards (0 or 6 required)

These cards are required when B9 is one and have the following input items:

O1, O2, O3, O4, ..., O17, O18, O19, O20, ..., O35, O36 with a FORMAT of 6F10.0 where

O1 = X-coordinate of second point on upper surface

O2 = Y-coordinate of second point on upper surface

03 = X-coordinate of third point on upper surface

04 = Y-coordinate of third point on upper surface

⋮

017 = X-coordinate of tenth point on upper surface

018 = Y-coordinate of tenth point on upper surface

019 = X-coordinate of second point on lower surface

020 = Y-coordinate of second point on lower surface

⋮

035 = X-coordinate of tenth point on lower surface

036 = Y-coordinate of tenth point on lower surface

Nine points must be listed between the leading and trailing edge, starting from the point adjacent to the leading edge and progressing along the upper surface to the point adjacent to the trailing edge. In a similar fashion for the lower surface, nine points should be listed, starting from the point adjacent to the leading edge. The nine points must not include either the trailing edge or the leading edge. The points should be selected so as to approximate a configuration larger than the expected slat configuration. With the leading edge as point 1 and the trailing edge as point 11, each of the nine points should be approximately equidistant from its adjacent points.

III-1.2 Restrictions

Maximum of 10 streamlines

Maximum of 100 points at which the velocity is calculated

Maximum of 50 values for specifying singularity strengths

Maximum of 50 values for specifying pressure coefficients

Maximum of 200 points for the sum of C1+C2+C3

where: C1 = number of corner points on main airfoil

C2 = number of vortex control points on slat

C3 = number of source control points on slat

III-2 Output Description

The output consists of the following items:

A listing of the options chosen, the angle of attack, the streamline starting coordinates, the point about which moments are taken, the airfoil and slat leading edges, the airfoil and slat trailing edges, the reference chord, the slat vortex and source control points, and the remote points at which the velocity is calculated.

For the airfoil, the set of corner points, the set of boundary points, and the slope at each boundary point.

If B8=1, the velocity components and the influence coefficients are printed for each boundary point.

A set of gammas for the vortex control points and, if B4=0, the slat source control points.

If B2=0, B4=0 and B8=1, the set of corner points for the slat and the velocity components at the boundary points during the intermediate calculation steps for the slat shape.

If B2=0 and B4=0, the corner points and pressure distribution for the slat.

For the airfoil, the pressure versus the boundary points, and the boundary points non-dimensionalized with respect to the reference chord.

If B6 is equal to 0 or 1, a listing of the pitching moment coefficient, the force coefficients normal and parallel to the reference chord obtained by integration of pressure distribution for the main airfoil and also for the slat if B2=0 and B4=0.

If B6 is equal to 0, a listing of the force coefficients normal and parallel to the reference chord and pitching moment coefficient for each segment of the surface lying between consecutive corner points for the main airfoil and also for the slat if B2=0 and B4=0.

If C5 is not equal to zero, a listing of the coordinates of the points defining the streamlines requested. Under the X-coordinate, the local flow angle is listed; under the Y-coordinate, the local pressure coefficient. The magnitude of the local stream function appears below the pressure coefficient.

If C6 is positive, a listing of the modulus, magnitude, and the components of the velocity vector at each remote point at which the velocity is calculated. Also listed are the non-dimensionalized coordinates of each remote point with respect to the reference chord. The magnitude of the local stream function is listed under the heading "FLOW".

III-3 Sample Test Problem

III-3.1 Input

The input for the sample test problem is presented on the following page to illustrate the method of card input:

- Cards 1 and 2 are the title cards.

- Cards 3 and 4 are the option indicator cards. Indicator 1 indicates slat singularity strength will be input. Indicator 2 indicates shape will be calculated. Indicator 4 indicates the slat configuration will be designed. Indicator 5 indicates airfoil pressures will be calculated. Indicator 6 indicates a detailed output of force calculations. Indicator 8 indicates an output of the intermediate calculations. Indicator 9 indicates the input of the initial slat shape.

```
                    ***    CARD INPUT    ***
1        10        20        30        40        50        60
 SIMPLE SYMMETRIC AIRFOIL AT 5 DEG. WITH SIMPLE SLAT           P020001
TRAILING-EDGE LOCATION -  BASELINE CASE                        P020002
  0  0  0  0  0  0  0  1  1  0                                 P020003
  0  0  0  0  0  0  0  0  0  0                                 P020004
 21  3  4 25  2  0                                             P020005
1.2      0.                                                    P020006
1.       0.        0.        1.2       0.                      P020007
-.06     .035      .02       .082                              P020008
5.                                                             P020009
1.2      0.        1.0       .02       .8        .05           P020010
0.6      .075      0.4       .09       .2        .082          P020011
0.1      .065      0.05      .048      .02       .031          P020012
0.0075   .018      0.        .0        .0075     -.018         P020013
0.02     -.031     0.05      -.048     .1        -.065         P020014
0.2      -.082     0.4       -.09      .6        -.075         P020015
0.8      -.05      1.0       -.02      1.2       +.0           P020016
.0198    .0824     -.0307    .0547     -.0461    .0454         P020017
.0198    .0824     -.0386    .0499     -.0461    .0454         P020018
-.0517   .0421                                                 P020019
-1.00                                                          P020020
.04      .0001     .002      .01       .00001    300.          P020021
-.0610   .0350     -.0600    .0340                             P020022
-.07     .0455     -.07      .0602     -.065     .0742         P020023
-.055    .085      -.045     .096      -.0309    .1000         P020024
-.0160   .098      -.0012    .096      +.014     .0955         P020025
-.060    .019      -.047     .010      -.0308    .010          P020026
-.0155   .015      -.C       .020      +.014     .030          P020027
+.020    .045      +.030     .0572     +.033     .073          P020028
1        10        20        30        40        50        60
```

- Card 5 indicates 21 corner points on main airfoil, with 3 vortex control points on slat, and 4 source control points on slat. Strength of control point 25 is the first to be specified. Two streamlines are to be calculated. Velocity is not required at any remote points.

- Card 6 indicates the coordinates of the point about which moments are calculated (1.2,0.0).

- Card 7 indicates a freestream velocity of 1.0, the coordinates of the airfoil leading edge (0.0,0.0), and the coordinates of the trailing edge (1.2,0.0).

- Card 8 indicates the coordinates of the slat leading edge (-0.06,0.035) and the coordinates of the slat trailing edge (0.02,0.082).

- Card 9 indicates a 5° angle of attack.

- Cards 10 through 16 indicate the coordinates of the airfoil corner points.

- Card 17 indicates the coordinates of the slat vortex control points.

- Cards 18 and 19 indicate the coordinates of the slat source control points.

- Card 20 indicates a slat singularity strength of -1.0 at control point 25.

- Card 21 indicates the input information for calculating streamlines.

- Card 22 indicates the coordinates of the starting points for the two streamlines.

- Cards 23 through 28 indicate the coordinates of the initial slat control points.

III-3.2 Output

The sample problem output listing is shown on the remaining pages of this appendix (75-124).

TWO-DIMENSIONAL SLAT DESIGN BY THE METHOD OF DISTRIBUTED SINGULARITIES

SIMPLE SYMMETRIC AIRFOIL AT 5 DEG. WITH SIMPLE SLAT   TRAILING-EDGE LOCATION - BASELINE CASE

```
-- PRESSURE SPECIFIED AT  1 POINTS, BEGINNING WITH POINT  25 ENDING WITH POINT  25
-- PRESSURES CALCULATED...
-- FORCES CALCULATED...
--  2 STREAMLINES CALCULATED...
-- ANGLE OF ATTACK OF SYSTEM IS   5.0000 DEGREES
-- OODENTS TAKEN ABOUT THE POINT (  1.2000,  0.0000)
-- SLAT VORTICITY DISTRIBUTION TO BE DEFINED AT  3POINTS...
-- SLAT SOURCE DISTRIBUTION TO BE DEFINED AT  4POINTS...
-- SLAT LEADING EDGE        X =  -.0600  Y =   .0350
-- SLAT TRAILING EDGE       X =   .0200  Y =   .0820
-- AIRFOIL LEADING EDGE     X =  0.0000  Y =  0.0000
-- AIRFOIL TRAILING EDGE    X =  1.2000  Y =  0.0000
-- REFERENCE CHORD          C =  1.2000

-- STREAMLINE STARTING COORDINATES
    1.   X =   -.0610  Y =    .0350    2.   X =   -.0600  Y =    .0340

-- SLAT VORTEX CONTROL POINTS
   22.   X =    .0198  Y =    .0824   23.   X =   -.0307  Y =    .0547   24.   X =   -.0461  Y =    .0454

-- SLAT SOURCE CONTROL POINTS
   25.   X =    .0198  Y =    .0824   26.   X =   -.0386  Y =    .0499   27.   X =   -.0461  Y =    .0454
   28.   X =   -.0517  Y =    .0421

-- SPECIFIED PRESSURES
   25.   X =    .0198  Y =    .0824  CP =  -1.0000
```

AIRFOIL DEFINING POINTS

INDEX	X	Y	XBP	YBP	SLOPE
1.	1.20000	0.00000	1.10000	.01000	-5.71059
2.	1.00000	.02000	.90000	.03500	-8.53077
3.	.80000	.05000	.70000	.06250	-7.12502
4.	.60000	.07500	.50000	.08250	-4.28915
5.	.40000	.09000	.30000	.08600	2.29061
6.	.20000	.08200	.15000	.07350	9.64805
7.	.10000	.06500	.07500	.05650	18.77803
8.	.05000	.04800	.03500	.03950	29.53878
9.	.02000	.03100	.01375	.02450	46.12330
10.	.00750	.01800	.00375	.00900	67.38014
11.	0.00000	0.00000	.00375	-.00900	-67.38014
12.	.00750	-.01800	.01375	-.02450	-46.12330
13.	.02000	-.03100	.03500	-.03950	-29.53878
14.	.05000	-.04800	.07500	-.05650	-18.77803
15.	.10000	-.06500	.15000	-.07350	-9.64805
16.	.20000	-.08200	.30000	-.08600	-2.29061
17.	.40000	-.09000	.50000	-.08250	4.28915
18.	.60000	-.07500	.70000	-.06250	7.12502
19.	.80000	-.05000	.90000	-.03500	8.53077
20.	1.00000	-.02000	1.10000	-.01000	5.71059
21.	1.20000	0.00000			

1 VELOCITY COMPONENTS (U,V)

1.	-.158365E-01	-.158365E+00	2.	.276908E-01	.261265E+00	3.	.147648E-01	.115073E+00
4.	.829325E-02	.647848E-01	5.	.507528E-02	.456308E-01	6.	.233525E-02	.276304E-01
7.	.725147E-03	.124463E-01	8.	.251001E-03	.664649E-02	9.	.843701E-04	.388637E-02
10.	.175678E-04	.273561E-02	11.	-.257605E-04	.282746E-02	12.	-.676391E-04	.273405E-02
13.	-.156508E-03	.388188E-02	14.	-.377707E-03	.663444E-02	15.	-.975913E-03	.124117E-01
16.	-.296588E-02	.275069E-01	17.	-.636825E-02	.452963E-01	18.	-.109006E-01	.639824E-01
19.	-.234017E-01	.112025E+00	20.	-.250759E+00	.194679E+00	21.	-.210288E+00	-.134311E+00
22.	.240361E-03	.416454E-02	23.	.245457E-03	.536431E-02	24.	.423071E-04	.125332E-02
25.	.482173E-02	-.270838E-03	26.	.535224E-02	-.236364E-03	27.	.105817E-02	-.330711E-04
28.	.449477E-03	-.129788E-04						

1 INFLUENCE COEFFICIENTS

1.	.15915498E+00	2.	-.26272342E+00	3.	-.11597154E+00	4.	-.65288485E-01	5.	-.45909332E-01
6.	-.27725681E-01	7.	-.12456727E-01	8.	-.66384798E-02	9.	-.38754767E-02	10.	-.27237844E-02
11.	-.28108603E-02	12.	-.27137511E-02	13.	-.38470454E-02	14.	-.65639277E-02	15.	-.12253008E-01
16.	-.27075232E-01	17.	-.44437803E-01	18.	-.62580203E-01	19.	-.10914084E+00	20.	-.16876136E+00
21.	.15456916E+00	22.	-.41677849E-02	23.	-.53621136E-02	24.	-.12513130E-02	25.	-.21028521E-03
26.	-.29737618E-03	27.	-.72384803E-04	28.	-.31810210E-04	29.	.18584828E+00		

2 VELOCITY COMPONENTS (U,V)

1.	-.879969E-02	-.709942E-01	2.	-.373247E-01	-.259172E+00	3.	.381117E-01	.259249E+00
4.	.151282E-01	.114503E+00	5.	.692583E-02	.648541E-01	6.	.263053E-02	.361169E-01
7.	.647558E-03	.156574E-01	8.	.141113E-03	.823198E-02	9.	-.897915E-05	.477716E-02
10.	-.674862E-04	.334787E-02	11.	-.134621E-03	.345249E-02	12.	-.194698E-03	.333786E-02
13.	-.371021E-03	.474813E-02	14.	-.816383E-03	.815241E-02	15.	-.201857E-02	.154186E-01
16.	-.631448E-02	.351605E-01	17.	-.156555E-01	.615289E-01	18.	-.403223E-01	.990890E-01
19.	-.199420E+00	.115966E+00	20.	-.185702E+00	-.156705E+00	21.	-.142898E-01	-.686504E-01
22.	.218057E-03	.510090E-02	23.	.186783E-03	.654031E-02	24.	.218515E-04	.152114E-02
25.	.590264E-02	-.241504E-03	26.	.652209E-02	-.173940E-03	27.	.128310E-02	-.146578E-04
28.	.544533E-03	-.470226E-05						

2 INFLUENCE COEFFICIENTS

1.	.71514114E-01	2.	.26184118E+00	3.	-.26203445E+00	4.	-.11548028E+00	5.	-.65163943E-01
6.	-.36107577E-01	7.	-.15580215E-01	8.	-.81618384E-02	9.	-.47229735E-02	10.	-.33008179E-02
11.	-.33943277E-02	12.	-.32720451E-02	13.	-.46405590E-02	14.	-.79411093E-02	15.	-.14948558E-01
16.	-.33834819E-01	17.	-.58525835E-01	18.	-.92011259E-01	19.	-.85101235E-01	20.	.18251869E+00
21.	.70010654E-01	22.	-.50768097E-02	23.	-.64956556E-02	24.	-.15075486E-02	25.	-.63676828E-03
26.	-.79547515E-03	27.	-.17583974E-03	28.	-.76126062E-04	29.	.23396745E+00		

3 VELOCITY COMPONENTS (U,V)

1.	-.476660E-02	-.367715E-01	2.	-.157053E-01	-.114696E+00	3.	-.333287E-01	-.260395E+00
4.	.309553E-01	.260096E+00	5.	.101683E-01	.115133E+00	6.	.239465E-02	.522744E-01
7.	.181691E-03	.210827E-01	8.	-.217444E-03	.107905E-01	9.	-.266681E-03	.618172E-02
10.	-.282476E-03	.429888E-02	11.	-.395401E-03	.441361E-02	12.	-.489851E-03	.426105E-02
13.	-.862656E-03	.607035E-02	14.	-.182453E-02	.104758E-01	15.	-.449125E-02	.200710E-01
16.	-.152612E-01	.473230E-01	17.	-.485274E-01	.861132E-01	18.	-.156887E+00	.755780E-01
19.	-.156998E+00	-.973419E-01	20.	-.353803E-01	-.101885E+00	21.	-.604466E-02	-.363461E-01
22.	.102741E-03	.657544E-02	23.	-.138740E-04	.836708E-02	24.	-.364381E-04	.193144E-02
25.	.760210E-02	-.101793E-03	26.	.833639E-02	.329748E-04	27.	.162674E-02	.364366E-04
28.	.689370E-03	.177413E-04						

3 INFLUENCE COEFFICIENTS

1.	.37078776E-01	2.	.11575876E+00	3.	.26251837E+00	4.	-.26192729E+00	5.	-.11550537E+00
6.	-.52167746E-01	7.	-.20942475E-01	8.	-.10680196E-01	9.	-.61009086E-02	10.	-.42306449E-02
11.	-.43304797E-02	12.	-.41673823E-02	13.	-.59164795E-02	14.	-.10168609E-01	15.	-.19358932E-01
16.	-.45064660E-01	17.	-.79429109E-01	18.	-.55534974E-01	19.	.11606342E+00	20.	.10548626E+00
21.	.36815168E-01	22.	-.65374186E-02	23.	-.83007445E-02	24.	-.19120076E-02	25.	-.84191736E-03
26.	-.10667216E-02	27.	-.23792790E-03	28.	-.10311007E-03	29.	.21004546E+00		

4 VELOCITY COMPONENTS (U,V)

1.	-.300445E-02	-.250424E-01	2.	-.788881E-02	-.650790E-01	3.	-.122833E-01	-.115563E+00
4.	-.212012E-01	-.261785E+00	5.	.160192E-01	.261705E+00	6.	.668692E-03	.960789E-01
7.	-.124066E-02	.321492E-01	8.	-.115453E-02	.155841E-01	9.	-.899919E-03	.870988E-02
10.	-.793833E-03	.596746E-02	11.	-.100620E-02	.607018E-02	12.	-.118076E-02	.583366E-02
13.	-.202859E-02	.830591E-02	14.	-.429003E-02	.143822E-01	15.	-.109900E-01	.277836E-01
16.	-.432374E-01	.640599E-01	17.	-.135489E+00	.599277E-01	18.	-.137522E+00	-.685373E-01
19.	-.459341E-01	-.892194E-01	20.	-.141832E-01	-.625728E-01	21.	-.357233E-02	-.248806E-01
22.	-.167618E-03	.922249E-02	23.	-.470341E-03	.115749E-01	24.	-.165547E-03	.263610E-02
25.	.106451E-01	.224359E-03	26.	.115142E-01	.502044E-03	27.	.221432E-02	.148961E-03
28.	.935969E-03	.668937E-04						

4 INFLUENCE COEFFICIENTS

1.	.25197015E-01	2.	.65486713E-01	3.	.11615826E+00	4.	.26263701E+00	5.	-.26217010E+00
6.	-.95859777E-01	7.	-.31966362E-01	8.	-.15454084E-01	9.	-.86181833E-02	10.	-.58913796E-02
11.	-.59779283E-02	12.	-.57290174E-02	13.	-.81309307E-02	14.	-.14021073E-01	15.	-.26883822E-01
16.	-.60646742E-01	17.	-.49626612E-01	18.	.78630609E-01	19.	.92404974E-01	20.	.63458336E-01
21.	.25078074E-01	22.	-.91841197E-02	23.	-.11507342E-01	24.	-.26163365E-02	25.	-.10198779E-02
26.	-.13617804E-02	27.	-.31415247E-03	28.	-.13670749E-03	29.	.16141700E+00		

5 VELOCITY COMPONENTS (U,V)

1.	-.181626E-02	-.190849E-01	2.	-.419006E-02	-.461175E-01	3.	-.460212E-02	-.657540E-01
4.	-.366635E-02	-.116210E+00	5.	.687698E-02	-.262392E+00	6.	-.105010E-01	.221097E+00
7.	-.666638E-02	.683123E-01	8.	-.415496E-02	.279943E-01	9.	-.280227E-02	.146824E-01
10.	-.228737E-02	.969536E-02	11.	-.277901E-02	.962391E-02	12.	-.321277E-02	.909720E-02
13.	-.556309E-02	.127828E-01	14.	-.121554E-01	.217191E-01	15.	-.333461E-01	.387100E-01
16.	-.114077E+00	.395207E-01	17.	-.132718E+00	-.590897E-01	18.	-.497641E-01	-.819052E-01
19.	-.176490E-01	-.606101E-01	20.	-.731259E-02	-.452614E-01	21.	-.214089E-02	-.190136E-01
22.	-.641461E-03	.154226E-01	23.	-.143324E-02	.187699E-01	24.	-.455819E-03	.414900E-02
25.	.177380E-01	.816564E-03	26.	.186067E-01	.150386E-02	27.	.346530E-02	.403116E-03
28.	.145679E-02	.178209E-03						

5 INFLUENCE COEFFICIENTS

1.	.18997060E-01	2.	.45913145E-01	3.	.65517566E-01	4.	.11597069E+00	5.	.26245766E+00
6.	-.22134050E+00	7.	-.68524190E-01	8.	-.28138036E-01	9.	-.14782678E-01	10.	-.97790331E-02
11.	-.97272925E-02	12.	-.92183418E-02	13.	-.12994938E-01	14.	-.22187530E-01	15.	-.40011815E-01
16.	-.44048572E-01	17.	.53738012E-01	18.	.79850760E-01	19.	.59856250E-01	20.	.44932917E-01
21.	.18912869E-01	22.	-.15435937E-01	23.	-.18812175E-01	24.	-.41638997E-02	25.	-.10695693E-03
26.	-.75898639E-03	27.	-.26429292E-03	28.	-.11984135E-03	29.	.47270153E-01		

6 VELOCITY COMPONENTS (U,V)

1.	-.110057E-02	-.162299E-01	2.	-.226694E-02	-.379794E-01	3.	-.171248E-02	-.501704E-01
4.	.497622E-03	-.735987E-01	5.	.100374E-01	-.145873E+00	6.	.451993E-01	-.315781E+00
7.	-.394303E-01	.219200E+00	8.	-.179095E-01	.721216E-01	9.	-.972346E-02	.305348E-01
10.	-.721068E-02	.182067E-01	11.	-.836329E-02	.168493E-01	12.	-.953230E-02	.149410E-01
13.	-.165112E-01	.192301E-01	14.	-.354680E-01	.267783E-01	15.	-.789696E-01	.190351E-01
16.	-.119009E+00	-.494603E-01	17.	-.660948E-01	-.877628E-01	18.	-.229422E-01	-.650346E-01
19.	-.959017E-02	-.482390E-01	20.	-.437695E-02	-.375848E-01	21.	-.133413E-02	-.161933E-01
22.	.143206E-03	.313146E-01	23.	-.236754E-02	.356082E-01	24.	-.962524E-03	.737318E-02
25.	.357444E-01	.147278E-03	26.	.350425E-01	.261845E-02	27.	.608616E-02	.861657E-03
28.	.253037E-02	.383568E-03						

6 INFLUENCE COEFFICIENTS

1.	.15815912E-01	2.	.37062333E-01	3.	.49173792E-01	4.	.72641082E-01	5.	.14549180E+00
6.	.31888933E+00	7.	-.22270760E+00	8.	-.74103089E-01	9.	-.31732480E-01	10.	-.19157621E-01
11.	-.18012629E-01	12.	-.16327243E-01	13.	-.21725284E-01	14.	-.32343811E-01	15.	-.32000819E-01
16.	.28815461E-01	17.	.75444250E-01	18.	.60269731E-01	19.	.45949467E-01	20.	.36319638E-01
21.	.15740689E-01	22.	-.30847669E-01	23.	-.35501300E-01	24.	-.74302032E-02	25.	.58454029E-02
26.	.32915437E-02	27.	.17054356E-03	28.	.45936634E-04	29.	-.81034745E-01		

7 VELOCITY COMPONENTS (U,V)

1.	-.708193E-03	-.151109E-01	2.	-.126412E-02	-.349252E-01	3.	-.304017E-03	-.448821E-01
4.	.242136E-02	-.623745E-01	5.	.111339E-01	-.104189E+00	6.	.345115E-01	-.177494E+00
7.	.922524E-01	-.302759E+00	8.	-.788738E-01	.221289E+00	9.	-.325737E-01	.722509E-01
10.	-.194616E-01	.334564E-01	11.	-.205617E-01	.267242E-01	12.	-.220874E-01	.202008E-01
13.	-.351602E-01	.196353E-01	14.	-.620309E-01	.118125E-01	15.	-.883678E-01	-.247902E-01
16.	-.837525E-01	-.806115E-01	17.	-.412031E-01	-.833351E-01	18.	-.153589E-01	-.582445E-01
19.	-.669885E-02	-.438209E-01	20.	-.304879E-02	-.346922E-01	21.	-.910447E-03	-.150884E-01
22.	.159181E-01	.601777E-01	23.	.500076E-02	.643967E-01	24.	-.834584E-03	.122971E-01
25.	.683060E-01	-.165880E-01	26.	.628742E-01	-.379188E-02	27.	.999751E-02	.883444E-03
28.	.409750E-02	.436087E-03						

7 INFLUENCE COEFFICIENTS

1.	.14078600E-01	2.	.32659296E-01	3.	.42395288E-01	4.	.59833895E-01	5.	.10222703E+00
6.	.17915591E+00	7.	.31633999E+00	8.	-.23489983E+00	9.	-.78890770E-01	10.	-.37940336E-01
11.	-.31920638E-01	12.	-.26235514E-01	13.	-.29908285E-01	14.	-.31151663E-01	15.	-.49751356E-02
16.	.49360592E-01	17.	.65636047E-01	18.	.50200257E-01	19.	.39332045E-01	20.	.31864211E-01
21.	.13992200E-01	22.	-.51850574E-01	23.	-.59359256E-01	24.	-.11911248E-01	25.	.37692949E-01
26.	.23829429E-01	27.	.23818068E-02	28.	.90612106E-03	29.	-.23816112E+00		

8 VELOCITY COMPONENTS (U,V)

1.	-.431838E-03	-.145775E-01	2.	-.564471E-03	-.334889E-01	3.	.685783E-03	-.424667E-01
4.	.384867E-02	-.575735E-01	5.	.130040E-01	-.903578E-01	6.	.311369E-01	-.127259E+00
7.	.597408E-01	-.157731E+00	8.	.137667E+00	-.262219E+00	9.	-.113394E+00	.192175E+00
10.	-.553802E-01	.696581E-01	11.	-.462912E-01	.397158E-01	12.	-.426205E-01	.209227E-01
13.	-.555694E-01	.850380E-02	14.	-.737648E-01	-.148323E-01	15.	-.763879E-01	-.519280E-01
16.	-.617366E-01	-.879727E-01	17.	-.304707E-01	-.799898E-01	18.	-.117224E-01	-.552581E-01
19.	-.509041E-02	-.418368E-01	20.	-.220717E-02	-.333461E-01	21.	-.620056E-03	-.145635E-01
22.	.698033E-01	.610758E-01	23.	.434762E-01	.901898E-01	24.	.226048E-02	.185907E-01
25.	.724838E-01	-.756725E-01	26.	.891098E-01	-.389995E-01	27.	.149930E-01	-.119381E-02
28.	.608644E-02	-.267294E-03						

8 INFLUENCE COEFFICIENTS

1.	.12469873E-01	2.	.28857834E-01	3.	.37285082E-01	4.	.51987640E-01	5.	.85024446E-01
6.	.12606915E+00	7.	.16668264E+00	8.	.29600820E+00	9.	-.22310180E+00	10.	-.87907184E-01
11.	-.57375734E-01	12.	-.39215665E-01	13.	-.34794900E-01	14.	-.23462553E-01	15.	.75183171E-02
16.	.46101275E-01	17.	.54570422E-01	18.	.42296456E-01	19.	.33889307E-01	20.	.27923641E-01
21.	.12364852E-01	22.	-.18723350E-01	23.	-.57032838E-01	24.	-.15059855E-01	25.	.10157216E+00
26.	.77862689E-01	27.	.84303613E-02	28.	.32332420E-02	29.	-.41530908E+00		

9 VELOCITY COMPONENTS (U,V)

1.	-.223464E-03	-.143088E-01	2.	-.360241E-04	-.327655E-01	3.	.144763E-02	-.412507E-01
4.	.500230E-02	-.551849E-01	5.	.148423E-01	-.839242E-01	6.	.321520E-01	-.109505E+00
7.	.494529E-01	-.109705E+00	8.	.902670E-01	-.139830E+00	9.	.214957E+00	-.232152E+00
10.	-.202366E+00	.176739E+00	11.	-.108290E+00	.578996E-01	12.	-.707679E-01	.119014E-01
13.	-.701011E-01	-.114156E-01	14.	-.732598E-01	-.380094E-01	15.	-.642732E-01	-.655866E-01
16.	-.494716E-01	-.906231E-01	17.	-.246973E-01	-.784865E-01	18.	-.955529E-02	-.538995E-01
19.	-.403619E-02	-.408912E-01	20.	-.161027E-02	-.326826E-01	21.	-.404830E-03	-.143006E-01
22.	.837133E-01	.208309E-01	23.	.850824E-01	.726404E-01	24.	.971029E-02	.218546E-01
25.	.295030E-01	-.949569E-01	26.	.757501E-01	-.798455E-01	27.	.181194E-01	-.656501E-02
28.	.748592E-02	-.220600E-02						

9 INFLUENCE COEFFICIENTS

1.	.97564846E-02	2.	.22684077E-01	3.	.29634698E-01	4.	.41854942E-01	5.	.68867379E-01
6.	.99075088E-01	7.	.11168456E+00	8.	.16198472E+00	9.	.31585490E+00	10.	-.26837174E+00
11.	-.11818960E+00	12.	-.59260857E-01	13.	-.42618947E-01	14.	-.26463415E-01	15.	-.87155319E-03
16.	.27150844E-01	17.	.36597090E-01	18.	.30470479E-01	19.	.25432625E-01	20.	.21491872E-01
21.	.96200483E-02	22.	.45905186E-01	23.	.10982571E-01	24.	-.81480833E-02	25.	.87082178E-01
26.	.10994479E+00	27.	.17611341E-01	28.	.69250993E-02	29.	-.65768167E+00		

10 VELOCITY COMPONENTS (U,V)

1.	-.242563E-04	-.141850E-01	2.	.475492E-03	-.324242E-01	3.	.221380E-02	-.406481E-01
4.	.625011E-02	-.539336E-01	5.	.172110E-01	-.804028E-01	6.	.354708E-01	-.100070E+00
7.	.500517E-01	-.897327E-01	8.	.721959E-01	-.864678E-01	9.	.113858E+00	-.885222E-01
10.	.233318E+00	-.108781E+00	11.	-.253765E+00	.779629E-01	12.	-.120381E+00	-.117705E-01
13.	-.827008E-01	-.372828E-01	14.	-.707033E-01	-.583513E-01	15.	-.557188E-01	-.759446E-01
16.	-.415420E-01	-.938752E-01	17.	-.206774E-01	-.785503E-01	18.	-.786984E-02	-.534847E-01
19.	-.314141E-02	-.405210E-01	20.	-.106828E-02	-.323946E-01	21.	-.202551E-03	-.141821E-01
22.	.692115E-01	.310320E-02	23.	.897868E-01	.414711E-01	24.	.158660E-01	.178633E-01
25.	.766975E-02	-.806595E-01	26.	.459568E-01	-.874927E-01	27.	.157643E-01	-.118300E-01
28.	.685461E-02	-.437952E-02						

10 INFLUENCE COEFFICIENTS

1.	.54333915E-02	2.	.12909744E-01	3.	.17677377E-01	4.	.26513009E-01	5.	.46811191E-01
6.	.71230710E-01	7.	.80714152E-01	8.	.99899247E-01	9.	.13914643E+00	10.	.25720888E+00
11.	-.26423028E+00	12.	-.10659344E+00	13.	-.61999639E-01	14.	-.42821722E-01	15.	-.22223251E-01
16.	-.22406477E-02	17.	.11124779E-01	18.	.13306563E-01	19.	.12685223E-01	20.	.11473370E-01
21.	.52676804E-02	22.	.62694009E-01	23.	.66929702E-01	24.	.77750097E-02	25.	.38102638E-01
26.	.76072732E-01	27.	.19101640E-01	28.	.80117625E-02	29.	-.88604290E+00		

11 VELOCITY COMPONENTS (U,V)

1.	.202551E-03	-.141821E-01	2.	.106828E-02	-.323946E-01	3.	.314141E-02	-.405210E-01
4.	.786984E-02	-.534847E-01	5.	.206774E-01	-.785503E-01	6.	.415420E-01	-.938752E-01
7.	.557188E-01	-.759446E-01	8.	.707033E-01	-.583513E-01	9.	.827008E-01	-.372828E-01
10.	.120381E+00	-.117705E-01	11.	.253765E+00	.779629E-01	12.	-.233318E+00	-.108781E+00
13.	-.113858E+00	-.885222E-01	14.	-.721959E-01	-.864678E-01	15.	-.500517E-01	-.897327E-01
16.	-.354708E-01	-.100070E+00	17.	-.172110E-01	-.804028E-01	18.	-.625011E-02	-.539336E-01
19.	-.221380E-02	-.406481E-01	20.	-.475492E-03	-.324242E-01	21.	.242563E-04	-.141850E-01
22.	.547676E-01	.167328E-02	23.	.741547E-01	.257505E-01	24.	.154715E-01	.120151E-01
25.	.445000E-02	-.640500E-01	26.	.288823E-01	-.733983E-01	27.	.109817E-01	-.122433E-01
28.	.495003E-02	-.482425E-02						

11 INFLUENCE COEFFICIENTS

1.	-.52676804E-02	2.	-.11473370E-01	3.	-.12685223E-01	4.	-.13306563E-01	5.	-.11124779E-01
6.	.22406477E-02	7.	.22223251E-01	8.	.42821722E-01	9.	.61999639E-01	10.	.10659344E+00
11.	.26423028E+00	12.	-.25720888E+00	13.	-.13914643E+00	14.	-.99899247E-01	15.	-.80714152E-01
16.	-.71230710E-01	17.	-.46811191E-01	18.	-.26513009E-01	19.	-.17677377E-01	20.	-.12909744E-01
21.	-.54333915E-02	22.	.51198284E-01	23.	.78354563E-01	24.	.18902553E-01	25.	-.20526922E-01
26.	-.15695095E-02	27.	.54279670E-02	28.	.27137746E-02	29.	-.95308578E+00		

12 VELOCITY COMPONENTS (U,V)

1.	.404830E-03	-.143006E-01	2.	.161027E-02	-.326826E-01	3.	.403619E-02	-.408912E-01
4.	.955529E-02	-.538995E-01	5.	.246973E-01	-.784865E-01	6.	.494716E-01	-.906231E-01
7.	.642732E-01	-.655866E-01	8.	.732598E-01	-.380094E-01	9.	.701011E-01	-.114156E-01
10.	.707679E-01	.119014E-01	11.	.108290E+00	.578996E-01	12.	.202366E+00	.176739E+00
13.	-.214957E+00	-.232152E+00	14.	-.902670E-01	-.139830E+00	15.	-.494529E-01	-.109705E+00
16.	-.321520E-01	-.109505E+00	17.	-.148423E-01	-.839242E-01	18.	-.500230E-02	-.551849E-01
19.	-.144763E-02	-.412507E-01	20.	.360241E-04	-.327655E-01	21.	.223464E-03	-.143088E-01
22.	.456211E-01	.560102E-02	23.	.594116E-01	.234856E-01	24.	.125335E-01	.938715E-02
25.	.824193E-02	-.530425E-01	26.	.255114E-01	-.587963E-01	27.	.849190E-02	-.100884E-01
28.	.381128E-02	-.405897E-02						

STREAMLINE 1

X	Y	X	Y	X	Y	X	Y	X	Y
-.0609	.0351	-.0608	.0351	-.0608	.0352	-.0607	.0353	-.0607	.0354
44.7512	.9993	49.6306	.9994	55.3008	.9995	61.8368	.9995	68.8008	.9995
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0606	.0355	-.0606	.0356	-.0606	.0357	-.0606	.0358	-.0606	.0359
75.5607	.9995	81.5684	.9995	86.5711	.9994	90.5913	.9993	93.7878	.9992
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0606	.0360	-.0606	.0362	-.0607	.0364	-.0607	.0365	-.0607	.0368
96.9314	.9991	99.9260	.9989	102.7581	.9987	104.8624	.9984	107.1554	.9981
	-.0643		-.0643		-.0643		-.0643		.1253
-.0608	.0370	-.0609	.0372	-.0609	.0373	-.0610	.0375	-.0611	.0377
109.1472	.9978	110.9875	.9975	112.7325	.9971	114.9749	.9967	117.0970	.9962
	.1166		.1166		.1166		-.0643		-.0643
-.0612	.0379	-.0613	.0381	-.0614	.0383	-.0616	.0385	-.0617	.0387
119.0626	.9957	120.8392	.9952	122.9256	.9944	124.6640	.9935	126.5062	.9921
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0619	.0389	-.0621	.0392	-.0624	.0395	-.0627	.0399	-.0631	.0404
127.8471	.9905	129.0387	.9881	129.8505	.9844	130.0248	.9784	129.3379	.9685
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0634	.0408	-.0637	.0411	-.0639	.0414	-.0641	.0417	-.0643	.0420
128.4989	.9599	127.5446	.9505	126.5526	.9405	125.5658	.9300	124.6071	.9191
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0646	.0424	-.0648	.0427	-.0650	.0431	-.0652	.0434	-.0654	.0437
123.3667	.9042	122.1890	.8887	121.0794	.8730	120.0388	.8571	119.0656	.8412
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0656	.0441	-.0658	.0444	-.0659	.0447	-.0661	.0451	-.0663	.0455
117.8453	.8201	116.7108	.7989	115.6565	.7779	114.6771	.7571	113.4572	.7299
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0664	.0458	-.0666	.0462	-.0667	.0465	-.0669	.0470	-.0670	.0474
112.3279	.7030	111.2818	.6766	110.3120	.6509	109.1077	.6174	107.9942	.5847
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0671	.0477	-.0673	.0481	-.0674	.0485	-.0675	.0490	-.0676	.0494
106.9638	.5529	106.0092	.5222	104.8252	.4825	103.7309	.4441	102.7183	.4071
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0677	.0497	-.0678	.0502	-.0679	.0507	-.0679	.0511	-.0680	.0515
101.7804	.3715	100.6177	.3259	99.5430	.2821	98.5484	.2401	97.6268	.1999
	-.0643		-.0643		-.0643		-.0643		-.0643

-.0681	.0519	-.0681	.0524	-.0681	.0528	-.0682	.0532	-.0682	.0538
96.4849	.1488	95.4291	.1000	94.4516	.0534	93.5457	.0091	92.4234	-.0471
	-.0643		-.0643		-.0643		-.0642		-.0642
-.0682	.0542	-.0682	.0547	-.0682	.0551	-.0682	.0555	-.0682	.0560
91.3852	-.1004	90.4237	-.1511	89.5325	-.1991	88.7056	-.2445	87.6805	-.3016
	-.0642		-.0642		-.0642		-.0642		-.0642
-.0682	.0564	-.0682	.0569	-.0682	.0573	-.0681	.0578	-.0681	.0583
86.7308	-.3557	85.8504	-.4067	85.0332	-.4548	84.0203	-.5151	83.0817	-.5720
	-.0642		-.0642		-.0642		-.0642		-.0642
-.0680	.0587	-.0680	.0591	-.0679	.0596	-.0678	.0601	-.0677	.0606
82.2112	-.6255	81.4033	-.6758	80.4016	-.7387	79.4733	-.7978	78.6122	-.8532
	-.0642		-.0642		-.0642		-.0642		-.0642
-.0676	.0610	-.0675	.0615	-.0674	.0620	-.0673	.0625	-.0672	.0629
77.8127	-.9052	76.8217	-.9699	75.9030	-1.0306	75.0506	-1.0873	74.2590	-1.1403
	-.0642		-.0642		-.0642		-.0642		-.0642
-.0670	.0634	-.0669	.0639	-.0667	.0644	-.0666	.0648	-.0664	.0654
73.2778	-1.2062	72.3680	-1.2677	71.5237	-1.3251	70.7396	-1.3786	69.7676	-1.4449
	-.0641		-.0641		-.0641		-.0641		-.0641
-.0662	.0659	-.0660	.0663	-.0658	.0668	-.0656	.0673	-.0654	.0678
68.8661	-1.5066	68.0295	-1.5640	67.2523	-1.6173	66.2889	-1.6832	65.3953	-1.7444
	-.0641		-.0641		-.0641		-.0641		-.0641
-.0652	.0683	-.0650	.0687	-.0648	.0691	-.0645	.0696	-.0643	.0701
64.5658	-1.8011	63.7951	-1.8537	63.0787	-1.9024	62.1901	-1.9625	61.3651	-2.0181
	-.0641		-.0641		-.0641		-.0641		-.0641
-.0640	.0705	-.0638	.0709	-.0635	.0715	-.0632	.0719	-.0630	.0724
60.5986	-2.0695	59.8860	-2.1171	59.0022	-2.1755	58.1815	-2.2293	57.4190	-2.2790
	-.0641		-.0640		-.0640		-.0640		-.0640
-.0627	.0728	-.0624	.0733	-.0620	.0737	-.0617	.0742	-.0615	.0746
56.7100	-2.3248	55.8307	-2.3809	55.0142	-2.4324	54.2555	-2.4798	53.5500	-2.5234
	-.0640		-.0640		-.0640		-.0640		-.0640
-.0611	.0751	-.0607	.0755	-.0604	.0760	-.0601	.0764	-.0597	.0768
52.6751	-2.5765	51.8625	-2.6251	51.1075	-2.6697	50.4053	-2.7106	49.5345	-2.7601
	-.0640		-.0640		-.0640		-.0640		-.0640
-.0593	.0773	-.0589	.0777	-.0586	.0781	-.0581	.0786	-.0577	.0790
48.7258	-2.8053	47.9743	-2.8465	47.2753	-2.8842	46.4086	-2.9296	45.6036	-2.9708
	-.0639		-.0639		-.0639		-.0639		-.0639
-.0573	.0795	-.0569	.0798	-.0565	.0802	-.0561	.0806	-.0556	.0810
44.8554	-3.0082	44.1596	-3.0422	43.5121	-3.0731	42.7088	-3.1102	41.9621	-3.1437
	-.0639		-.0639		-.0639		-.0639		-.0639

-.0552	.0814	-.0548	.0818	-.0543	.0822	-.0539	.0826	-.0534	.0829
41.2676	-3.1740	40.6214	-3.2013	39.8196	-3.2339	39.0744	-3.2631	38.3812	-3.2893
	-.0639		-.0639		-.0639		-.0638		-.0638
-.0530	.0833	-.0525	.0837	-.0520	.0841	-.0515	.0844	-.0510	.0847
37.7362	-3.3128	36.9360	-3.3404	36.1921	-3.3650	35.5003	-3.3867	34.8564	-3.4060
	-.0638		-.0638		-.0638		-.0638		-.0638
-.0505	.0851	-.0499	.0855	-.0494	.0858	-.0489	.0861	-.0483	.0865
34.0576	-3.4284	33.3151	-3.4479	32.6244	-3.4649	31.9816	-3.4797	31.1842	-3.4964
	-.0638		-.0638		-.0638		-.0638		-.0638
-.0477	.0869	-.0472	.0872	-.0467	.0875	-.0462	.0877	-.0456	.0881
30.4428	-3.5106	29.7532	-3.5226	29.1114	-3.5327	28.5136	-3.5412	27.7717	-3.5502
	-.0637		-.0637		-.0637		-.0637		-.0637
-.0450	.0884	-.0445	.0886	-.0440	.0889	-.0433	.0892	-.0427	.0895
27.0816	-3.5574	26.4392	-3.5629	25.8409	-3.5670	25.0983	-3.5707	24.4075	-3.5727
	-.0637		-.0637		-.0637		-.0637		-.0637
-.0422	.0898	-.0416	.0900	-.0410	.0903	-.0403	.0905	-.0397	.0908
23.7644	-3.5735	23.1654	-3.5732	22.4220	-3.5712	21.7303	-3.5680	21.0863	-3.5638
	-.0637		-.0637		-.0636		-.0636		-.0636
-.0392	.0910	-.0385	.0913	-.0378	.0915	-.0372	.0917	-.0366	.0919
20.4865	-3.5589	19.7420	-3.5511	19.0492	-3.5425	18.4042	-3.5332	17.8033	-3.5235
	-.0636		-.0636		-.0636		-.0636		-.0636
-.0360	.0921	-.0353	.0923	-.0347	.0925	-.0341	.0927	-.0335	.0928
17.2432	-3.5136	16.5477	-3.4997	15.9000	-3.4856	15.2966	-3.4714	14.7341	-3.4571
	-.0636		-.0636		-.0636		-.0636		-.0636
-.0328	.0930	-.0321	.0932	-.0314	.0933	-.0308	.0935	-.0301	.0936
14.0355	-3.4380	13.3849	-3.4189	12.7786	-3.4000	12.2134	-3.3814	11.5114	-3.3568
	-.0635		-.0635		-.0635		-.0635		-.0635
-.0294	.0938	-.0287	.0939	-.0281	.0940	-.0273	.0942	-.0266	.0943
10.8575	-3.3326	10.2480	-3.3089	9.6797	-3.2859	8.9737	-3.2558	8.3159	-3.2264
	-.0635		-.0635		-.0635		-.0635		-.0635
-.0259	.0944	-.0252	.0945	-.0246	.0945	-.0238	.0946	-.0231	.0947
7.7027	-3.1979	7.1308	-3.1704	6.5972	-3.1438	5.9339	-3.1095	5.3155	-3.0764
	-.0635		-.0635		-.0635		-.0634		-.0634
-.0225	.0948	-.0218	.0948	-.0210	.0949	-.0203	.0949	-.0196	.0950
4.7385	-3.0445	4.2000	-3.0139	3.5306	-2.9745	2.9062	-2.9367	2.3236	-2.9004
	-.0634		-.0634		-.0634		-.0634		-.0634
-.0189	.0950	-.0181	.0950	-.0174	.0950	-.0167	.0950	-.0160	.0950
1.7797	-2.8656	1.1033	-2.8211	.4722	-2.7784	-.1169	-2.7376	-.6671	-2.6987
	-.0634		-.0634		-.0634		-.0634		-.0634

-.0154 -1.1810	.0950 -2.6616 -.0634	-.0146 -1.8206	.0950 -2.6142 -.0633	-.0138 -2.4178	.0950 -2.5691 -.0633	-.0131 -2.9757	.0950 -2.5260 -.0633	-.0125 -3.4972	.0949 -2.4851 -.0633
-.0117 -4.1462	.0949 -2.4329 -.0633	-.0109 -4.7526	.0948 -2.3833 -.0633	-.0102 -5.3193	.0948 -2.3360 -.0633	-.0096 -5.8492	.0947 -2.2912 -.0633	-.0087 -6.5090	.0946 -2.2341 -.0633
-.0079 -7.1257	.0945 -2.1798 -.0633	-.0072 -7.7025	.0944 -2.1283 -.0633	-.0065 -8.2421	.0943 -2.0793 -.0633	-.0059 -8.7470	.0942 -2.0329 -.0633	-.0051 -9.3762	.0941 -1.9740 -.0633
-.0044 -9.9649	.0940 -1.9182 -.0633	-.0037 -10.5160	.0939 -1.8652 -.0632	-.0030 -11.0320	.0938 -1.8149 -.0632	-.0022 -11.6752	.0936 -1.7513 -.0632	-.0014 -12.2776	.0934 -1.6909 -.0632
-.0007 -12.8418	.0933 -1.6337 -.0632	-.0000 -13.3705	.0931 -1.5794 -.0632	.0008 -14.0299	.0929 -1.5107 -.0632	.0016 -14.6480	.0927 -1.4456 -.0632	.0023 -15.2274	.0926 -1.3838 -.0632
.0030 -15.7707	.0924 -1.3252 -.0632	.0036 -16.2803	.0922 -1.2698 -.0632	.0044 -16.9165	.0920 -1.1997 -.0632	.0051 -17.5134	.0917 -1.1332 -.0632	.0058 -18.0736	.0915 -1.0702 -.0632
.0065 -18.5994	.0913 -1.0105 -.0632	.0073 -19.2560	.0910 -.9349 -.0632	.0080 -19.8725	.0908 -.8631 -.0631	.0087 -20.4513	.0905 -.7950 -.0631	.0094 -20.9948	.0903 -.7304 -.0631
.0100 -21.5049	.0900 -.6692 -.0631	.0108 -22.1412	.0897 -.5915 -.0631	.0115 -22.7378	.0894 -.5177 -.0631	.0121 -23.2965	.0892 -.4474 -.0631	.0128 -23.8188	.0889 -.3806 -.0631
.0135 -24.4650	.0886 -.2956 -.0631	.0143 -25.0629	.0882 -.2143 -.0631	.0149 -25.6110	.0879 -.1365 -.0631	.0156 -26.1066	.0876 -.0620 -.0631	.0163 -26.6789	.0872 .0336 -.0631
.0171 -27.1359	.0869 .1260 -.0631	.0180 -27.4843	.0864 .2456 -.0631	.0191 -27.1008	.0858 .4006 -.0631	.0196 -26.4056	.0856 .4617 -.0631	.0200 -25.2584	.0854 .5168 -.0631
.0203 -24.0627	.0852 .5527 .1265	.0206 -22.6000	.0851 .5832 .1265	.0208 -20.9120	.0850 .6079 .1265	.0211 -19.0665	.0849 .6272 .1178	.0213 -17.1448	.0848 .6412 .1178
.0216 -15.2264	.0847 .6508 .1178	.0218 -13.3760	.0847 .6567 .1178	.0220 -11.6383	.0846 .6597 .1178	.0223 -9.5149	.0846 .6603 .1178	.0225 -7.6151	.0845 .6580 .1178
.0227 -5.9406	.0845 .6538 .1178	.0230 -4.0093	.0845 .6463 .1178	.0233 -2.3704	.0845 .6374 .1178	.0236 -.9815	.0844 .6278 -.0631	.0239 .5709	.0844 .6144 -.0631

.0242	.0844	.0245	.0844	.0249	.0845	.0253	.0845	.0257	.0845
1.8546	.6009	3.2590	.5831	4.3975	.5658	5.6222	.5437	6.6001	.5226
	-.0631		-.0631		-.0631		-.0631		-.0631
.0263	.0846	.0267	.0847	.0273	.0848	.0279	.0848	.0286	.0850
7.6386	.4964	8.4581	.4719	9.3194	.4417	9.9923	.4139	10.6928	.3800
	-.0631		-.0632		-.0632		-.0632		-.0632
.0292	.0851	.0300	.0853	.0311	.0855	.0320	.0857	.0332	.0859
11.2346	.3492	11.7927	.3119	12.3467	.2674	12.7541	.2277	13.1515	.1807
	-.0632		-.0632		-.0632		-.0632		-.0632
.0347	.0863	.0360	.0866	.0378	.0870	.0399	.0876	.0426	.0883
13.5190	.1258	13.7669	.0779	13.9839	.0225	14.1516	-.0405	14.2523	-.1106
	-.0632		-.0632		-.0632		-.0632		-.0632

STREAMLINE 2

X	Y	X	Y	X	Y	X	Y	X	Y
-.0599	.0339	-.0599	.0338	-.0598	.0336	-.0597	.0334	-.0595	.0332
-55.7518	.9987	-59.0446	.9983	-61.7991	.9978	-63.8879	.9970	-65.2619	.9956
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0594	.0329	-.0592	.0325	-.0590	.0320	-.0588	.0316	-.0586	.0311
-65.9255	.9935	-65.9042	.9901	-65.2196	.9847	-64.2640	.9781	-63.2272	.9707
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0584	.0307	-.0582	.0304	-.0580	.0300	-.0578	.0297	-.0576	.0293
-62.1838	.9626	-61.1666	.9540	-60.1911	.9449	-59.2640	.9355	-58.0813	.9227
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0574	.0290	-.0571	.0286	-.0569	.0283	-.0567	.0280	-.0565	.0277
-56.9695	.9095	-55.9271	.8961	-54.9514	.8827	-54.0390	.8693	-52.8935	.8516
	-.0643		-.0643		-.0643		-.0643		-.0643
-.0562	.0273	-.0560	.0270	-.0558	.0267	-.0555	.0264	-.0552	.0261
-51.8263	.8340	-50.8319	.8166	-49.9052	.7995	-48.7469	.7772	-47.6702	.7553
	-.0643		-.0644		-.0644		-.0644		-.0644
-.0549	.0258	-.0547	.0255	-.0543	.0252	-.0540	.0249	-.0537	.0246
-46.6686	.7340	-45.7364	.7133	-44.5738	.6865	-43.4938	.6606	-42.4897	.6355
	-.0644		-.0644		-.0644		-.0644		-.0644
-.0534	.0243	-.0531	.0240	-.0527	.0237	-.0524	.0235	-.0521	.0232
-41.5556	.6114	-40.3919	.5804	-39.3110	.5507	-38.3061	.5221	-37.3712	.4948
	-.0644		-.0644		-.0644		-.0644		-.0644
-.0518	.0230	-.0514	.0227	-.0511	.0225	-.0507	.0222	-.0504	.0220
-36.5008	.4688	-35.4168	.4357	-34.4088	.4042	-33.4708	.3742	-32.5975	.3457
	-.0644		-.0644		-.0644		-.0644		-.0644
-.0500	.0218	-.0496	.0215	-.0493	.0213	-.0489	.0211	-.0485	.0209
-31.5098	.3097	-30.4982	.2754	-29.5566	.2430	-28.6795	.2123	-27.5873	.1736
	-.0644		-.0644		-.0644		-.0644		-.0644
-.0481	.0207	-.0477	.0205	-.0473	.0203	-.0469	.0201	-.0464	.0199
-26.5710	.1369	-25.6246	.1023	-24.7428	.0696	-23.6447	.0286	-22.6224	-.0102
	-.0644		-.0644		-.0644		-.0644		-.0644
-.0460	.0197	-.0456	.0196	-.0453	.0194	-.0448	.0193	-.0444	.0191
-21.6700	-.0467	-20.7824	-.0812	-19.9545	-.1136	-18.9231	-.1543	-17.9619	-.1926
	-.0644		-.0644		-.0644		-.0644		-.0644
-.0440	.0190	-.0436	.0189	-.0431	.0187	-.0426	.0186	-.0422	.0185
-17.0657	-.2286	-16.2297	-.2625	-15.1881	-.3050	-14.2170	-.3449	-13.3114	-.3825
	-.0645		-.0645		-.0645		-.0645		-.0645

-.0418 -12.4664	.0184 -.4178 -.0645	-.0413 -11.4134	.0183 -.4621 -.0645	-.0408 -10.4316	.0182 -.5037 -.0645	-.0404 -9.5158	.0181 -.5428 -.0645	-.0399 -8.6612	.0180 -.5797 -.0645
-.0394 -7.5962	.0179 -.6258 -.0645	-.0389 -6.6031	.0179 -.6692 -.0645	-.0385 -5.6767	.0178 -.7100 -.0645	-.0380 -4.8122	.0178 -.7485 -.0645	-.0376 -4.0052	.0177 -.7848 -.0645
-.0371 -2.9994	.0177 -.8302 -.0645	-.0366 -2.0613	.0177 -.8730 -.0645	-.0362 -1.1861	.0177 -.9134 -.0645	-.0357 -.3691	.0177 -.9515 -.0645	-.0352 .6486	.0177 -.9993 -.0645
-.0347 1.5975	.0177 -1.0444 -.0645	-.0342 2.4826	.0177 -1.0870 -.0646	-.0338 3.3083	.0177 -1.1273 -.0646	-.0332 4.3364	.0177 -1.1779 -.0646	-.0327 5.2944	.0178 -1.2259 -.0646
-.0322 6.1874	.0178 -1.2713 -.0646	-.0318 7.0198	.0179 -1.3143 -.0646	-.0314 7.7961	.0179 -1.3550 -.0646	-.0309 8.7619	.0180 -1.4062 -.0646	-.0304 9.6613	.0181 -1.4548 -.0646
-.0299 10.4989	.0181 -1.5009 -.0646	-.0295 11.2792	.0182 -1.5446 -.0646	-.0289 12.2490	.0183 -1.5998 -.0646	-.0284 13.1509	.0184 -1.6522 -.0646	-.0280 13.9897	.0185 -1.7021 -.0646
-.0275 14.7702	.0186 -1.7494 -.0646	-.0270 15.7387	.0188 -1.8091 -.0646	-.0265 16.6378	.0189 -1.8661 -.0647	-.0260 17.4728	.0191 -1.9202 -.0647	-.0256 18.2483	.0192 -1.9716 -.0647
-.0250 19.2088	.0194 -2.0366 -.0647	-.0245 20.0987	.0196 -2.0986 -.0647	-.0240 20.9234	.0197 -2.1575 -.0647	-.0236 21.6878	.0199 -2.2135 -.0647	-.0230 22.6325	.0201 -2.2843 -.0647
-.0225 23.5055	.0204 -2.3516 -.0647	-.0220 24.3125	.0206 -2.4156 -.0647	-.0216 25.0588	.0208 -2.4763 -.0647	-.0210 25.9788	.0210 -2.5529 -.0647	-.0205 26.8266	.0213 -2.6256 -.0647
-.0200 27.6082	.0215 -2.6945 -.0647	-.0196 28.3292	.0218 -2.7597 -.0647	-.0190 29.2154	.0221 -2.8417 -.0648	-.0185 30.0295	.0223 -2.9191 -.0648	-.0180 30.7779	.0226 -2.9922 -.0648
-.0176 31.4665	.0229 -3.0610 -.0648	-.0170 32.3103	.0232 -3.1470 -.0648	-.0165 33.0831	.0236 -3.2277 -.0648	-.0160 33.7916	.0239 -3.3033 -.0648	-.0156 34.4415	.0242 -3.3740 -.0648
-.0150 35.2358	.0245 -3.4616 -.0648	-.0145 35.9610	.0249 -3.5430 -.0648	-.0141 36.6240	.0253 -3.6185 -.0648	-.0136 37.2308	.0256 -3.6884 -.0648	-.0131 37.9704	.0260 -3.7740 -.0649
-.0125 38.6440	.0264 -3.8524 -.0649	-.0121 39.2584	.0268 -3.9240 -.0649	-.0115 40.0044	.0273 -4.0104 -.0649	-.0109 40.6814	.0277 -4.0881 -.0649	-.0104 41.2968	.0281 -4.1577 -.0649

-.0100 41.8575	.0286 -4.2197 -.0649	-.0094 42.5375	.0291 -4.2924 -.0649	-.0088 43.1542	.0296 -4.3554 -.0649	-.0083 43.7149	.0300 -4.4097 -.0649	-.0079 44.2259	.0305 -4.4561 -.0649
-.0073 44.8462	.0311 -4.5075 -.0650	-.0068 45.4098	.0316 -4.5490 -.0650	-.0063 45.9234	.0321 -4.5816 -.0650	-.0057 46.5469	.0327 -4.6137 -.0650	-.0051 47.1140	.0333 -4.6348 -.0650
-.0046 47.6317	.0339 -4.6467 -.0650	-.0041 48.1057	.0344 -4.6509 -.0650	-.0035 48.6841	.0351 -4.6464 -.0650	-.0029 49.2136	.0358 -4.6327 -.0650	-.0024 49.6998	.0364 -4.6116 -.0650
-.0019 50.1476	.0369 -4.5847 -.0650	-.0013 50.6974	.0377 -4.5416 -.0651	-.0007 51.2040	.0383 -4.4920 -.0651	-.0002 51.6721	.0390 -4.4378 -.0651	.0003 52.1057	.0396 -4.3806 -.0651
.0009 52.6411	.0403 -4.3006 -.0651	.0014 53.1375	.0411 -4.2177 -.0651	.0019 53.5990	.0417 -4.1336 -.0651	.0024 54.0287	.0424 -4.0494 -.0651	.0030 54.5620	.0432 -3.9374 -.0651
.0035 55.0595	.0439 -3.8265 -.0651	.0040 55.5244	.0447 -3.7177 -.0651	.0045 55.9593	.0453 -3.6121 -.0651	.0051 56.5013	.0462 -3.4755 -.0652	.0056 57.0891	.0470 -3.3437 -.0652
.0061 57.4852	.0477 -3.2174 -.0652	.0065 57.9319	.0485 -3.0970 -.0652	.0070 58.3511	.0491 -2.9828 -.0652	.0075 58.8742	.0500 -2.8385 -.0652	.0080 59.3650	.0508 -2.7023 -.0652
.0084 59.8252	.0515 -2.5742 -.0652	.0088 60.2566	.0522 -2.4540 -.0652	.0093 60.7938	.0531 -2.3039 -.0652	.0098 61.2962	.0539 -2.1637 -.0652	.0102 61.7656	.0547 -2.0331 -.0652
.0106 62.2036	.0555 -1.9116 -.0652	.0111 62.7457	.0564 -1.7613 -.0652	.0115 63.2489	.0573 -1.6221 -.0653	.0120 63.7151	.0581 -1.4933 -.0653	.0123 64.1462	.0589 -1.3744 -.0653
.0128 64.6728	.0598 -1.2281 -.0653	.0132 65.1541	.0608 -1.0937 -.0653	.0136 65.5918	.0616 -.9703 -.0653	.0140 65.9882	.0624 -.8568 -.0653	.0145 66.4582	.0634 -.7182 -.0653
.0149 66.8704	.0644 -.5918 -.0653	.0153 67.2266	.0653 -.4763 -.0653	.0157 67.6181	.0664 -.3364 -.0653	.0162 67.9209	.0675 -.2098 -.0653	.0167 68.1823	.0688 -.0580 -.0653
.0174 68.2605	.0705 .1210 -.0653	.0180 67.9678	.0720 .2774 -.0653	.0183 67.5992	.0729 .3552 -.0653	.0187 67.0684	.0736 .4250 -.0653	.0190 66.3577	.0744 .4874 -.0653
.0192 65.6949	.0749 .5294 -.0653	.0194 64.9224	.0753 .5674 -.0653	.0196 64.0349	.0758 .6016 -.0653	.0198 63.0271	.0762 .6323 -.0653	.0200 61.8938	.0765 .6598 -.0653

.0202	.0769	.0204	.0772	.0205	.0775	.0207	.0778	.0209	.0780
60.6303	.6843	59.2340	.7060	57.7045	.7250	56.0456	.7414	54.2670	.7555
	-.0653		-.0653		-.0653		-.0653		-.0653
.0210	.0783	.0212	.0785	.0214	.0787	.0215	.0789	.0217	.0790
52.3848	.7672	50.4227	.7767	48.4114	.7840	46.3868	.7894	44.3864	.7930
	-.0653		-.0653		-.0653		-.0653		-.0653
.0218	.0792	.0220	.0793	.0222	.0795	.0223	.0796	.0225	.0797
42.4464	.7949	40.5975	.7954	38.2875	.7943	36.1694	.7913	34.2653	.7870
	-.0653		-.0653		-.0653		-.0653		-.0653
.0227	.0798	.0229	.0800	.0231	.0801	.0233	.0802	.0236	.0804
32.5790	.7817	30.6272	.7734	28.9751	.7642	27.1506	.7511	25.6800	.7375
	-.0653		-.0653		-.0653		-.0653		-.0653
.0239	.0805	.0242	.0806	.0246	.0808	.0249	.0809	.0254	.0811
24.1296	.7191	22.9347	.7009	21.7234	.6773	20.8220	.6546	19.9341	.6260
	-.0653		-.0653		-.0652		-.0652		-.0652
.0259	.0813	.0266	.0816	.0273	.0818	.0281	.0820	.0291	.0824
19.1008	.5904	18.3578	.5468	17.8654	.5070	17.4238	.4590	17.0421	.4021
	-.0652		-.0652		-.0652		-.0652		-.0652
.0304	.0828	.0321	.0832	.0341	.0838	.0360	.0844	.0384	.0851
16.7188	.3356	16.4428	.2592	16.1944	.1732	16.0080	.1010	15.8019	.0204
	-.0652		-.0652		-.0652		-.0652		-.0652
.0406	.0857								
15.6224	-.0466								
	-.0652								

DLAM = .10000E-04 DMU = -.58750E-05

NUMBER OF SEGMENTS = 10 SEGMENT LENGTH = .14826E-01 ERROR = .12590E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.302327	.953204	107.597394	133.602820	26.005426
2	-.070000	.045500	.029720	.999558	88.296934	90.000001	1.703067
3	-.070000	.060200	.414714	.909952	65.498683	70.346177	4.847494
4	-.065000	.074200	.694138	.719842	46.041423	47.202599	1.161175
5	-.055000	.085000	.867030	.498256	29.884661	47.726311	17.841650
6	-.045000	.096000	.952108	.305761	17.803941	15.838000	-1.965941
7	-.030900	.100000	.993462	.114161	6.555261	-7.645013	-14.200274
8	-.016000	.098000	.996211	-.086973	-4.989497	-7.696052	-2.706554
9	-.001200	.096000	.975020	-.222119	-12.833527	-1.884050	10.949477
10	.014000	.095500	.930304	-.366790	-21.517785	-66.037512	-44.519726

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .63524E+00

DLAM = .30929E+00 DDU = -.32335E+00

NUMBER OF SEGMENTS = 10 SEGMENT LENGTH = .12222E-01 ERROR = .11656E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.883578	.468284	152.077066	107.672395	-44.404671
2	-.063710	.046645	-.367735	.929930	111.576020	88.413109	-23.162911
3	-.063372	.058863	.266132	.963937	74.565795	65.646486	-8.919309
4	-.058332	.069998	.648246	.761431	49.590511	46.198098	-3.392414
5	-.049872	.078819	.859702	.510796	30.716880	30.035086	-.681795
6	-.039291	.084937	.964171	.265283	15.383755	17.941849	2.558094
7	-.027663	.088702	.999169	.040766	2.336386	6.675986	4.339600
8	-.015524	.090123	.986116	-.166057	-9.558659	-4.891248	4.667411
9	-.003346	.089080	.930404	-.366535	-21.502099	-12.752895	8.749205
10	.008575	.086382	.797166	-.603760	-37.139678	-21.458414	15.681264

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .13170E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .84179E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .14748E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .44905E-01
NORMALIZATION FACTOR FOR A = .99208093E+00 B = .96537245E+00

DLAM = .23092E+01 DMU = -.14298E+01

NUMBER OF SEGMENTS = 20 SEGMENT LENGTH = .67992E-02 ERROR = .13175E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.242288	.970204	104.021644	129.309477	25.287833
2	-.064307	.040261	-.313429	.949611	108.266027	129.309477	21.043450
3	-.068615	.045522	-.098283	.995159	95.640297	100.160722	4.520425
4	-.069814	.052214	.121638	.992575	83.013364	100.160722	17.147359
5	-.071014	.058907	.300666	.953729	72.502365	70.885517	-1.616848
6	-.068787	.065331	.453351	.891332	63.041097	70.885517	7.844419
7	-.066561	.071755	.582492	.812836	54.373963	48.933750	-5.440213
8	-.062094	.076881	.692457	.721459	46.175072	48.933750	2.758678
9	-.057628	.082008	.782734	.622356	38.488394	31.489770	-6.998625
10	-.051830	.085559	.855981	.517008	31.131769	31.489770	.358001
11	-.046032	.089111	.912339	.409435	24.169321	17.749383	-6.419938
12	-.039556	.091183	.953753	.300592	17.493136	17.749383	.256248
13	-.033081	.093256	.981113	.193434	11.153245	5.506964	-5.646281
14	-.026313	.093909	.996160	.087557	5.023067	5.506964	.483897
15	-.019545	.094561	.999875	-.015805	-.905623	-6.349933	-5.444310
16	-.012788	.093809	.992942	-.118601	-6.811381	-6.349933	.461449
17	-.006030	.093057	.975407	-.220412	-12.733227	-16.419374	-3.686147
18	.000492	.091135	.945833	-.324652	-18.944520	-16.419374	2.525146
19	.007014	.089213	.897352	-.441315	-26.187819	-28.858783	-2.670964
20	.012968	.085932	.766114	-.642704	-39.993780	-28.858783	11.134997

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .13715E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .31815E+01

ABSOLUTE SUO OF CORRECTIONS TO DLAM, DMU, AND DL IS .54559E+00

DLAM = -.54192E+01 DMU = .31850E+01

NUMBER OF SEGMENTS = 20 SEGMENT LENGTH = .63111E-02 ERROR = .15255E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.591569	.806255	126.268411	103.694108	-22.574303
2	-.061494	.041132	-.876600	.481220	151.234874	108.021541	-43.213332
3	-.063447	.047133	-.507210	.861823	120.478139	95.154798	-25.323340
4	-.064014	.053419	-.124047	.992276	97.125724	82.311600	-14.814124
5	-.063169	.059673	.189608	.981860	79.070114	71.647295	-7.422820
6	-.061182	.065663	.430433	.902623	64.504967	62.073424	-2.431542
7	-.058226	.071240	.609801	.792555	52.424881	53.326914	.902034
8	-.054457	.076302	.741065	.671433	42.177758	45.075440	2.897682
9	-.050000	.080770	.835734	.549134	33.307630	37.360178	4.052547
10	-.044984	.084600	.902688	.430295	25.486300	29.995318	4.509018
11	-.039518	.087755	.948495	.316793	18.469068	23.042306	4.573238
12	-.033710	.090225	.977892	.209111	12.070275	16.390667	4.320392
13	-.027656	.092006	.994256	.107027	6.143952	10.087738	3.943785
14	-.021442	.093112	.999950	.009963	.570841	4.005447	3.434606
15	-.015147	.093552	.996557	-.082914	-4.756083	-1.866139	2.889943
16	-.008839	.093347	.984967	-.172745	-9.947466	-7.705116	2.242350
17	-.002585	.092501	.965261	-.261288	-15.146498	-13.550813	1.595686
18	.003551	.091022	.936026	-.351930	-20.605397	-19.673250	.932146
19	.009494	.088897	.890459	-.455063	-27.069015	-26.802678	.266337
20	.015127	.086051	.759526	-.650477	-40.577590	-40.367313	.210276

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .32130E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .15683E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .24026E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .51183E-01
NORMALIZATION FACTOR FOR A = .99996906E+00 B = .98030107E+00

DLAM = .52901E+00 DMU = -.31444E+00

NUMBER OF SEGMENTS = 40 SEGMENT LENGTH = .33675E-02 ERROR = .13938E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.247975	.968766	104.357735	114.904308	10.546573
2	-.061418	.038054	-.690415	.723414	133.662937	114.904308	-18.758629
3	-.062836	.041109	-.657065	.753834	131.076419	129.236066	-1.840353
4	-.064966	.043717	-.425810	.904812	115.201959	129.236066	14.034107
5	-.067096	.046325	-.245521	.969391	104.212627	107.816635	3.604007
6	-.068127	.049531	-.097230	.995262	95.579691	107.816635	12.236944
7	-.069157	.052737	.029483	.999565	88.310515	90.022413	1.711899
8	-.069158	.056105	.141317	.989964	81.875939	90.022413	8.146475
9	-.069160	.059472	.242131	.970244	75.987667	75.867656	-.120011
10	-.068337	.062738	.333904	.942607	70.494115	75.867656	5.373541
11	-.067515	.066004	.417432	.908708	65.327413	63.930084	-1.397330
12	-.066035	.069028	.493727	.869617	60.414185	63.930084	3.515899
13	-.064555	.072053	.562989	.826464	55.737239	53.596679	-2.140560
14	-.062557	.074764	.625906	.779898	51.251249	53.596679	2.345431
15	-.060558	.077474	.682548	.730848	46.956893	44.391034	-2.565859
16	-.058152	.079830	.733491	.679699	42.820161	44.391034	1.570873
17	-.055746	.082186	.778844	.627218	38.845187	36.116840	-2.728347
18	-.053025	.084170	.819101	.573650	35.005115	36.116840	1.111725
19	-.050305	.086155	.854408	.519602	31.305567	28.524629	-2.780938
20	-.047346	.087764	.885206	.465200	27.723166	28.524629	.801464
21	-.044387	.089372	.911672	.410918	24.262514	21.527828	-2.734686
22	-.041255	.090607	.934185	.356788	20.903071	21.527828	.624756
23	-.038122	.091843	.952935	.303175	17.648429	14.981721	-2.666708
24	-.034869	.092714	.968237	.250035	14.479613	14.981721	.502108
25	-.031616	.093584	.980274	.197642	11.399090	8.839190	-2.559900
26	-.028289	.094102	.989308	.145897	8.389250	8.839190	.449940
27	-.024961	.094619	.995478	.094995	5.451019	2.977957	-2.473062
28	-.021598	.094794	.998996	.044789	2.567096	2.977957	.410861
29	-.018235	.094969	.999989	-.004594	-.263240	-2.659694	-2.396453
30	-.014871	.094813	.998576	-.053343	-3.057759	-2.659694	.398065
31	-.011507	.094656	.994844	-.101422	-5.821039	-8.217965	-2.396926
32	-.008174	.094175	.988824	-.149084	-8.573872	-8.217965	.355907
33	-.004842	.093694	.980520	-.196417	-11.327529	-13.788754	-2.461225
34	-.001571	.092891	.969824	-.243805	-14.111235	-13.788754	.322481
35	.001699	.092088	.956519	-.291669	-16.957923	-19.635514	-2.677591
36	.004871	.090957	.940122	-.340838	-19.927967	-19.635514	.292453
37	.008043	.089825	.919475	-.393149	-23.150606	-26.504706	-3.354100
38	.011056	.088322	.892022	-.451992	-26.871587	-26.504706	.366882
39	.014070	.086820	.845355	-.534205	-32.290007	-40.210919	-7.920911
40	.016642	.084646	.683362	-.730080	-46.893084	-40.210919	6.682166

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .13482E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .31460E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .16157E+00

DLAM = -.62839E+01 DMU = .36914E+01

NUMBER OF SEGMENTS = 40 SEGMENT LENGTH = .32982E-02 ERROR = .74688E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.379992	.924990	112.333179	104.164849	-8.168330
2	-.060807	.038198	-.689044	.724720	133.554455	133.822409	.267954
3	-.063091	.040577	-.580102	.814544	125.457704	131.205135	5.747432
4	-.065264	.043059	-.428641	.903475	115.381364	115.138432	-.242932
5	-.066665	.046045	-.301141	.953579	107.526172	104.018048	-3.508124
6	-.067464	.049245	-.176715	.984262	100.178483	95.287100	-4.891383
7	-.067767	.052529	-.053959	.998543	93.093123	87.940618	-5.152505
8	-.067649	.055825	.065019	.997884	86.272069	81.442689	-4.829380
9	-.067158	.059086	.177974	.984035	79.748206	75.501343	-4.246862
10	-.066332	.062279	.283330	.959022	73.540922	69.963011	-3.577910
11	-.065202	.065378	.380229	.924892	67.652118	64.758743	-2.893375
12	-.063796	.068361	.468348	.883544	62.072865	59.814161	-2.258703
13	-.062138	.071212	.547741	.836648	56.787860	55.111537	-1.676323
14	-.060251	.073917	.618699	.785628	51.778783	50.604889	-1.173894
15	-.058158	.076466	.681665	.731664	47.026074	46.294514	-.731560
16	-.055879	.078850	.737157	.675722	42.510206	42.145901	-.364305
17	-.053434	.081064	.785725	.618576	38.212226	38.162844	-.049382
18	-.050840	.083102	.827922	.560843	34.114115	34.318097	.203981
19	-.048116	.084961	.864285	.503003	30.198849	30.616969	.418120
20	-.045278	.086641	.895320	.445424	26.450491	27.035773	.585282
21	-.042340	.088140	.921497	.388386	22.854097	23.578832	.724735
22	-.039317	.089459	.943248	.332090	19.395690	20.225366	.829677
23	-.036223	.090599	.960962	.276679	16.062115	16.978722	.916607
24	-.033068	.091563	.974991	.222246	12.840970	13.819752	.978782
25	-.029866	.092350	.985643	.168841	9.720430	10.750716	1.030286
26	-.026625	.092966	.993193	.116482	6.689140	7.753886	1.064747
27	-.023357	.093411	.997875	.065159	3.736001	4.830025	1.094024
28	-.020071	.093688	.999890	.014835	.849990	1.961769	1.111779
29	-.016775	.093801	.999403	-.034553	-1.980154	-.851725	1.128429
30	-.013477	.093752	.996542	-.083091	-4.766276	-3.628232	1.138044
31	-.010185	.093543	.991396	-.130897	-7.521418	-6.372393	1.149024
32	-.006907	.093177	.984007	-.178127	-10.260692	-9.104933	1.155759
33	-.003651	.092656	.974359	-.224999	-13.002847	-11.837092	1.165755
34	-.000423	.091979	.962346	-.271828	-15.773045	-14.597898	1.175147
35	.002769	.091148	.947720	-.319103	-18.608668	-17.420009	1.188659
36	.005916	.090160	.929955	-.367673	-21.572166	-20.363230	1.208936
37	.009008	.089013	.907879	-.419232	-24.786082	-23.555479	1.230603
38	.012031	.087695	.878435	-.477861	-28.545814	-27.239831	1.305983
39	.014964	.086185	.830034	-.557713	-33.897810	-32.602261	1.295548
40	.017742	.084408	.644785	-.764364	-49.850428	-47.042427	2.808000

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .81557E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .23075E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .11289E+00
NORMALIZATION FACTOR FOR A = .99991230E+00 B = .99938003E+00

DLAM = -.44064E+01 DMU = .25884E+01

NUMBER OF SEGMENTS = 80 SEGMENT LENGTH = .16716E-02 ERROR = .12978E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.286582	.958056	106.653464	108.228856	1.575392
2	-.060523	.036588	-.431873	.901935	115.586468	108.228856	-7.357612
3	-.061046	.038175	-.624068	.781370	128.613809	133.720937	5.107128
4	-.062201	.039384	-.634824	.772656	129.406967	133.720937	4.313969
5	-.063356	.040592	-.585508	.810667	125.838858	128.343969	2.505111
6	-.064393	.041903	-.510525	.859863	120.698835	128.343969	7.645134
7	-.065431	.043214	-.439921	.898036	116.098857	115.247305	-.851553
8	-.066144	.044726	-.373355	.927689	111.922680	115.247305	3.324624
9	-.066856	.046238	-.306454	.951886	107.845647	105.740149	-2.105497
10	-.067310	.047846	-.240325	.970693	103.905707	105.740149	1.834443
11	-.067763	.049455	-.174721	.984618	100.062428	97.683259	-2.379169
12	-.067987	.051112	-.110324	.993896	96.333980	97.683259	1.349279
13	-.068210	.052769	-.047366	.998878	92.714889	90.451172	-2.263717
14	-.068224	.054440	.013828	.999904	89.207670	90.451172	1.243503
15	-.068237	.056112	.073033	.997329	85.811769	83.777075	-2.034694
16	-.068056	.057773	.130141	.991495	82.522249	83.777075	1.254826
17	-.067874	.059435	.185003	.982738	79.338713	77.531575	-1.807138
18	-.067513	.061067	.237629	.971356	76.253347	77.531575	1.278229
19	-.067153	.062700	.287939	.957649	73.265400	71.647629	-1.617772
20	-.066626	.064286	.336084	.941861	70.366401	71.647629	1.281227
21	-.066100	.065873	.381790	.924249	67.555370	66.091666	-1.463703
22	-.065422	.067401	.425397	.905007	64.824206	66.091666	1.267460
23	-.064745	.068929	.466815	.884355	62.172281	60.821934	-1.350347
24	-.063930	.070389	.506152	.862444	59.592113	60.821934	1.229821
25	-.063115	.071848	.543421	.839460	57.083143	55.821787	-1.261356
26	-.062176	.073231	.578731	.815518	54.638659	55.821787	1.183128
27	-.061237	.074614	.612101	.790780	52.258432	51.058931	-1.199501
28	-.060186	.075914	.643637	.765331	49.936424	51.058931	1.122507
29	-.059136	.077214	.673368	.739308	47.672472	46.523595	-1.148877
30	-.057986	.078427	.701392	.712776	45.461196	46.523595	1.062399
31	-.056835	.079640	.727741	.685852	43.302637	42.188624	-1.114013
32	-.055597	.080763	.752508	.658584	41.191943	42.188624	.996681
33	-.054358	.081886	.775725	.631071	39.129187	38.046313	-1.082874
34	-.053042	.082916	.797479	.603347	37.110005	38.046313	.936308
35	-.051725	.083946	.817803	.575499	35.134564	34.073916	-1.060648
36	-.050341	.084882	.836775	.547547	33.198885	34.073916	.875031
37	-.048956	.085819	.854430	.519566	31.303136	30.265474	-1.037662
38	-.047512	.086662	.870839	.491568	29.443691	30.265474	.821782
39	-.046069	.087504	.886036	.463617	27.620747	26.601085	-1.019662
40	-.044574	.088253	.900084	.435717	25.830954	26.601085	.770131
41	-.043079	.089001	.913016	.407924	24.074494	23.075360	-.999134
42	-.041541	.089656	.924890	.380236	22.348273	23.075360	.727087
43	-.040004	.090311	.935737	.352699	20.652460	19.670854	-.981606
44	-.038430	.090874	.945609	.325307	18.984156	19.670854	.686698
45	-.036855	.091437	.954535	.298100	17.343509	16.382603	-.960906
46	-.035252	.091908	.962560	.271068	15.727800	16.382603	.654803
47	-.033648	.092380	.969714	.244244	14.137138	13.194783	-.942355
48	-.032021	.092761	.976035	.217614	12.568936	13.194783	.625847
49	-.030393	.093143	.981550	.191208	11.023272	10.102569	-.920783
50	-.028747	.093436	.986292	.165008	9.497680	10.102569	.604890
51	-.027102	.093729	.990287	.139038	7.992167	7.091386	-.900781

52	-.025443	.093936	.993563	.113279	6.504346	7.091386	.587040
53	-.023784	.094142	.996143	.087750	5.034173	4.156037	-.878137
54	-.022117	.094263	.998049	.062430	3.579315	4.156037	.576722
55	-.020450	.094384	.999303	.037335	2.139619	1.282305	-.857313
56	-.018778	.094422	.999923	.012440	.712759	1.282305	.569546
57	-.017107	.094459	.999925	-.012243	-.701517	-1.535874	-.834358
58	-.015436	.094414	.999325	-.036741	-2.105565	-1.535874	.569690
59	-.013765	.094369	.998135	-.061046	-3.499837	-4.313323	-.813486
60	-.012098	.094244	.996365	-.085187	-4.886794	-4.313323	.573471
61	-.010431	.094118	.994024	-.109164	-6.267122	-7.058884	-.791763
62	-.008773	.093913	.991115	-.133008	-7.643475	-7.058884	.584591
63	-.007114	.093707	.987642	-.156727	-9.016975	-9.790470	-.773495
64	-.005466	.093423	.983601	-.180359	-10.390642	-9.790470	.600173
65	-.003819	.093139	.978988	-.203919	-11.766231	-12.523057	-.756827
66	-.002187	.092776	.973788	-.227457	-13.147412	-12.523057	.624355
67	-.000555	.092414	.967985	-.251007	-14.537131	-15.283705	-.746574
68	.001057	.091973	.961548	-.274636	-15.940295	-15.283705	.656589
69	.002670	.091533	.954438	-.298409	-17.362051	-18.107369	-.745318
70	.004258	.091013	.946594	-.322428	-18.809811	-18.107369	.702442
71	.005847	.090493	.937929	-.346829	-20.293461	-21.055050	-.761589
72	.007407	.089893	.928316	-.371792	-21.826152	-21.055050	.771103
73	.008967	.089292	.917549	-.397624	-23.429703	-24.251714	-.822010
74	.010491	.088606	.905324	-.424722	-25.133096	-24.251714	.881382
75	.012015	.087919	.891049	-.453907	-26.994647	-27.965928	-.971281
76	.013492	.087135	.873765	-.486349	-29.100893	-27.965928	1.134965
77	.014968	.086351	.850930	-.525279	-31.687009	-33.311472	-1.624463
78	.016365	.085433	.817048	-.576570	-35.209629	-33.311472	1.898157
79	.017762	.084515	.743861	-.668335	-41.938664	-48.470362	-6.531698
80	.018870	.083264	.415191	-.909734	-65.468680	-48.470362	16.998319

NORMALIZATION FACTOR FOR A = .99988601E+00 B = .10000304E+01

AFTER 1 ITERATIONS , THE RELATIVE CHANGE IN SEGMENT LENGTH = .41804E-04

DLAM = .10000E-04 DMU = -.58750E-05

NUMBER OF SEGMENTS = 10 SEGMENT LENGTH = .16142E-01 ERROR = .32681E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.881456	-.472267	-28.181534	-90.000001	-61.818467
2	-.060000	.019000	.984745	-.174002	-10.020559	-34.695154	-24.674595
3	-.047000	.010000	.996333	.085566	4.908567	0.000000	-4.908567
4	-.030800	.010000	.897211	.441602	26.206164	18.097257	-8.108907
5	-.015500	.015000	.688228	.725495	46.510015	17.878697	-28.631319
6	0.000000	.020000	.657503	.753452	48.890286	35.537678	-13.352608
7	.014000	.030000	.720415	.693543	43.911265	68.198591	24.287326
8	.020000	.045000	.725964	.687732	43.450881	50.659482	7.208602
9	.030000	.057200	.786637	.617416	38.127686	79.249034	41.121348
10	.033000	.073000	.842363	.538910	32.609473	145.304848	112.695375

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .12375E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .16498E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .38938E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .15671E+00

DLAM = .15806E+02 DMU = -.92761E+01

NUMBER OF SEGMENTS = 10 SEGMENT LENGTH = .10280E-01 ERROR = .37670E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	-.491614	-.870813	-119.446724	-25.127214	94.319511
2	-.050693	.030635	.456627	-.889658	-62.830323	-5.657744	57.172579
3	-.040463	.029621	.965205	-.261495	-15.158769	9.965898	25.124667
4	-.030339	.031400	.951269	.308363	17.960583	31.602185	13.641683
5	-.021583	.036787	.746531	.665351	41.709224	51.544100	9.834876
6	-.015190	.044037	.491926	.870637	60.532759	53.842279	-6.690480
7	-.009125	.053137	.258139	.966108	75.040349	49.025996	-26.014354
8	-.002304	.060898	.065991	.997820	86.216254	48.578981	-37.637353
9	.004417	.068687	-.077438	.996997	94.441332	43.386982	-51.054350
10	.011887	.075668	-.055342	.998467	93.172589	37.959846	-55.212663

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .18131E+03

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .11037E+03

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .28146E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .18575E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .51432E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .15829E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .98018E-01
NORMALIZATION FACTOR FOR A = .88409451E+00 B = .96884714E+00

DLAM = -.67196E+01 DMU = .39524E+01

NUMBER OF SEGMENTS = 20 SEGMENT LENGTH = .63473E-02 ERROR = .94691E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.317376	-.948300	-71.495679	-66.143698	5.351980
2	-.057433	.029195	.698588	-.715524	-45.686143	-66.143698	-20.457555
3	-.054866	.023390	.860401	-.509617	-30.638354	-33.876870	-3.238516
4	-.049596	.019852	.950956	-.309326	-18.018634	-33.876870	-15.858236
5	-.044326	.016314	.993236	-.116114	-6.667896	-7.355452	-.687556
6	-.038031	.015581	.997095	.076172	4.368577	-7.355452	-11.724029
7	-.031736	.014689	.967267	.253761	14.700171	17.385743	2.685572

8	-.025679	.016585	.910508	.413492	24.424374	17.385743	-7.038631
9	-.019621	.018482	.841891	.539647	32.659607	38.342195	5.682588
10	-.014643	.022419	.775879	.630882	39.115234	38.342195	-.773039
11	-.009664	.026357	.718261	.695774	44.088910	49.112971	5.024061
12	-.005510	.031156	.669235	.743051	47.991947	49.112971	1.121024
13	-.001355	.035954	.623487	.781833	51.428750	54.844645	3.415896
14	.002300	.041144	.577808	.816173	54.703473	54.844645	.141172
15	.005955	.046333	.529636	.848225	58.019158	61.389169	3.370011
16	.008994	.051906	.478159	.878273	61.434748	61.389169	-.045579
17	.012034	.057478	.431901	.901921	64.411729	64.058811	-.352918
18	.014810	.063186	.403344	.915048	66.212605	64.058811	-2.153794
19	.017587	.068893	.441591	.897216	63.794563	60.517723	-3.276840
20	.020711	.074419	.526567	.850134	58.226221	60.517723	2.291503

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .24036E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .41334E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .63534E+00

DLAM = .74461E+01 DMU = -.43714E+01

NUMBER OF SEGMENTS = 20 SEGMENT LENGTH = .59342E-02 ERROR = .12412E+03

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.413795	-.910370	-65.556530	-71.806446	-6.249916
2	-.058147	.029362	.615747	-.787944	-51.993772	-45.324589	6.669183
3	-.053975	.025143	.768616	-.639710	-39.770224	-29.911697	9.858527
4	-.048831	.022183	.894881	-.446304	-26.506812	-17.025850	9.480962
5	-.043157	.020446	.973091	-.230420	-13.321807	-5.478771	7.843037
6	-.037250	.019879	.999931	-.011764	-.674039	5.702013	6.376052
7	-.031345	.020469	.981123	.193385	11.150373	16.122617	4.972244
8	-.025644	.022117	.928365	.371670	21.818677	25.887678	4.069001
9	-.020305	.024708	.856901	.515481	31.029582	34.124420	3.094838
10	-.015393	.028037	.780471	.625192	38.696254	40.560175	1.863921
11	-.010885	.031895	.707251	.706962	44.988284	45.506182	.517898
12	-.006726	.036129	.639344	.768921	50.257108	49.380162	-.876946
13	-.002862	.040633	.575296	.817945	54.879614	52.786171	-2.093442
14	.000727	.045359	.512563	.858649	59.165273	56.027140	-3.138133
15	.004043	.050280	.448608	.893729	63.345609	59.304387	-4.041222
16	.007072	.055383	.381518	.924361	67.572245	62.676046	-4.896199
17	.009796	.060655	.310734	.950497	71.896552	65.611278	-6.285274
18	.012246	.066060	.238607	.971116	76.195656	67.385386	-8.810270
19	.014528	.071538	.182989	.983115	79.456091	65.003026	-14.453066
20	.017036	.076916	.207224	.978294	78.040283	59.508910	-18.531373

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .27443E+02

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .89872E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .56065E+00
NORMALIZATION FACTOR FOR A = .99615870E+00 B = .99585280E+00

DLAM = .14626E+01 DMU = -.84710E+00

NUMBER OF SEGMENTS = 40 SEGMENT LENGTH = .30656E-02 ERROR = .93846E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.272901	-.962042	-74.163055	-68.630584	5.532470
2	-.058883	.032145	.457558	-.889180	-62.770340	-68.630584	-5.860245
3	-.057766	.029290	.576884	-.816826	-54.768357	-48.719764	6.048593
4	-.055743	.026987	.676670	-.736287	-47.416048	-48.719764	-1.303716
5	-.053721	.024683	.766715	-.641988	-39.940206	-34.999431	4.940775
6	-.051210	.022924	.842087	-.539342	-32.638874	-34.999431	-2.360557
7	-.048699	.021166	.901445	-.432893	-25.651308	-22.023125	3.628184
8	-.045857	.020017	.946029	-.324081	-18.909899	-22.023125	-3.113226
9	-.043015	.018867	.976503	-.215502	-12.445007	-9.741807	2.703201
10	-.039993	.018348	.994209	-.107462	-6.169038	-9.741807	-3.572769
11	-.036972	.017830	.999998	-.002194	-.125721	2.107495	2.233216
12	-.033908	.017942	.994981	.100068	5.743075	2.107495	-3.635580
13	-.030845	.018055	.980354	.197247	11.376040	13.186589	1.810548
14	-.027860	.018754	.957435	.288649	16.777064	13.186589	-3.590476
15	-.024875	.019454	.928098	.372337	21.859802	23.380167	1.520365
16	-.022062	.020670	.894219	.447630	26.591747	23.380167	-3.211580
17	-.019248	.021887	.857965	.513708	30.911140	32.097066	1.185926
18	-.016651	.023516	.821239	.570584	34.790981	32.097066	-2.693915
19	-.014054	.025145	.785598	.618737	38.223967	39.151947	.927980
20	-.011676	.027080	.752129	.659016	41.224839	39.151947	-2.072892
21	-.009299	.029016	.721272	.692652	43.840394	44.780382	.939988
22	-.007123	.031175	.692962	.720974	46.134955	44.780382	-1.354573
23	-.004947	.033334	.666715	.745313	48.185976	49.364052	1.178077
24	-.002951	.035661	.641852	.766829	50.069959	49.364052	-.705907
25	-.000954	.037987	.617773	.786356	51.846307	53.392292	1.545985
26	.000874	.040448	.593875	.804557	53.567540	53.392292	-.175248
27	.002702	.042909	.569719	.821840	55.269396	57.171184	1.901788
28	.004364	.045485	.544852	.838532	56.985429	57.171184	.185754
29	.006026	.048061	.518996	.854777	58.735068	60.917514	2.182447
30	.007516	.050740	.491952	.870622	60.531022	60.917514	.386492
31	.009006	.053419	.463681	.886002	62.375132	64.736659	2.361526
32	.010314	.056191	.434207	.900813	64.265148	64.736659	.471511
33	.011623	.058964	.403797	.914848	66.184213	68.389134	2.204921
34	.012752	.061814	.372596	.927993	68.124164	68.389134	.264970
35	.013881	.064664	.341382	.939924	70.038876	71.450441	1.411565
36	.014856	.067570	.310520	.950567	71.909421	71.450441	-.458979
37	.015831	.070477	.284308	.958733	73.482517	71.909542	-1.572975
38	.016783	.073391	.265791	.964031	74.586054	71.909542	-2.676512
39	.017735	.076305	.270004	.962859	74.335511	68.445732	-5.889779
40	.018861	.079156	.301091	.953596	72.476884	68.445732	-4.031152

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .85091E+01

ABSOLUTE SUO OF CORRECTIONS TO DLAO, DMU, AND DL IS .17791E+01

DLAM = -.27887E+01 DMU = .16316E+01

NUMBER OF SEGMENTS = 40 SEGMENT LENGTH = .30767E-02 ERROR = .32382E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.375742	-.926724	-67.929841	-74.092120	-6.162279
2	-.059157	.032041	.487658	-.873035	-60.813253	-62.755008	-1.941755
3	-.057748	.029306	.594359	-.804200	-53.533075	-54.792565	-1.259490
4	-.055974	.026792	.696212	-.717836	-45.876092	-47.476194	-1.600102
5	-.053895	.024524	.783162	-.621817	-38.448965	-40.035891	-1.586926
6	-.051539	.022545	.853550	-.521012	-31.400126	-32.767725	-1.367599
7	-.048952	.020880	.908612	-.417641	-24.685718	-25.809975	-1.124257
8	-.046182	.019540	.949746	-.313022	-18.241467	-19.095129	-.853662
9	-.043275	.018534	.978100	-.208137	-12.013220	-12.653337	-.640116
10	-.040273	.017860	.994588	-.103900	-5.963770	-6.397285	-.433515
11	-.037215	.017517	.999999	-.001291	-.073956	-.370588	-.296632
12	-.034138	.017497	.995133	.098542	5.655209	5.484653	-.170556
13	-.031076	.017791	.980949	.194265	11.201765	11.107138	-.094628
14	-.028057	.018384	.958699	.284422	16.524326	16.500539	-.023788
15	-.025107	.019258	.929985	.367596	21.567459	21.578336	.010877
16	-.022245	.020389	.896706	.442626	26.271572	26.307666	.036094
17	-.019487	.021753	.860879	.508810	30.584620	30.626361	.041742
18	-.016840	.023320	.824391	.566021	34.473212	34.506955	.033743
19	-.014304	.025063	.788781	.614674	37.928229	37.941693	.013464
20	-.011878	.026955	.755096	.655615	40.966273	40.944929	-.021344
21	-.009554	.028971	.723835	.689973	43.627947	43.563173	-.064775
22	-.007324	.031092	.694995	.719014	45.973171	45.860569	-.112602
23	-.005182	.033300	.668201	.743981	48.071643	47.914499	-.157144
24	-.003120	.035583	.642916	.765937	49.990381	49.801462	-.188919
25	-.001134	.037933	.618611	.785698	51.785256	51.580887	-.204369
26	.000778	.040344	.594819	.803859	53.500247	53.305346	-.194901
27	.002617	.042811	.571131	.820859	55.170857	55.010626	-.160231
28	.004381	.045331	.547186	.837011	56.825846	56.730345	-.095501
29	.006069	.047904	.522710	.852511	58.485792	58.483978	-.001815
30	.007677	.050527	.497563	.867428	60.161078	60.284277	.123199
31	.009202	.053199	.471752	.881731	61.851892	62.133103	.281211
32	.010640	.055919	.445418	.895323	63.549931	64.028214	.478283
33	.011988	.058685	.418823	.908068	65.239703	65.952719	.713016
34	.013241	.061495	.392359	.919812	66.898612	67.898434	.999823
35	.014399	.064345	.366610	.930375	68.493281	69.819093	1.325812
36	.015461	.067233	.342467	.939530	69.972768	71.695686	1.722917
37	.016427	.070154	.321613	.946871	71.239507	73.274047	2.034540
38	.017312	.073101	.307409	.951577	72.096835	74.381373	2.284538
39	.018141	.076064	.307320	.951606	72.102178	74.129962	2.027784
40	.018982	.079023	.329410	.944187	70.767011	72.265030	1.498019

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .99818E+01

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .19478E+01
NORMALIZATION FACTOR FOR A = .99992005E+00 B = .99997668E+00

DLAM = -.19940E+00 DMU = .94014E-01

NUMBER OF SEGMENTS = 80 SEGMENT LENGTH = .15296E-02 ERROR = .80951E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.286207	-.958168	-73.368984	-71.030226	2.338757
2	-.059503	.033553	.359297	-.933223	-68.942989	-71.030226	-2.087237
3	-.059006	.032107	.417629	-.908618	-65.315021	-61.788281	3.526740
4	-.058282	.030759	.476953	-.878929	-61.513429	-61.788281	-.274852
5	-.057559	.029411	.537924	-.842993	-57.457541	-54.151463	3.306078
6	-.056664	.028171	.596968	-.802265	-53.346916	-54.151463	-.804547
7	-.055768	.026931	.652265	-.757991	-49.287422	-46.648682	2.638740
8	-.054718	.025819	.703241	-.710951	-45.312378	-46.648682	-1.336304
9	-.053668	.024707	.749465	-.662044	-41.455952	-39.199816	2.256135
10	-.052482	.023740	.791137	-.611639	-37.708116	-39.199816	-1.491701
11	-.051297	.022773	.828360	-.560196	-34.069385	-32.027600	2.041784
12	-.050000	.021962	.861420	-.507894	-30.523659	-32.027600	-1.503941
13	-.048703	.021151	.890471	-.455040	-27.067491	-25.179256	1.888235
14	-.047319	.020500	.915765	-.401715	-23.685450	-25.179256	-1.493806
15	-.045935	.019849	.937421	-.348198	-20.377142	-18.588828	1.788314
16	-.044485	.019362	.955647	-.294516	-17.128501	-18.588828	-1.460328
17	-.043035	.018874	.970547	-.240914	-13.940465	-12.244256	1.696209
18	-.041540	.018550	.982283	-.187403	-10.801255	-12.244256	-1.443002
19	-.040046	.018225	.990949	-.134239	-7.714638	-6.083241	1.631397
20	-.038525	.018063	.996679	-.081437	-4.671157	-6.083241	-1.412084
21	-.037003	.017901	.999572	-.029247	-1.675972	-.117987	1.557985
22	-.035474	.017898	.999751	.022293	1.277372	-.117987	-1.395359
23	-.033944	.017895	.997338	.072911	4.181193	5.679960	1.498766
24	-.032422	.018046	.992463	.122542	7.038833	5.679960	-1.358874
25	-.030900	.018198	.985285	.170918	9.841205	11.268989	1.427784
26	-.029400	.018497	.975961	.217945	12.588340	11.268989	-1.319350
27	-.027900	.018795	.964697	.263363	15.269693	16.630285	1.360591
28	-.026434	.019233	.951688	.307067	17.882542	16.630285	-1.252257
29	-.024969	.019671	.937181	.348844	20.416627	21.692950	1.276323
30	-.023547	.020236	.921406	.388601	22.867478	21.692950	-1.174528
31	-.022126	.020802	.904632	.426193	25.226217	26.410884	1.184667
32	-.020756	.021482	.887106	.461566	27.488204	26.410884	-1.077320
33	-.019386	.022163	.869084	.494664	29.647596	30.727149	1.079553
34	-.018071	.022944	.850801	.525487	31.701052	30.727149	-.973903
35	-.016756	.023726	.832473	.554066	33.646401	34.611506	.965105
36	-.015497	.024595	.814292	.580456	35.482628	34.611506	-.871122
37	-.014238	.025463	.796411	.604756	37.211253	38.055705	.844453
38	-.013034	.026406	.778965	.627067	38.834090	38.055705	-.778385
39	-.011830	.027349	.762037	.647533	40.355892	41.075387	.719495
40	-.010677	.028354	.745692	.666291	41.781422	41.075387	-.706035
41	-.009523	.029359	.729948	.683502	43.117934	43.714229	.596295
42	-.008418	.030416	.714805	.699324	44.372768	43.714229	-.658539
43	-.007312	.031473	.700229	.713918	45.554614	46.034344	.479730
44	-.006250	.032574	.686171	.727440	46.672201	46.034344	-.637857
45	-.005188	.033675	.672572	.740032	47.734150	48.109309	.375159
46	-.004167	.034814	.659360	.751828	48.748948	48.109309	-.639639
47	-.003146	.035953	.646472	.762938	49.723874	50.010894	.287020

48	-.002163	.037125	.633839	.773465	50.666074	50.010894	-.655180
49	-.001180	.038296	.621406	.783489	51.581147	51.796743	.215596
50	-.000234	.039498	.609110	.793085	52.474788	51.796743	-.678044
51	.000712	.040700	.596906	.802311	53.351353	53.515122	.163768
52	.001622	.041930	.584739	.811222	54.215462	53.515122	-.700340
53	.002531	.043160	.572566	.819859	55.070658	55.201659	.131000
54	.003404	.044416	.560341	.828262	55.920611	55.201659	-.718952
55	.004277	.045672	.548029	.836459	56.768079	56.887525	.119447
56	.005113	.046953	.535597	.844473	57.615563	56.887525	-.728038
57	.005948	.048235	.523023	.852319	58.464755	58.592732	.127978
58	.006745	.049540	.510291	.860002	59.316773	58.592732	-.724041
59	.007543	.050846	.497396	.867523	60.172110	60.328836	.156726
60	.008300	.052175	.484343	.874878	61.030568	60.328836	-.701732
61	.009057	.053504	.471143	.882057	61.891496	62.096860	.205365
62	.009773	.054856	.457818	.889046	62.753629	62.096860	-.656769
63	.010489	.056207	.444396	.895831	63.615324	63.891541	.276217
64	.011162	.057581	.430914	.902393	64.474451	63.891541	-.582910
65	.011835	.058954	.417419	.908714	65.328241	65.696706	.368465
66	.012464	.060348	.403966	.914774	66.173665	65.696706	-.476959
67	.013094	.061743	.390625	.920550	67.006597	67.496996	.490399
68	.013679	.063156	.377473	.926021	67.822751	67.496996	-.325755
69	.014265	.064569	.364629	.931153	68.615264	69.252639	.637375
70	.014807	.065999	.352208	.935922	69.377588	69.252639	-.124949
71	.015349	.067430	.340403	.940280	70.098552	70.928737	.830186
72	.015848	.068875	.329402	.944190	70.767490	70.928737	.161247
73	.016348	.070321	.319623	.947545	71.359861	72.349575	.989714
74	.016812	.071779	.311412	.950275	71.855654	72.349575	.493921
75	.017276	.073236	.305602	.952159	72.205627	73.330718	1.125090
76	.017715	.074702	.302834	.953043	72.372097	73.330718	.958621
77	.018153	.076167	.305209	.952285	72.229259	73.207677	.978417
78	.018595	.077631	.313580	.949562	71.724882	73.207677	1.482795
79	.019037	.079096	.328907	.944362	70.797568	71.609462	.811895
80	.019520	.080547	.319417	.947614	71.372321	71.609462	.237141

NORMALIZATION FACTOR FOR A = .10004392E+01 B = .10020818E+01

AFTER 1 ITERATIONS , THE RELATIVE CHANGE IN SEGMENT LENGTH = .12605E-02

DLAM = -.19940E+00 DMU = .94014E-01

NUMBER OF SEGMENTS = 80 SEGOENT LENGTH = .15315E-02 ERROR = .42808E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.307340	-.951600	-72.101026	-72.198882	-.097855
2	-.059532	.033542	.367629	-.929972	-68.430516	-69.986255	-1.555739
3	-.059008	.032103	.425571	-.904925	-64.813207	-63.551604	1.261603
4	-.058325	.030731	.486766	-.873532	-60.871766	-61.651128	-.779362
5	-.057598	.029384	.548049	-.836446	-56.766753	-55.805116	.961637
6	-.056738	.028117	.606343	-.795203	-52.674440	-53.750145	-1.075705
7	-.055832	.026882	.660429	-.750889	-48.667430	-47.969340	.698090
8	-.054806	.025744	.710015	-.704187	-44.763861	-45.982138	-1.218277
9	-.053742	.024643	.754975	-.655754	-40.976847	-40.329794	.647052
10	-.052575	.023652	.795517	-.605931	-37.295844	-38.456162	-1.160318

11	-.051375	.022699	.831801	-.555074	-33.715794	-33.050957	.664837
12	-.050092	.021864	.864065	-.503380	-30.223892	-31.278347	-1.054455
13	-.048783	.021069	.892475	-.451097	-26.814070	-26.126326	.687744
14	-.047408	.020394	.917235	-.398347	-23.474890	-24.435526	-.960637
15	-.046013	.019761	.938476	-.345343	-20.202760	-19.486363	.716396
16	-.044569	.019250	.956367	-.292168	-16.987817	-17.862233	-.874416
17	-.043112	.018780	.971017	-.239009	-13.828058	-13.096104	.731953
18	-.041620	.018433	.982562	-.185935	-10.715673	-11.526661	-.810987
19	-.040119	.018127	.991097	-.133139	-7.651040	-6.902992	.748048
20	-.038599	.017943	.996739	-.080695	-4.628507	-5.381385	-.752878
21	-.037074	.017799	.999585	-.028798	-1.650232	-.901286	.748946
22	-.035543	.017775	.999748	.022457	1.286802	.575279	-.711522
23	-.034011	.017790	.997343	.072855	4.177969	4.926069	.748101
24	-.032485	.017922	.992496	.122276	7.023500	6.354808	-.668692
25	-.030963	.018092	.985358	.170499	9.816832	10.550441	.733609
26	-.029458	.018372	.976086	.217383	12.555366	11.923952	-.631414
27	-.027959	.018688	.964876	.262707	15.230747	15.945232	.714485
28	-.026487	.019109	.951926	.306329	17.838140	17.251623	-.586517
29	-.025024	.019563	.937472	.348061	20.368743	21.049975	.681232
30	-.023595	.020113	.921751	.387783	22.816595	22.275388	-.541207
31	-.022177	.020694	.905022	.425364	25.173709	25.813721	.640011
32	-.020799	.021361	.887538	.460735	27.434555	26.944718	-.489838
33	-.019433	.022055	.869551	.493844	29.593546	30.182557	.589011
34	-.018109	.022825	.851297	.524684	31.646982	31.209301	-.437681
35	-.016800	.023618	.832993	.553283	33.592522	34.124175	.531653
36	-.015532	.024478	.814833	.579696	35.429174	35.042314	-.386860
37	-.014278	.025357	.796972	.604016	37.158044	37.628754	.470710
38	-.013065	.026292	.779544	.626348	38.781170	38.440203	-.340967
39	-.011865	.027244	.762636	.646828	40.302855	40.710976	.408121
40	-.010704	.028243	.746312	.665597	41.728128	41.423774	-.304355
41	-.009556	.029256	.730592	.682814	43.063956	43.411485	.347528
42	-.008443	.030309	.715475	.698639	44.317893	44.038935	-.278958
43	-.007342	.031374	.700927	.713233	45.498590	45.789949	.291359
44	-.006274	.032471	.686898	.726753	46.614909	46.348776	-.266133
45	-.005217	.033580	.673327	.739345	47.675661	47.917266	.241605
46	-.004191	.034716	.660141	.751142	48.689373	48.424699	-.264674
47	-.003174	.035862	.647274	.762258	49.663627	49.862988	.199361
48	-.002187	.037033	.634656	.772795	50.605517	50.334121	-.271396
49	-.001210	.038212	.622228	.782836	51.520951	51.684616	.163666
50	-.000260	.039414	.609931	.792455	52.415507	52.131470	-.284037
51	.000680	.040623	.597712	.801711	53.293793	53.428977	.135183
52	.001593	.041853	.585520	.810658	54.160292	53.861066	-.299226
53	.002496	.043089	.573309	.819339	55.018724	55.131968	.113244
54	.003371	.044346	.561034	.827793	55.872641	55.556980	-.315660
55	.004238	.045609	.548660	.836046	56.724883	56.823684	.098802
56	.005076	.046891	.536153	.844121	57.577869	57.247464	-.330404
57	.005904	.048179	.523491	.852031	58.433286	58.524702	.091416
58	.006704	.049485	.510660	.859783	59.292201	58.950752	-.341450
59	.007494	.050797	.497655	.867375	60.155027	60.246512	.091485
60	.008254	.052127	.484481	.874802	61.021519	60.675783	-.345736
61	.009004	.053462	.471152	.882052	61.890906	61.990303	.099396
62	.009723	.054814	.457690	.889112	62.761844	62.421413	-.348430
63	.010432	.056172	.444126	.895964	63.632536	63.749645	.117109
64	.011110	.057545	.430500	.902590	64.500689	64.179254	-.321435
65	.011777	.058924	.416861	.908970	65.363404	65.508777	.145374

66	.012412	.060318	.403269	.915082	66.217316	65.931536	-.285780
67	.013036	.061716	.389798	.920901	67.058096	67.248193	.190097
68	.013629	.063129	.376535	.926402	67.880781	67.656316	-.224465
69	.014211	.064545	.363603	.931554	68.678365	68.930440	.252074
70	.014761	.065974	.351140	.936323	69.442939	69.311646	-.131293
71	.015303	.067407	.339349	.940661	70.162782	70.510220	.347438
72	.015813	.068851	.328467	.944515	70.824243	70.844729	.020486
73	.016316	.070298	.318903	.947787	71.403392	71.851369	.447977
74	.016793	.071753	.311116	.950372	71.873486	72.099296	.225810
75	.017264	.073210	.305944	.952049	72.185007	72.764876	.579869
76	.017718	.074673	.304346	.952561	72.281161	72.848120	.566959
77	.018169	.076137	.308392	.951259	72.037641	72.715173	.677531
78	.018624	.077599	.320193	.947352	71.325412	72.462953	1.137541
79	.019086	.079059	.343355	.939206	69.918584	71.200132	1.281548
80	.019579	.080509	.347491	.937683	69.666067	71.487544	1.821477

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .52194E+01
NORMALIZATION FACTOR FOR A = .10000291E+01 B = .10003613E+01

AFTER 2 ITERATIONS , THE RELATIVE CHANGE IN SEGMENT LENGTH = .91269E-03

DLAM = .11254E+01 DMU = -.11909E+01

NUMBER OF SEGMENTS = 80 SEGMENT LENGTH = .15301E-02 ERROR = .22221E+02

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.307067	-.951688	-72.117423	-72.185123	-.067700
2	-.059532	.033543	.362594	-.931947	-68.740413	-69.239140	-.498727
3	-.058989	.032112	.422157	-.906523	-65.029140	-64.208686	.820454
4	-.058324	.030735	.485086	-.874466	-60.981869	-61.282734	-.300865
5	-.057589	.029393	.547001	-.837132	-56.838493	-56.301911	.536582
6	-.056740	.028120	.605414	-.795911	-52.741351	-53.222894	-.481544
7	-.055823	.026894	.659461	-.751739	-48.741234	-48.323671	.417563
8	-.054806	.025751	.708964	-.705244	-44.849288	-45.373080	-.523792
9	-.053731	.024662	.753910	-.656978	-41.069818	-40.648352	.421466
10	-.052570	.023665	.794480	-.607290	-37.393790	-37.866148	-.472358
11	-.051362	.022726	.830843	-.556508	-33.814614	-33.368806	.445808
12	-.050084	.021885	.863203	-.504858	-30.321898	-30.732007	-.410109
13	-.048769	.021103	.891728	-.452572	-26.908800	-26.446718	.462082
14	-.047399	.020421	.916599	-.399807	-23.566105	-23.927532	-.361427
15	-.046000	.019801	.937957	-.346753	-20.288819	-19.812866	.475953
16	-.044561	.019282	.955951	-.293526	-17.069196	-17.389481	-.320285
17	-.043101	.018825	.970700	-.240293	-13.903812	-13.422865	.480947
18	-.041612	.018469	.982330	-.187155	-10.786782	-11.078451	-.291669
19	-.040111	.018175	.990944	-.134276	-7.716778	-7.230977	.485800
20	-.038593	.017983	.996652	-.081764	-4.689973	-4.955761	-.265788
21	-.037068	.017851	.999556	-.029788	-1.706991	-1.223607	.483383
22	-.035538	.017818	.999768	.021528	1.233549	.985978	-.247570
23	-.034008	.017844	.997405	.071995	4.128597	4.609558	.480962
24	-.032483	.017967	.992595	.121469	6.976917	6.749110	-.227807
25	-.030964	.018147	.985487	.169750	9.773272	10.245820	.472548
26	-.029458	.018419	.976244	.216676	12.513849	12.303882	-.209967
27	-.027963	.018745	.965055	.262046	15.191481	15.654063	.462583

28	-.026490	.019158	.952127	.305702	17.800402	17.612620	-.187782
29	-.025031	.019621	.937691	.347470	20.332659	20.778579	.445919
30	-.023601	.020164	.921987	.387221	22.781700	22.616515	-.165185
31	-.022188	.020752	.905271	.424834	25.140126	25.565371	.425245
32	-.020808	.021413	.887798	.460233	27.402135	27.262265	-.139870
33	-.019448	.022114	.869819	.493371	29.562416	29.961503	.399088
34	-.018122	.022878	.851569	.524243	31.617263	31.502280	-.114984
35	-.016817	.023677	.833265	.552874	33.564385	33.933048	.368663
36	-.015548	.024532	.815098	.579323	35.402954	35.310895	-.092059
37	-.014299	.025416	.797226	.603681	37.133984	37.468904	.334920
38	-.013085	.026347	.779778	.626056	38.759745	38.686462	-.073284
39	-.011890	.027303	.762844	.646582	40.284393	40.582888	.298495
40	-.010728	.028299	.746485	.665402	41.713166	41.652060	-.061106
41	-.009585	.029316	.730724	.682673	43.052906	43.313915	.261009
42	-.008472	.030365	.715555	.698556	44.311281	44.254651	-.056630
43	-.007376	.031433	.700948	.713212	45.496872	45.720513	.223641
44	-.006307	.032529	.686852	.726797	46.618544	46.558063	-.060481
45	-.005255	.033640	.673205	.739456	47.685102	47.872634	.187533
46	-.004229	.034775	.659936	.751322	48.705000	48.633135	-.071866
47	-.003217	.035923	.646978	.762509	49.685881	49.839315	.153434
48	-.002231	.037092	.634262	.773118	50.634742	50.545718	-.089024
49	-.001258	.038274	.621728	.783233	51.557591	51.678557	.120966
50	-.000309	.039474	.609316	.792927	52.459922	52.349121	-.110801
51	.000625	.040686	.596975	.802260	53.346446	53.436860	.090414
52	.001537	.041915	.584652	.811284	54.221575	54.085982	-.135593
53	.002434	.043154	.572302	.820043	55.089115	55.150462	.061347
54	.003309	.044410	.559879	.828574	55.952567	55.789724	-.162844
55	.004169	.045675	.547347	.836986	56.814825	56.848977	.034153
56	.005006	.046956	.534673	.845059	57.678289	57.487124	-.191165
57	.005828	.048247	.521833	.853047	58.544689	58.553199	.008510
58	.006626	.049552	.508814	.860876	59.415121	59.195410	-.219711
59	.007410	.050866	.495610	.868545	60.290042	60.274415	-.015627
60	.008169	.052195	.482223	.876048	61.169287	60.921990	-.247297
61	.008912	.053532	.468668	.883375	62.052153	62.013618	-.038535
62	.009630	.054883	.454964	.890510	62.937411	62.664300	-.273110
63	.010333	.056243	.441140	.897438	63.823372	63.763404	-.059968
64	.011009	.057615	.427234	.904141	64.707877	64.411911	-.295966
65	.011670	.058995	.413292	.910599	65.588221	65.507642	-.080579
66	.012305	.060388	.399370	.916790	66.461189	66.145577	-.315612
67	.012924	.061787	.385540	.922691	67.322736	67.223886	-.098850
68	.013516	.063198	.371886	.928279	68.168046	67.838875	-.329171
69	.014093	.064615	.358520	.933522	68.990662	68.874312	-.116349
70	.014645	.066042	.345580	.938389	69.782813	69.446791	-.336022
71	.015182	.067475	.333254	.942837	70.533600	70.405600	-.128000
72	.015695	.068917	.321780	.946815	71.229409	70.903209	-.326200
73	.016196	.070363	.311525	.950238	71.848857	71.695767	-.153090
74	.016676	.071815	.302947	.953007	72.365296	72.054499	-.310797
75	.017148	.073271	.296795	.954941	72.734809	72.542862	-.191947
76	.017607	.074731	.294016	.955801	72.901470	72.632503	-.268967
77	.018063	.076191	.296307	.955093	72.764079	72.444416	-.319663
78	.018525	.077650	.305454	.952207	72.214515	71.962614	-.251901
79	.018999	.079105	.322700	.946501	71.173742	70.628594	-.545148
80	.019506	.080549	.314112	.949386	71.692788	70.646181	-1.046607

31798

ABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL IS .35178E+01
NORMALIZATION FACTOR FOR A = .99957140E+00 B = .10000390E+01

AFTER 3 ITERATIONS , THE RELATIVE CHANGE IN SEGMENT LENGTH = .55882E-03

DLAM = .37256E+00 DMU = -.18482E+00

NUMBER OF SEGMENTS = 80 SEGMENT LENGTH = .15310E-02 ERROR = .92063E+01

I	X(I)	Y(I)	U(I)	V(I)	THETA UV	THETA AB	ERR(I)
1	-.060000	.035000	.306211	-.951964	-72.168956	-72.136687	.032269
2	-.059530	.033543	.360380	-.932805	-68.876447	-68.976011	-.099564
3	-.058981	.032114	.421473	-.906841	-65.072376	-64.606102	.466274
4	-.058325	.030731	.485343	-.874324	-60.965075	-61.120593	-.155518
5	-.057585	.029390	.547484	-.836816	-56.805426	-56.559684	.245742
6	-.056741	.028112	.605877	-.795558	-52.708004	-52.972834	-.264830
7	-.055819	.026890	.659855	-.751393	-48.711186	-48.524411	.186775
8	-.054806	.025743	.709285	-.704922	-44.823267	-45.104394	-.281127
9	-.053725	.024659	.754194	-.656652	-41.044999	-40.853539	.191460
10	-.052567	.023657	.794745	-.606943	-37.368781	-37.625656	-.256875
11	-.051354	.022722	.831104	-.556116	-33.787638	-33.588616	.199022
12	-.050079	.021875	.863461	-.504415	-30.292548	-30.525059	-.232510
13	-.048760	.021098	.891984	-.452066	-26.876345	-26.677045	.199300
14	-.047392	.020410	.916845	-.399244	-23.530900	-23.747263	-.216363
15	-.045991	.019794	.938188	-.346125	-20.250477	-20.052420	.198057
16	-.044553	.019269	.956160	-.292844	-17.028325	-17.232026	-.203701
17	-.043090	.018815	.970883	-.239554	-13.860241	-13.667107	.193133
18	-.041603	.018454	.982479	-.186374	-10.741234	-10.937439	-.196206
19	-.040099	.018163	.991055	-.133451	-7.669062	-7.479725	.189337
20	-.038581	.017964	.996722	-.080908	-4.640749	-4.829708	-.188959
21	-.037056	.017835	.999582	-.028900	-1.656099	-1.473101	.182998
22	-.035525	.017796	.999748	.022434	1.285494	1.101035	-.184459
23	-.033995	.017825	.997338	.072922	4.181807	4.359459	.177652
24	-.032468	.017941	.992480	.122405	7.030895	6.852544	-.178351
25	-.030948	.018124	.985324	.170694	9.828195	9.998266	.170070
26	-.029440	.018390	.976034	.217620	12.569287	12.396817	-.172470
27	-.027945	.018719	.964798	.262990	15.247577	15.409999	.162423
28	-.026469	.019125	.951826	.306639	17.856798	17.693059	-.163739
29	-.025010	.019591	.937347	.348398	20.389368	20.541534	.152167
30	-.023577	.020128	.921603	.388133	22.838366	22.684437	-.153928
31	-.022164	.020718	.904852	.425726	25.196606	25.337540	.140934
32	-.020780	.021374	.887349	.461099	27.458053	27.316499	-.141554
33	-.019420	.022076	.869345	.494206	29.617435	29.745811	.128376
34	-.018091	.022836	.851078	.525040	31.670942	31.543219	-.127723
35	-.016786	.023637	.832765	.553627	33.616210	33.731800	.115590
36	-.015513	.024487	.814598	.580026	35.452405	35.339683	-.112722
37	-.014264	.025372	.796736	.604327	37.180419	37.283913	.103493
38	-.013046	.026300	.779310	.626639	38.802575	38.705314	-.097261
39	-.011851	.027257	.762410	.647095	40.322917	40.415622	.092705
40	-.010685	.028250	.746095	.665840	41.746770	41.664392	-.082379
41	-.009542	.029268	.730390	.683031	43.080935	43.165009	.084074
42	-.008425	.030315	.715288	.698830	44.333184	44.264405	-.068780
43	-.007329	.031384	.700758	.713399	45.512148	45.589988	.077840

44	-.006257	.032477	.686748	.726896	46.626798	46.569472	-.057326
45	-.005205	.033589	.673192	.739467	47.686057	47.759922	.073866
46	-.004176	.034722	.660022	.751246	48.698461	48.650018	-.048443
47	-.003164	.035872	.647165	.762350	49.671788	49.743455	.071667
48	-.002175	.037040	.634554	.772878	50.613078	50.571002	-.042076
49	-.001202	.038223	.622126	.782917	51.528458	51.598768	.070311
50	-.000251	.039423	.609821	.792539	52.423432	52.385144	-.038288
51	.000683	.040635	.597586	.801805	53.302815	53.372211	.069396
52	.001596	.041864	.585368	.810768	54.171018	54.134276	-.036742
53	.002493	.043105	.573120	.819472	55.031929	55.100231	.068302
54	.003369	.044360	.560797	.827953	55.889056	55.851537	-.037518
55	.004229	.045627	.548361	.836242	56.745343	56.812247	.066904
56	.005067	.046909	.535780	.844358	57.603197	57.563012	-.040185
57	.005888	.048201	.523029	.852315	58.464360	58.529212	.064852
58	.006687	.049507	.510094	.860119	59.329925	59.285501	-.044424
59	.007469	.050823	.496969	.867768	60.200330	60.262436	.062106
60	.008229	.052152	.483658	.875257	61.075390	61.025821	-.049569
61	.008970	.053492	.470175	.882573	61.954366	62.013050	.058684
62	.009689	.054844	.456539	.889703	62.835996	62.781005	-.054990
63	.010389	.056205	.442781	.896630	63.718546	63.773528	.054983
64	.011066	.057578	.428938	.903334	64.599807	64.540029	-.059778
65	.011724	.058961	.415058	.909795	65.477041	65.528066	.051025
66	.012358	.060354	.401197	.915992	66.346955	66.283522	-.063433
67	.012974	.061756	.387428	.921900	67.205467	67.253459	.047992
68	.013566	.063168	.373836	.927495	68.047616	67.983622	-.063995
69	.014140	.064587	.360535	.932746	68.866947	68.912665	.045718
70	.014691	.066016	.347665	.937619	69.655408	69.595000	-.060488
71	.015224	.067451	.335418	.942069	70.402040	70.449821	.047780
72	.015737	.068893	.324040	.946043	71.092580	71.046554	-.046027
73	.016234	.070341	.313894	.949458	71.705965	71.752580	.046615
74	.016713	.071795	.305461	.952205	72.214096	72.190187	-.023909
75	.017182	.073253	.299484	.954101	72.573394	72.619141	.045747
76	.017639	.074714	.296980	.954884	72.723681	72.747300	.023620
77	.018093	.076176	.299613	.954061	72.565635	72.584555	.018920
78	.018551	.077637	.309424	.950924	71.975472	72.068848	.093376
79	.019023	.079094	.328051	.944660	70.849478	70.881410	.031933
80	.019524	.080540	.322159	.946686	71.206447	71.149747	-.056701

NORMALIZATION FACTOR FOR A = .99984186E+00 B = .10000455E+01

AFTER 4 ITERATIONS , THE RELATIVE CHANGE IN SEGMENT LENGTH = .56328E-04

SLAT COORDINATES AND PRESSURE DISTRIBUTION FOR UPPER SURFACE ***

I	X(I)	Y(I)	XBP(I)	YBP(I)	CP(I)
1	.02000	.08200	.01944	.08263	.54448
2	.01887	.08326	.01832	.08389	.37919
3	.01776	.08452	.01706	.08497	.17519
4	.01637	.08543	.01567	.08589	-.02167
5	.01497	.08635	.01423	.08674	-.20524
6	.01349	.08714	.01275	.08753	-.38073
7	.01202	.08792	.01125	.08826	-.54908
8	.01049	.08861	.00973	.08895	-.71200
9	.00897	.08929	.00819	.08959	-.86989
10	.00741	.08989	.00663	.09019	-1.02352
11	.00585	.09049	.00505	.09075	-1.17300
12	.00426	.09101	.00346	.09127	-1.31873
13	.00267	.09153	.00186	.09175	-1.46069
14	.00106	.09197	.00025	.09219	-1.59911
15	-.00056	.09241	-.00137	.09260	-1.73391
16	-.00219	.09278	-.00300	.09296	-1.86520
17	-.00382	.09314	-.00464	.09328	-1.99287
18	-.00547	.09342	-.00629	.09357	-2.11695
19	-.00711	.09371	-.00794	.09381	-2.23731
20	-.00877	.09391	-.00960	.09402	-2.35391
21	-.01043	.09412	-.01126	.09418	-2.46661
22	-.01210	.09424	-.01293	.09431	-2.57532
23	-.01377	.09437	-.01460	.09439	-2.67987
24	-.01544	.09441	-.01627	.09444	-2.78015
25	-.01711	.09446	-.01794	.09444	-2.87595
26	-.01878	.09442	-.01961	.09440	-2.96713
27	-.02045	.09438	-.02128	.09432	-3.05347
28	-.02212	.09426	-.02295	.09420	-3.13478
29	-.02378	.09414	-.02461	.09404	-3.21084
30	-.02544	.09394	-.02627	.09383	-3.28141
31	-.02710	.09373	-.02792	.09358	-3.34628
32	-.02875	.09344	-.02957	.09329	-3.40516
33	-.03039	.09314	-.03121	.09295	-3.45782
34	-.03202	.09276	-.03283	.09257	-3.50395
35	-.03365	.09238	-.03445	.09214	-3.54334
36	-.03525	.09191	-.03605	.09167	-3.57562
37	-.03686	.09144	-.03764	.09116	-3.60058
38	-.03843	.09087	-.03922	.09059	-3.61784
39	-.04000	.09031	-.04077	.08998	-3.62721
40	-.04154	.08966	-.04231	.08933	-3.62827

SLAT COORDINATES AND PRESSURE DISTRIBUTION FOR UPPER SURFACE ***

I	X(I)	Y(I)	XBP(I)	YBP(I)	CP(I)
41	-.04308	.08900	-.04383	.08863	-3.62086
42	-.04457	.08825	-.04532	.08788	-3.60453
43	-.04607	.08750	-.04679	.08708	-3.57919
44	-.04751	.08666	-.04823	.08624	-3.54438
45	-.04896	.08582	-.04965	.08535	-3.50006
46	-.05034	.08488	-.05103	.08441	-3.44574
47	-.05173	.08395	-.05238	.08343	-3.38152
48	-.05304	.08292	-.05370	.08240	-3.30692
49	-.05436	.08189	-.05498	.08132	-3.22211
50	-.05560	.08076	-.05622	.08020	-3.12667
51	-.05684	.07964	-.05741	.07903	-3.02094
52	-.05799	.07843	-.05856	.07782	-2.90453
53	-.05914	.07721	-.05966	.07656	-2.77797
54	-.06019	.07591	-.06071	.07526	-2.64093
55	-.06124	.07461	-.06171	.07392	-2.49422
56	-.06218	.07323	-.06265	.07254	-2.33762
57	-.06312	.07185	-.06352	.07112	-2.17218
58	-.06393	.07039	-.06434	.06966	-1.99785
59	-.06474	.06893	-.06508	.06817	-1.81603
60	-.06542	.06740	-.06576	.06664	-1.62689
61	-.06610	.06587	-.06636	.06508	-1.43220
62	-.06663	.06429	-.06689	.06349	-1.23238
63	-.06715	.06270	-.06733	.06188	-1.02967
64	-.06751	.06107	-.06769	.06025	-.82482
65	-.06787	.05944	-.06796	.05860	-.62054
66	-.06806	.05777	-.06815	.05694	-.41795
67	-.06824	.05611	-.06823	.05528	-.22028
68	-.06822	.05444	-.06822	.05360	-.02902
69	-.06821	.05277	-.06810	.05194	.15216
70	-.06799	.05111	-.06788	.05028	.32153
71	-.06776	.04946	-.06754	.04865	.47525
72	-.06731	.04785	-.06708	.04704	.61179
73	-.06686	.04624	-.06650	.04548	.72773
74	-.06614	.04473	-.06579	.04397	.82263
75	-.06543	.04321	-.06491	.04256	.89451
76	-.06439	.04190	-.06387	.04125	.94544
77	-.06336	.04059	-.06278	.03999	.97704
78	-.06220	.03938	-.06162	.03878	.99242
79	-.06105	.03818	-.06078	.03738	.99730
80	-.06052	.03659	-.06026	.03579	.99952

SLAT COORDINATES AND PRESSURE DISTRIBUTION FOR LOWER SURFACE ***

I	X(I)	Y(I)	XBP(I)	YBP(I)	CP(I)
81	-.06000	.03500	-.05977	.03427	.99939
82	-.05953	.03354	-.05926	.03283	.99271
83	-.05898	.03211	-.05865	.03142	.97542
84	-.05832	.03073	-.05795	.03006	.94483
85	-.05759	.02939	-.05716	.02875	.89985
86	-.05674	.02811	-.05628	.02750	.84051
87	-.05582	.02689	-.05531	.02632	.76776
88	-.05481	.02574	-.05427	.02520	.68292
89	-.05372	.02466	-.05315	.02416	.58764
90	-.05257	.02366	-.05196	.02319	.48353
91	-.05135	.02272	-.05072	.02230	.37224
92	-.05008	.02188	-.04942	.02149	.25517
93	-.04876	.02110	-.04808	.02075	.13365
94	-.04739	.02041	-.04669	.02010	.00868
95	-.04599	.01979	-.04527	.01953	-.11885
96	-.04455	.01927	-.04382	.01904	-.24838
97	-.04309	.01882	-.04235	.01863	-.37946
98	-.04160	.01845	-.04085	.01831	-.51198
99	-.04010	.01816	-.03934	.01806	-.64590
100	-.03858	.01796	-.03782	.01790	-.78149
101	-.03706	.01783	-.03629	.01782	-.91905
102	-.03553	.01780	-.03476	.01781	-1.05918
103	-.03399	.01783	-.03323	.01788	-1.20245
104	-.03247	.01794	-.03171	.01803	-1.34965
105	-.03095	.01812	-.03019	.01826	-1.50147
106	-.02944	.01839	-.02869	.01855	-1.65872
107	-.02794	.01872	-.02721	.01892	-1.82202
108	-.02647	.01913	-.02574	.01936	-1.99198
109	-.02501	.01959	-.02429	.01986	-2.16884
110	-.02358	.02013	-.02287	.02042	-2.35269
111	-.02216	.02072	-.02147	.02105	-2.54309
112	-.02078	.02137	-.02010	.02172	-2.73934
113	-.01942	.02208	-.01876	.02246	-2.94007
114	-.01809	.02284	-.01744	.02324	-3.14357
115	-.01679	.02364	-.01615	.02406	-3.34742
116	-.01551	.02449	-.01489	.02493	-3.54892
117	-.01426	.02537	-.01365	.02584	-3.74475
118	-.01305	.02630	-.01245	.02678	-3.93143
119	-.01185	.02726	-.01127	.02775	-4.10513
120	-.01069	.02825	-.01011	.02876	-4.26210

SLAT COORDINATES AND PRESSURE DISTRIBUTION FOR LOWER SURFACE ***

I	X(I)	Y(I)	XBP(I)	YBP(I)	CP(I)
121	-.00954	.02927	-.00898	.02979	-4.39863
122	-.00843	.03031	-.00788	.03085	-4.51149
123	-.00733	.03138	-.00679	.03193	-4.59790
124	-.00626	.03248	-.00573	.03303	-4.65584
125	-.00520	.03359	-.00469	.03416	-4.68405
126	-.00418	.03472	-.00367	.03530	-4.68213
127	-.00316	.03587	-.00267	.03646	-4.65039
128	-.00217	.03704	-.00169	.03763	-4.58984
129	-.00120	.03822	-.00073	.03882	-4.50203
130	-.00025	.03942	.00022	.04003	-4.38891
131	.00068	.04064	.00114	.04125	-4.25279
132	.00160	.04186	.00204	.04248	-4.09618
133	.00249	.04310	.00293	.04373	-3.92175
134	.00337	.04436	.00380	.04499	-3.73222
135	.00423	.04563	.00465	.04627	-3.53035
136	.00507	.04691	.00548	.04755	-3.31877
137	.00589	.04820	.00629	.04885	-3.10002
138	.00669	.04951	.00708	.05016	-2.87642
139	.00747	.05082	.00785	.05149	-2.65007
140	.00823	.05215	.00860	.05282	-2.42282
141	.00897	.05349	.00933	.05417	-2.19627
142	.00969	.05484	.01004	.05552	-1.97177
143	.01039	.05620	.01073	.05689	-1.75048
144	.01107	.05758	.01139	.05827	-1.53332
145	.01172	.05896	.01204	.05966	-1.32110
146	.01236	.06035	.01267	.06105	-1.11445
147	.01297	.06176	.01327	.06246	-.91391
148	.01357	.06317	.01385	.06388	-.71992
149	.01414	.06459	.01442	.06530	-.53288
150	.01469	.06602	.01496	.06673	-.35314
151	.01522	.06745	.01548	.06817	-.18102
152	.01574	.06889	.01599	.06962	-.01685
153	.01623	.07034	.01647	.07107	.13901
154	.01671	.07180	.01695	.07252	.28619
155	.01718	.07325	.01741	.07398	.42425
156	.01764	.07471	.01787	.07545	.55271
157	.01809	.07618	.01832	.07691	.67105
158	.01855	.07764	.01879	.07837	.77893
159	.01902	.07909	.01927	.07982	.87674
160	.01952	.08054	.01976	.08127	.96717
161	.02000	.08200			

```
            PRESSURE DUMP
 1  -.137073E+01   .378068E+00  -.378068E-01   .995037E+00  -.995037E-01   .494093E-01
            PRESSURE DUMP
 2  -.100939E+01   .503398E+00  -.755097E-01   .988936E+00  -.148340E+00  -.842944E-01
            PRESSURE DUMP
 3  -.111967E+01   .560164E+00  -.700205E-01   .992278E+00  -.124035E+00  -.319359E+00
            PRESSURE DUMP
 4  -.121677E+01   .612819E+00  -.459614E-01   .997199E+00  -.747899E-01  -.569855E+00
            PRESSURE DUMP
 5  -.133683E+01   .649873E+00   .259949E-01   .999201E+00   .399680E-01  -.777596E+00
            PRESSURE DUMP
 6  -.139466E+01   .663043E+00   .112717E+00   .985856E+00   .167595E+00  -.864229E+00
            PRESSURE DUMP
 7  -.137658E+01   .613593E+00   .208622E+00   .946773E+00   .321903E+00  -.754632E+00
            PRESSURE DUMP
 8  -.132956E+01   .691721E+00   .391975E+00   .870022E+00   .493013E+00  -.181353E+01
            PRESSURE DUMP
 9  -.219963E+01   .818466E+00   .851205E+00   .693109E+00   .720833E+00  -.450918E+01
            PRESSURE DUMP
10  -.246557E+01   .356197E+00   .854874E+00   .384615E+00   .923077E+00  -.256873E+01
            PRESSURE DUMP
11  -.138640E+01  -.968229E-01   .232375E+00  -.384615E+00   .923077E+00   .630283E+00
            PRESSURE DUMP
12  -.388159E-01   .702192E-01  -.730280E-01  -.693109E+00   .720833E+00   .959863E+00
            PRESSURE DUMP
13   .434941E+00   .256384E+00  -.145284E+00  -.870022E+00   .493013E+00   .649758E+00
            PRESSURE DUMP
14   .753560E+00   .388373E+00  -.132047E+00  -.946773E+00   .321903E+00   .309139E+00
            PRESSURE DUMP
15   .930334E+00   .466580E+00  -.793187E-01  -.985856E+00   .167595E+00   .717472E-01
            PRESSURE DUMP
16   .103040E+01   .503737E+00  -.201495E-01  -.999201E+00   .399680E-01  -.699667E-01
            PRESSURE DUMP
17   .109061E+01   .512799E+00   .384599E-01  -.997199E+00  -.747899E-01  -.102317E+00
            PRESSURE DUMP
18   .105209E+01   .493201E+00   .616501E-01  -.992278E+00  -.124035E+00  -.260463E-01
            PRESSURE DUMP
19   .101151E+01   .461838E+00   .692757E-01  -.988936E+00  -.148340E+00   .842171E-01
            PRESSURE DUMP
20   .948329E+00   .361991E+00   .361991E-01  -.995037E+00  -.995037E-01   .109693E+00
```

PRESSURE DISTRIBUTION ON WING

NO.	XBP	YBP	X/CREF	Y/CREF	XTR/CR	YTR/CR	CP-INNER	CP
1	1.1000	.0100	.9167	.0083			.9537	.0494
2	.9000	.0350	.7500	.0292			.9995	-.0843
3	.7000	.0625	.5833	.0521			.9996	-.3194
4	.5000	.0825	.4167	.0688			.9994	-.5699
5	.3000	.0860	.2500	.0717			.9989	-.7776
6	.1500	.0735	.1250	.0612			.9996	-.8642
7	.0750	.0565	.0625	.0471			.9992	-.7546
8	.0350	.0395	.0292	.0329			.9924	-1.8135
9	.0137	.0245	.0115	.0204			.9998	-4.5092
10	.0038	.0090	.0031	.0075			.9986	-2.5687
11	.0038	-.0090	.0031	-.0075			.9891	.6303
12	.0137	-.0245	.0115	-.0204			1.0000	.9599
13	.0350	-.0395	.0292	-.0329			1.0000	.6498
14	.0750	-.0565	.0625	-.0471			.9999	.3091
15	.1500	-.0735	.1250	-.0612			.9997	.0717
16	.3000	-.0860	.2500	-.0717			.9993	-.0700
17	.5000	-.0825	.4167	-.0688			.9995	-.1023
18	.7000	-.0625	.5833	-.0521			.9996	-.0260
19	.9000	-.0350	.7500	-.0292			.9995	.0842
20	1.1000	-.0100	.9167	-.0083			.9534	.1097

*** FORCE COEFFICIENTS FOR SLAT ***

NO.	DC-N	DC-C	DC-M
1	-.0004	-.0007	-.0005
2	-.0004	-.0004	-.0004
3	-.0002	-.0001	-.0002
4	.0000	.0000	.0000
5	.0002	.0001	.0003
6	.0005	.0002	.0005
7	.0007	.0003	.0007
8	.0009	.0004	.0009
9	.0011	.0004	.0012
10	.0013	.0005	.0014
11	.0015	.0005	.0016
12	.0017	.0006	.0018
13	.0020	.0006	.0020
14	.0022	.0006	.0022
15	.0024	.0005	.0024
16	.0025	.0006	.0026
17	.0027	.0005	.0028
18	.0029	.0005	.0030
19	.0031	.0004	.0031
20	.0033	.0004	.0033
21	.0034	.0003	.0035
22	.0036	.0003	.0036
23	.0037	.0001	.0038
24	.0039	.0001	.0039
25	.0040	-.0001	.0041
26	.0041	-.0001	.0042
27	.0042	-.0003	.0043
28	.0044	-.0003	.0044
29	.0044	-.0005	.0045
30	.0045	-.0006	.0046
31	.0046	-.0008	.0046
32	.0047	-.0009	.0047
33	.0047	-.0011	.0047
34	.0047	-.0011	.0048
35	.0047	-.0014	.0048
36	.0048	-.0014	.0048
37	.0047	-.0016	.0048
38	.0047	-.0017	.0048
39	.0047	-.0019	.0047
40	.0046	-.0020	.0046
41	.0045	-.0022	.0045
42	.0045	-.0023	.0045
43	.0043	-.0025	.0043
44	.0042	-.0025	.0042
45	.0041	-.0027	.0040
46	.0040	-.0027	.0039
47	.0037	-.0029	.0037
48	.0036	-.0029	.0036
49	.0034	-.0030	.0033
50	.0032	-.0029	.0032

104	-.0017	.0002	-.0018
105	-.0019	.0003	-.0019
106	-.0021	.0005	-.0021
107	-.0022	.0006	-.0023
108	-.0024	.0008	-.0025
109	-.0026	.0010	-.0026
110	-.0028	.0012	-.0028
111	-.0029	.0014	-.0030
112	-.0031	.0016	-.0031
113	-.0033	.0019	-.0033
114	-.0034	.0021	-.0034
115	-.0036	.0024	-.0036
116	-.0037	.0026	-.0037
117	-.0038	.0029	-.0038
118	-.0039	.0031	-.0039
119	-.0040	.0034	-.0039
120	-.0041	.0036	-.0040
121	-.0041	.0038	-.0040
122	-.0041	.0040	-.0040
123	-.0041	.0042	-.0040
124	-.0041	.0043	-.0040
125	-.0040	.0044	-.0039
126	-.0039	.0045	-.0038
127	-.0038	.0045	-.0037
128	-.0037	.0045	-.0036
129	-.0036	.0045	-.0034
130	-.0034	.0044	-.0033
131	-.0032	.0044	-.0031
132	-.0031	.0042	-.0029
133	-.0029	.0041	-.0027
134	-.0027	.0039	-.0025
135	-.0025	.0038	-.0023
136	-.0023	.0036	-.0021
137	-.0021	.0034	-.0019
138	-.0019	.0032	-.0017
139	-.0017	.0029	-.0015
140	-.0015	.0027	-.0014
141	-.0013	.0025	-.0012
142	-.0011	.0022	-.0010
143	-.0010	.0020	-.0009
144	-.0008	.0018	-.0007
145	-.0007	.0015	-.0006
146	-.0006	.0013	-.0005
147	-.0005	.0011	-.0004
148	-.0003	.0009	-.0003
149	-.0002	.0006	-.0002
150	-.0002	.0004	-.0001
151	-.0001	.0002	-.0001
152	-.0000	.0000	-.0000
153	.0001	-.0002	.0000
154	.0001	-.0003	.0001
155	.0002	-.0005	.0001
156	.0002	-.0007	.0002
157	.0003	-.0008	.0002
158	.0003	-.0009	.0002
159	.0004	-.0011	.0003
160	.0004	-.0011	.0003

```
NORMAL FORCE COEFFICIENT            =    .061404
LIFT COEFFICIENT                    =    .055346
CHORDWISE FORCE COEFFICIENT         =    .066826
DRAG COEFFICIENT                    =    .071924
PITCHING MOMENT COEFFICIENT         =    .065312
POINT ABOUT WHICH MOMENTS ARE TAKEN (  1.2000,  0.0000)
```

*** FORCE COEFFICIENTS FOR MAIN AIRFOIL ***

NO.	DC-N	DC-C	DC-M
1	-.0067	-.0007	-.0004
2	.0162	.0024	.0044
3	.0535	.0067	.0231
4	.0941	.0071	.0558
5	.1265	-.0051	.0947
6	.0700	-.0119	.0606
7	.0366	-.0124	.0337
8	.0471	-.0267	.0449
9	.0397	-.0413	.0384
10	.0150	-.0360	.0147
11	.0017	.0040	.0016
12	.0090	.0094	.0087
13	.0156	.0089	.0149
14	.0127	.0043	.0117
15	.0057	.0010	.0050
16	-.0103	-.0004	-.0076
17	-.0148	.0011	-.0087
18	-.0036	.0005	-.0017
19	.0123	-.0018	.0030
20	.0165	-.0016	.0014

NORMAL FORCE COEFFICIENT = .536704

LIFT COEFFICIENT = .542744

CHORDWISE FORCE COEFFICIENT = -.092731

DRAG COEFFICIENT = -.045601

PITCHING MOMENT COEFFICIENT = .398264

POINT ABOUT WHICH MOMENTS ARE TAKEN (1.2000, 0.0000)

```
*** TOTAL FORCE COEFFICIENTS ***

        NORBAL FORCE COEFFICIENT            =   .598108

        LIFT COEFFICIENT                    =   .598090

        CHORDWISE FORCE COEFFICIENT         =  -.025905

        DRAG COEFFICIENT                    =   .026322

        PITCHING MOMENT COEFFICIENT         =   .463576

        POINT ABOUT WHICH MOMENTS ARE TAKEN (  1.2000,  0.0000)
```

APPENDIX IV

PROGRAM LISTING

```
      SUBROUTINE  CLCR (C1, C2, C3, C4, C5, C6, C7, C8)                 XOVS0010
C   **                                                                  XOVS0020
C   **             SUBROUTINE CLCR CALCULATES THE INCOMPRESSIBLE        XOVS003
C   **        PRESSURE COEFFIENTS AT ANY BOUNDARY POINT AFTER THE       XOVS004
C   **        VELOCITY COMPONENTS HAVE BEEN CALCULATED                  XOVS0041
C   **                                                                  XOVS0050
      GVG = (C1 + C2)/4.                                                XOVS0060
      G1 = GVG * C5                                                     XOVS0070
      G2 = GVG * C6                                                     XOVS0080
      C7 = 1. - (C3 - G1) ** 2 - (C4 - G2) ** 2                         XOVS0090
      C8 = 1. - (C3 + G1) ** 2 - (C4 + G2) ** 2                         XOVS0100
      RETURN                                                            XOVS0110
      END                                                               XOVS0120
```

```
      SUBROUTINE  CONTRB  (I, K, TX, TY, Q)                             XOVH0010
C  **                                                                   XOVH0011
C  **              SUBROUTINE CONTRB CALCULATES THE PORTION OF THE      XOVH0012
C  **         STREAM FUNCTION INDUCED BY A SINGLE DISTRIBUTED           XOVH0013
C  **         SINGULARITY                                               XOVH0014
C  **                                                                   XOVH0015
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)  XOVH0020
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMCD,NPTS,NB,NCTS,NSOR1,NVORT  XOVH0030
      DATA  EPS, FOUR, CV, CS /1.0 E-10, 4.0, .0397887358,-.0795774716 /XOVH0040
     1      , PIHALF /1.5707963/                                        XOVH0050
C                                                                       XOVH0060
      DELX = X(I) - X(K)                                                XOVH0070
      DELY = Y(I) - Y(K)                                                XOVH0080
      DELX2 = DELX * DELX                                               XOVH0090
      DELY2 = DELY * DELY                                               XOVH0100
      DIST2 = DELX2 + DELY2                                             XOVH0110
      IF  (DIST2 .GE. EPS) GO  TO 10                                    XOVH0120
      WRITE  (6,80)                                                     XOVH0130
      CALL  EXIT                                                        XOVH0140
10    CONTINUE                                                          XOVH0150
      DIST = SQRT (DIST2)                                               XOVH0160
      SINA = DELY / DIST                                                XOVH0170
      COSA = DELX / DIST                                                XOVH0180
C                   ROTATION OF AXIS TO GET  XI AND XK ON X AXIS        XOVH0190
      TMXK = TX - X(K)                                                  XOVH0200
      TMYK = TY - Y(K)                                                  XOVH0210
      TXP = TMXK * COSA + TMYK * SINA                                   XOVH0220
      TYP = - TMXK * SINA + TMYK * COSA                                 XOVH0230
      XIP = DELX * COSA + DELY * SINA                                   XOVH0240
C     YIP = - DELX * SINA + DELY * COSA                                 XOVH0250
C                                                                       XOVH0260
      TXP2 = TXP * TXP                                                  XOVH0270
      TYP2 = TYP * TYP                                                  XOVH0280
      XIP2 = XIP * XIP                                                  XOVH0290
      TMXIP = TXP - XIP                                                 XOVH0300
      TMXIP2 = TMXIP * TMXIP                                            XOVH0310
      XLK = ALOG ((TMXIP2 + TYP2) / (TXP2 + TYP2))                      XOVH0320
      IF  (ABS(TYP) .GT. EPS ) GO  TO 20                                XOVH0330
      THET = PIHALF                                                     XOVH0340
      IF  (TMXIP .LT. 0.0) THET = - PIHALF                              XOVH0350
      GO  TO 30                                                         XOVH0360
20    CONTINUE                                                          XOVH0370
      A = TMXIP / TYP                                                   XOVH0380
      THET = ATAN(A)                                                    XOVH0390
30    CONTINUE                                                          XOVH0400
      IF  (ABS(TYP).GT. EPS) GO  TO 40                                  XOVH0410
      THETB = PIHALF                                                    XOVH0420
      IF  (TXP .LT. 0.0) THETB = - PIHALF                               XOVH0430
      GO  TO 50                                                         XOVH0440
40    CONTINUE                                                          XOVH0450
      A = TXP / TYP                                                     XOVH0460
      THETB = ATAN(A)                                                   XOVH0470
50    CONTINUE                                                          XOVH0480
      THETA = THET - THETB                                              XOVH0490
C                                                                       XOVH0500
      IF  (I .GT. NVORT) GO  TO 60                                      XOVH0510
```

```
C
      Q = (CV / XIP) * ((TXP2 - TYP2) * XLK + FOUR * TYP * TXP * THETA
     1      - XIP2 * ALOG(TMXIP2 + TYP2) + (TXP + TXP + XIP) * XIP )
      GO  TO 70
C
60    CONTINUE
C
      Q = (CS / XIP) * ((TYP2 - TXP2)* THETA + TYP * XIP + TXP * TYP *
     1      XLK + XIP2 * THET)
C
70    CONTINUE
C
      RETURN
C
80    FORMAT  (1H0, 53H*** DISTANCE BETWEEN TWO POINTS IS ZERO IN CONTRB
     1 *** )
      END
```

```
      SUBROUTINE  CRWTB (C1,C2,C3,M)
C   **
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1             DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)
      COMMON /BLKK/ SCOSLP(160), CPSLAT(160), SSINLP(160), XSLAT(161)
     1      , XXBP(160), YSLAT(161), YYBP(160)
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE
      DIMENSION C1(M), C2(M), C3(M), HD(12)
      DIMENSION YPP(200),XPP(200)
      DATA HD / 10H         NO, 10H.          , 10HXBP
     1     ,     10H     YBP    , 10H      X/CR, 10HEF
     2     ,     10HY/CREF      , 10H     XTR/CR, 10H          YT
     3     ,     10HR/CR        , 10H  CP-INNER, 10H          CP /
C   **
C  **                  WRITES PRESSURE COEFFIENTS...
C   **
      WRITE  (6,20)    HD
      DO 10 IZ = 1, NB
      XPP(IZ) = (XBP(IZ) - XLE) / CREF
      YPP(IZ) = (YBP(IZ) - YLE) / CREF
      WRITE  (6,30)    IZ, XBP(IZ), YBP(IZ), XPP(IZ), YPP(IZ), C2(IZ),
     1                 C1(IZ)
10    CONTINUE
      RETURN
C
20    FORMAT  (1H0, 12A10)
30    FORMAT  (8X, I3, 4(6X, F8.4), 29X, 2(6X, F8.4))
      END
```

```
      SUBROUTINE  DSTRBD (X1, Y1, X2, Y2, U, V, A, X3, Y3)
C  **
C  **              SUBROUTINE 'DSTRBD' COMPUTES THE 'DISTRIBUTED'
C  **         COMPONENTS OF VELOCITY...
C  **
      CALL  DSTSQD (X2, Y2, X1, Y1, X4, Y4, B, B1, 1)
      CALL  DSTSQD (X2, Y2, X3, Y3, X5, Y5, C, B1, 0)
      BSQ = A - B - C
      DSQ = 2. * B  +  BSQ
      CSQ = 2. * (X4 * Y5  -  X5 * Y4)
      IF  (ABS(CSQ) .LT. 1.0E-13) GO  TO 10
      D = (ATAN(DSQ/CSQ) - ATAN(BSQ/CSQ))/CSQ
      GO  TO 20
10    CONTINUE
      SB = 1.0 / BSQ
      SD = 1.0 / DSQ
      SB2 = SB * SB
      SB3 = SB * SB2
      SB5 = SB3 * SB2
      SD2 = SD * SD
      SD3 = SD * SD2
      SD5 = SD3 * SD2
      CSQ2 = CSQ * CSQ
      CSQ4 = CSQ2 * CSQ2
      D = SB - SD + (CSQ2 * (SD3 - SB3) / 3.0) - (CSQ4*(SD5-SB5)/5.0)
20    CONTINUE
      E = ALOG(A/C)
      BIK = (0.5 * E  -  BSQ * D)/B
      FIK = (B - 0.5 * BSQ * E + (BSQ * BSQ - 2. * B * C) * D)/(B * B)
      SQBP = B1/6.283184
      U = SQBP * (Y5 * BIK - Y4 * FIK)
      V = SQBP * (X4 * FIK - X5 * BIK)
      RETURN
      END
```

```
      SUBROUTINE  DSTSQD (X1, Y1, X2, Y2, X3, Y3, S1, S2, K)
C   **
C   **                  SUBROUTINE 'DSTSQD' CALCULATES THE FOLLOWING, USING
C   **            ANY TWO POINTS IN TWO-SPACE--
C   **                        -- THE DIFFERENCE BETWEEN THE ABSCISSAS...
C   **                        -- THE DIFFERENCE BETWEEN THE ORDINATES...
C   **                        -- THE SQUARE OF THE DISTANCE BETWEEN THE
C   **                           POINTS...
C   **                        -- THE DISTANCE BETWEEN THE POINTS, IF K.NE.0...
C   **
      X3 = X1 - X2
      Y3 = Y1 - Y2
C   **
      S1 = X3 * X3  +  Y3 * Y3
C   **
      IF  (K.NE.0) S2 = SQRT(S1)
C   **
      RETURN
C   **
      END
```

```
      SUBROUTINE  FLOSAT  (USUM, TX, TY)
C   **
C   **               SUBROUTINF FLOSAT DIRECTS THE COMPUTATION OF THE
C   **         STREAM FUNCTION AT ANY POINT BY SUBROUTINE CONTRB
C   **
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)
      COMMON /BLKE/ U(200),V(200),CPUP(200),CPLR(200),CPM(200)
      COMMON /BLKG/ GAMMA(200)
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE
      DIMENSION T(400)
      EQUIVALENCE (T(1), U(1))
C   **
      TX = XSLE
      TY = YSLE
      K = 0
10    CONTINUE
      CALL  CONTRB  (1, 2, TX, TY, UQ)
      K = K + 1
      T(K) = UQ
C   **
      DO 20 IY = 2, NB
      K = K + 1
      CALL  CONTRB  (IY, IY+1, TX, TY, UQ)
      T(K) = UQ
      CALL  CONTRB  (IY, IY-1, TX, TY, UQ)
      T(K) = T(K) + UQ
20    CONTINUE
      CALL  CONTRB  (NPTS, NPTS-1, TX, TY, UQ)
      K = K + 1
      T(K) = UQ
C   **
C                      VORTEX CONTROL POINTS ON SLAT
C   **
      IF  (NVOR .EQ. 0) GO  TO 5C
      IY = NCTS + 1
      CALL  CONTRB  (NCTS, IY, TX, TY, UQ)
      K = K + 1
      T(K) = UQ
C   **
      IF  (IY .EQ. NSOR1) GO  TO 50
      NVORM1 = NVORT - 1
      IF  (NVOR .EQ. 2) GO  TO 4C
      DO 30 I = IY, NVORM1
      K = K + 1
      CALL  CONTRB  (I, I+1, TX, TY, UQ)
      T(K) = UQ
      CALL  CONTRB  (I, I-1, TX, TY, UQ)
      T(K) = T(K) + UQ
30    CONTINUE
40    CONTINUE
      CALL  CONTRB  (NVORT, NVORM1, TX, TY, UQ)
```

```
      K = K + 1
      T(K) = UQ
50    CONTINUE
C  **
C                        SOURCE CONTROL POINTS ON SLAT
C  **
      IF  (NSOR .EQ. 0) GO  TO 80
      IY = NSOR1 + 1
      CALL  CONTRB  (NSOR1, IY, TX, TY, UQ)
      K = K + 1
      T(K) = UQ
C  **
      IF  (NSOR1 .EQ. NCPTS) GO  TO 80
      IF  (NSOR .EQ. 2) GO  TO 70
C  **
      DO 60 I = IY, IB
      K = K + 1
      CALL  CONTRB  (I, I+1, TX, TY, UQ)
      T(K) = UQ
      CALL  CONTRB  (I, I-1, TX, TY, UQ)
      T(K) = T(K) + UQ
60    CONTINUE
70    CONTINUE
      CALL  CONTRB  (NCPTS, NCPTS-1, TX, TY, UQ)
      K = K + 1
      T(K) = UQ
80    CONTINUE
C  **
      IF  (K .GT. 200) GO  TO 90
C  **
      TX = XSTE
      TY = YSTE
      K = 200
      GO  TO 10
90    CONTINUE
      RETURN
      ENTRY FLOCAL
      USUM = USUBT * (TY * CS - TX * SN)
C  **
      CALL  CONTRB ( 1, 2, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(1)
C  **
      DO 100 IY = 2, NB
      CALL  CONTRB ( IY, IY+1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(IY)
      CALL  CONTRB ( IY, IY-1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(IY)
100   CONTINUE
C  **
      CALL  CONTRB  (NPTS, NPTS-1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(NPTS)
C  **
C                        VORTEX CONTROL POINTS ON SLAT
C  **
      IF  (NVOR .EQ. 0) GO  TO 130
      IY = NCTS + 1
```

```
      CALL  CONTRB  (NCTS, IY, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(NCTS)
C  **
      IF  (IY .EQ. NSOR1) GO  TO 130
      NVORM1 = NVORT - 1
      IF  (NVOR .EQ. 2) GO  TO 120
      DO 110 I = IY, NVORM1
      CALL  CONTRB  (I, I + 1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(I)
      CALL  CONTRB  (I, I - 1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(I)
110   CONTINUE
120   CONTINUE
      CALL  CONTRB  (NVORT, NVORM1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(NVORT)
130   CONTINUE
C  **
C                       SOURCE CONTROL POINTS ON SLAT
C  **
      IF  (NSOR .EQ. 0) GO  TO 160
      IY = NSOR1 + 1
      CALL  CONTRB  (NSOR1, IY, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(NSOR1)
C  **
      IF  (NSOR1 .EQ. NCPTS) GO  TO 160
      IF  (NSOR .EQ. 2) GO  TO 150
C  **
      DO 140 I = IY, IB
      CALL  CONTRB (I, I + 1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(I)
      CALL  CONTRB (I, I - 1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(I)
140   CONTINUE
150   CONTINUE
      CALL  CONTRB  (NCPTS, NCPTS - 1, TX, TY, UQ)
      USUM = USUM + UQ * GAMMA(NCPTS)
160   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  FRCE
C   **
C   **
C                   SUBROUTINE FRCE WRITES THE FORCE COEFFICIENTS
C   **
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)
      COMMON /BLKE/ U(200),V(200),CPUP(200),CPLR(200),CPM(200)
      COMMON /BLKI/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      COMMON /BLKK/ SCOSLP(160), CPSLAT(160), SSINLP(160), XSLAT(161)
     1      , XXBP(160), YSLAT(161), YYBP(160)
C   **
      CN2T = 0.0
      CL1T = 0.0
      CC2T = 0.0
      CD1T = 0.0
      CM2T = 0.0
      IF  (IND (4) .EQ. 1) GO  TO 10
      WRITE  (6,20)
      CALL  SBFRCE  (CPSLAT, XSLAT, YSLAT, XXBP, YYBP, SCCSLP, SSINLP, 1
     160      , 161, CN2, CL1, CC2, CD1, CM2, CS, SN)
      CN2T = CN2T + CN2
      CL1T = CL1T + CL1
      CC2T = CC2T + CC2
      CD1T = CD1T + CD1
      CM2T = CM2T + CM2
      WRITE  (6,30)   CN2, CL1, CC2, CD1, CM2, XMBAR, YMBAR
C   **
10    CONTINUE
      WRITE  (6,40)
      CALL  SBFRCE  (CPM, X, Y, XBP, YBP, COSSLP, SINSLP, NB
     1      , NPTS, CN2, CL1, CC2, CD1, CM2, CS, SN)
      CN2T = CN2T + CN2
      CL1T = CL1T + CL1
      CC2T = CC2T + CC2
      CD1T = CD1T + CD1
      CM2T = CM2T + CM2
      WRITE  (6,30)  CN2, CL1, CC2, CD1, CM2, XMBAR, YMBAR
      WRITE  (6,50)
      WRITE  (6,30)  CN2T, CL1T, CC2T, CD1T, CM2T, XMBAR, YMBAR
      RETURN
C
20    FORMAT  (1H1, 35H*** FORCE COEFFICIENTS FOR SLAT *** )
30    FORMAT  (1H0, 12X, 37HNORMAL FORCE COEFFICIENT            =, F10.6
     1/        1H0, 12X, 37HLIFT COEFFICIENT                    =, F10.6
     2/        1H0, 12X, 37HCHORDWISE FORCE COEFFICIENT         =, F10.6
     3/        1H0, 12X, 37HDRAG COEFFICIENT                    =, F10.6
     4/        1H0, 12X, 37HPITCHING MOMENT COEFFICIENT         =, F10.6
     5/        1H0, 12X, 37HPOINT ABOUT WHICH MOMENTS ARE TAKEN (, F8.4,
     6         1H, F8.4, 1H) // )
40    FORMAT  (1H1, 43H*** FORCE COEFFICIENTS FOR MAIN AIRFOIL *** )
50    FORMAT  (1H1, 32H*** TOTAL FORCE COEFFICIENTS *** )
      END
```

```
      SUBROUTINE  GEOM
C  **
      COMMON /BLKA/ IND(30), NCPTS, NSTLNS, NVLPTS, NPNTS, IB
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      DIMENSION HD(9)
      DATA HD / 10H          , 10H       IND, 10HEX
     1     ,    10H X        , 10H        Y    , 10H
     2     ,    10H XBP      , 10H      YBP    , 10H    SLOPE    /
C  **
C  **               SUBROUTINE 'GEOM' CALCULATES AND WRITES THE BOUNDARY
C  **          POINT COORDINATES, AND THE SLOPE ASSOCIATED WITH EACH
C  **          BOUNDARY POINT.  THE COORDINATES OF THE POINT ASSOCIATED
C  **          WITH EACH BOUNDARY POINT ARE ALSO WRITTEN...
C  **
      APHA=ALPHA/57.29578
      SN=SIN(APHA)
      CS=COS(APHA)
      WRITE  (6,20)
      WRITE  (6,30)  HD
      DO 10 IZ=1, NB
      IY = IZ + 1
      XBP(IZ)= (X(IZ) + X(IY))/2.
      YBP(IZ)= (Y(IZ) + Y(IY))/2.
      CALL  DSTSQD (X(IZ), Y(IZ), X(IY), Y(IY), S, T, D1, D2, 1)
      CALL  SLPCLR (SLPE, S, T)
      COSSLP(IZ) = S/D2
      SINSLP(IZ) = T/D2
      WRITE  (6,40)  IZ, X(IZ), Y(IZ), XBP(IZ), YBP(IZ), SLPE
10    CONTINUE
      WRITE (6,50)  IZ,X(IZ),Y(IZ)
      RETURN
C
20    FORMAT  (1H1, 20X, 23HAIRFOIL DEFINING POINTS )
30    FORMAT  (1H0, 9A10)
40    FORMAT  (19X,I3, 1H.,4X,4(F10.5,5X),4X,F9.5)
50    FORMAT  (19X, I3, 1H., 4X, 2(F10.5, 5X))
      END
```

```
      SUBROUTINE  KEOF (TAPENO, KIND, KSUBR, KSTMT)
C   **
C   **               SUBROUTINE KEOF WRITES THE LOCATION OF AN UNEXPECTED
C   **         END OF FILE DURING A READ OPERATION
C   **
      IF   (KIND .NE. 0) GO  TO 10
      WRITE  (6,30)
      GO  TO 20
10    CONTINUE
      WRITE  (6,40)  TAPENO, KSUBR, KSTMT
20    CONTINUE
      CALL  EXIT
      RETURN
C
30    FORMAT  (22H THIS JOB IS COMPLETE. )
40    FORMAT  (32H UNEXPECTED END OF FILE ON UNIT , I3 /
     1         15H IN SUBROUTINE , A6, 11H STATEMENT , I6 )
      END
```

```
      SUBROUTINE  MTRXGN
C   **
C   **                SUBROUTINE MTRXGN GENERATES THE MATRIX TO BE SOLVED
C   **
C   **
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)
      COMMON /BLKE/ U(200),V(200),CPUP(200),CPLR(200),CPM(200)
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      COMMON /BLK2/ CPMOD(50)
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE
      DIMENSION A(201), E(4)
      EQUIVALENCE (E(1), XSLE)
      REWIND  9
      NSING = 6
      IBM5 = IB - NSING + 1
      NL = NCTS - MCPS
      IF (IND(1) .EQ. 1) GO  TO 50
      DO 20 IZ=1,NB
      CALL  VELOC (UO, VO, XBP(IZ), YBP(IZ), VM, IZ)
      A(NPNTS) = VO * COSSLP(IZ) - UO * SINSLP(IZ)
      WRITE  (1)  (U(J),V(J),J=1,NCPTS)
      IF (IND(8).NE.0) WRITE (6,190)  IZ,(J,U(J),V(J),J=1,NCPTS)
      DO 10 IY=1,NCPTS
      A(IY) = U(IY) * SINSLP(IZ) - V(IY) * COSSLP(IZ)
10    CONTINUE
      IF  (IND(8) .NE. 0) WRITE  (6,200)  IZ, (J, A(J), J = 1, NPNTS)
      WRITE  (9)  (A(J), J = 1, NPNTS )
20    CONTINUE
      IF  (IND(4) .EQ. 1) GO  TO 90
      DO 40 IZ = 1, NMOD
      A(NPNTS) = CPMOD(IZ)
      DO 30 IY=1, NCPTS
      A(IY) = C.0
30    CONTINUE
      A(MCPS+IZ-1) = 1.0
      I = IZ + NPTS
      IF  (IND(8) .NE. 0) WRITE  (6,200)   I, (J, A(J), J = 1, NPNTS)
      WRITE  (9)  (A(J), J = 1, NPNTS)
40    CONTINUE
      GO  TO 90
50    CONTINUE
      DO 80 IZ = 1, NB
      CALL  VELOC (UO, VO, XBP(IZ), YBP(IZ), VM, IZ)
      A (NPNTS) = USUBI * (SN * COSSLP(IZ) - CS * SINSLP(IZ))
      WRITE  (1)  (U(J), V(J), J = 1, NCPTS)
      IF  (IND(8) .NE. 0) WRITE (6,190)  IZ, (J, U(J), V(J), J = 1, NCPT
     1S)
      DO 60 IY = 1, NCPTS
      A(IY) = U(IY) * SINSLP(IZ) - V(IY) * COSSLP(IZ)
60    CONTINUE
      IF  (IND(8) .NE. 0) WRITE  (6,200)  IZ, (J, A(J), J = 1, NPNTS)
      WRITE  (9)  (A(J), J = 1, NPNTS)
```

```
      IF  (IZ.LT.MCPS.OR.IZ.GT.MCPF.OR.IND(4).EQ.1) GO  TO 80           XOVN0550
      A(NPNTS) = USUBI * (SQRT(1.0 - CPMOD(IZ-MCPS+1))- CS *            XOVN0560
     1            COSSLP(IZ) - SN * SINSLP(IZ))                          XOVN0570
      DO 70 IY = 1, NCPTS                                                XOVN0580
      A(IY) = U(IY) * COSSLP(IZ) + V(IY) * SINSLP(IZ)                    XOVN0590
      IF (IY .EQ. IZ .OR. IY .EQ. IZ + 1) A(IY) = A(IY) - 0.25          XOVN0600
70    CONTINUE                                                           XOVN0610
      I = NL + IZ                                                        XOVN0620
      IF  (IND(8) .NE. 0) WRITE (6,200)  I , (J, A(J), J = 1, NPNTS)    XOVN0630
      WRITE  (9)  (A(J), J = 1, NPNTS)                                   XOVN0640
80    CONTINUE                                                           XOVN0650
90    CONTINUE                                                           XOVN0660
      DO 100 I=2,NPNTS                                                   XOVN0670
      A(I) = 0.0                                                         XOVN0680
100   CONTINUE                                                           XOVN0690
      A(NPTS) = 1.0                                                      XOVN0700
      A(1) = 1.0                                                         XOVN0710
      IF  (IND(8) .NE. 0) WRITE  (6,200)  NPTS, (J, A(J), J = 1, NPNTS) XOVN0720
      WRITE  (9)  (A(J), J = 1, NPNTS )                                  XOVN0730
      IF  (IND(4) .EQ. 1) GO  TO 180                                     XOVN0740
      DO 130 IZ = 1, 2                                                   XOVN0750
      IZ2 = IZ + IZ                                                      XOVN0760
      I = IZ2 - 1                                                        XOVN0770
      CALL  VELOC  (U0, V0, E(I), F(IZ2), VM, 202)                       XOVN0780
      DO 110 IY = 1, NCPTS                                               XOVN0790
      A(IY) = U(IY)                                                      XOVN0800
110   CONTINUE                                                           XOVN0810
      A(NPNTS) = - U0                                                    XOVN0820
      IY = IBM5 + I                                                      XOVN0830
      IF  (IND(8) .NE. 0) WRITE (6,200)  IY, (J, A(J), J = 1, NPNTS)    XOVN0840
      WRITE  (9)  (A(J), J = 1, NPNTS)                                   XOVN0850
      DO 120 IY = 1, NCPTS                                               XOVN0860
      A(IY) = V(IY)                                                      XOVN0870
120   CONTINUE                                                           XOVN0880
      A(NPNTS) = - V0                                                    XOVN0890
      IY = IBM5 + IZ2                                                    XOVN0900
      IF  (IND(8) .NE. 0) WRITE  (6,200)  IY, (J, A(J), J = 1, NPNTS)   XOVN0910
      WRITE  (9)  (A(J), J = 1, NPNTS)                                   XOVN0920
130   CONTINUE                                                           XOVN0930
      A(NPNTS) = 0.0                                                     XOVN0940
      DO 140 IY=1, NVORT                                                 XOVN0950
      A(IY) = 0.0                                                        XOVN0960
140   CONTINUE                                                           XOVN0970
      D1 = 0.                                                            XOVN0980
      DO 150 IY=NSOR1, IB                                                XOVN0990
      CALL  DSTSQD (X(IY),Y(IY),X(IY+1),Y(IY+1),X3,Y3,0,D2,1)            XOVN1000
      A(IY) = D1 + D2                                                    XOVN1010
      D1 = D2                                                            XOVN1020
150   CONTINUE                                                           XOVN1030
      A(NCPTS) = D1                                                      XOVN1040
      IF  (IND(8) .NE. 0) WRITE  (6,200)   IB  , (J, A(J), J = 1, NPNTS)XOVN1050
      WRITE  (9)  (A(J), J = 1, NPNTS )                                  XOVN1060
C  **                                                                    XOVN1070
      CALL  FLOSAT  (USUM, TX, TY)                                       XOVN1080
      DO 160 IY = 1, NCPTS                                               XOVN1090
      A(IY) = U(IY) - V(IY)                                              XOVN1100
```

```
160   CONTINUE
      A(NPNTS) = USUBI * (( YSTE - YSLE) * CS - (XSTE - XSLE) * SN)
C  **
      IF (IND(8).EQ.0) GO  TO 170
      WRITE  (6,210)  NCPTS, (J, U(J), V(J), J = 1, NCPTS)
      WRITE  (6,200)  NCPTS, (J, A(J), J = 1, NPNTS)
170   CONTINUE
      WRITE  (9)    (A(J), J = 1, NPNTS)
180   CONTINUE
C  **
      END  FILE  1
      REWIND  1
      END  FILE  9
      REWIND  9
      RETURN
C
190   FORMAT (//I12,10X,25HVELOCITY COMPONENTS (U,V)/,3(I5,1H.,2E14.6))
200   FORMAT (/I12,10X,22HINFLUENCE COEFFICIENTS/,5(I5,1H.,E20.8))
210   FORMAT  (//I12, 10X, 24HSTREAM FUNCTION (LE, TE) / 3(I5, 1H.,
     1      2E14.6))
      END
```

```
      SUBROUTINE  PRSS                                                  XOVR0010
C  **                                                                   XOVR0011
C  **                SUBROUTINE PRSS DIRECTS THE CALCULATION AND WRITING XOVR0012
C  **            OF THE PRESSURE COEFFICIENTS                           XOVR0013
C  **                                                                   XOVR0014
C  **                                                                   XOVR0020
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB                XOVR0030
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,XOVR0040
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ                      XOVR0050
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)   XOVR0060
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)     XOVR0070
      COMMON /BLKE/ U(200),V(200),CPUP(200),CPLR(200),CPM(200)          XOVR0080
      COMMON /BLKG/ GAMMA(200)                                          XOVR0090
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT   XOVR0110
      LOGICAL BLIP                                                      XOVR0120
C  **                                                                   XOVR0130
      DATA KNSUB / 4HPRSS/                                              XOVR0140
      BLIP = .TRUE.                                                     XOVR0190
      DO 30 IZ=1,NB                                                     XOVR0200
      IY = IZ + 1                                                       XOVR0210
      READ  (1)   (U(J),V(J),J=1,NCPTS)                                 XOVR0220
      IF   (EOF(5) .NE. 0.0) CALL   KEOF   (5, 1, KNSUB, 114)           XOVR0230
      UVT = USUBI * CS                                                  XOVR0240
      VVT = USUBI * SN                                                  XOVR0250
      DO 10 K=1, NCPTS                                                  XOVR0260
      UVT = UVT + U(K) * GAMMA(K)                                       XOVR0270
10    VVT = VVT + V(K) * GAMMA(K)                                       XOVR0280
      CALL  CLCR (GAMMA(IZ),GAMMA(IY),UVT,VVT,COSSLP(IZ),SINSLP(IZ),    XOVR0290
     1            CPUP(IZ),CPLR(IZ))                                    XOVR0300
      IF (IND(8).NE.0) WRITE (6,40)  IZ,GAMMA(IZ),UVT,VVT,COSSLP(IZ),   XOVR0310
     1                  SINSLP(IZ),CPUP(IZ)                             XOVR0320
      CPM(IZ)= CPUP(IZ)                                                 XOVR0330
      IF (XBP(IZ).LE.X(IZ)) GO  TO 20                                   XOVR0340
      IF (BLIP) IULT = IZ                                               XOVR0350
      BLIP = .FALSE.                                                    XOVR0360
20    CONTINUE                                                          XOVR0370
30    CONTINUE                                                          XOVR0380
      REWIND  1                                                         XOVR0390
      WRITE  (6,50)                                                     XOVR0400
      CALL  CRWTB (CPUP,CPLR,CPM,IB)                                    XOVR0410
      RETURN                                                            XOVR0420
C                                                                       XOVR0430
40    FORMAT (20X,13HPRESSURE DUMP/(I10,6E15.6))                        XOVR0440
50    FORMAT (1H1,63X,29HPRESSURE DISTRIBUTION ON WING)                 XOVR0450
      END                                                               XOVR0460
```

```
      SUBROUTINE  RD
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)
      COMMON /BLKJ/ TITLE(12)
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      COMMON /BLK2/ CPMOD(50)
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE
      DATA KNSUB /2HRD /
C  **            SUBROUTINE 'RD' PERFORMS THE INPUT FUNCTIONS FOR
C  **        EACH PROBLEM.  THE NECESSARY VARIABLES ARE CHECKED FOR
C  **        RANGE, AND ERRORS CAUSE PROBLEM REJECTION...
C  **
C  **                        1 - SINGULARITY, INPUT (0)
C  **                                        CALCULATION (1)
C  **                        2 - SHAPE OF SLAT REQUIRED, YES(0)
C  **                                                 NO (1)
C  **                        4 - SLAT REQUIRED, YES(0)
C  **                                          NO (1)
C  **                        5 - PRESSURES REQUIRED... (0,1)
C  **                        6 - FORCES REQUIRED WITH DETAILED OUTPUT,
C  **                            FORCES REQUIRED WITH OUT DETAILED
C  **                            OUTPUT, NO FORCES REQUIRED... (0,1,2)
C  **                        8 - PRINT DUMP, NO (0)
C  **                                       YES (1)
C  **                   -- NSTLNS - NUMBER OF STREAMLINES...
C  **                   -- NVLPTS - NUMBER OF POINTS AT WHICH VELOCITY
C  **                                IS REQUIRED...
C  **                   -- TITLE - PROBLEM INPUT TITLE...
C  **                   -- XMBAR, YMBAR - COORDINATES ABOUT WHICH
C  **                                     MOMENTS ARE CALCULATED...
C  **                   -- USUBI - FREE STREAM VELOCITY...
C  **                   -- CREF - REFERENCE CHORD...
C  **                   -- XTE, YTE - REFERENCE TRAILING EDGE
C  **                   -- XLE, YLE - REFERENCE LEADING EDGE
C  **                   -- XSUBC - X-VALUE TO WHICH STREAMLINES ARE
C  **                              CALCULATED...
C  **                   -- DLTLMN - MINIMUM DELTA-L...
C  **                   -- DLTALN - NOMINAL DELTA-L...
C  **                   -- DLTALM - MAXIMUM DELTA-L...
C  **                   -- TOLLMT - TOLERANCE USED IN CALCULATING
C  **                                STREAMLINES...
C  **                   -- ALPHA - ANGLE/S OF ATTACK...
C  **                   -- XS, YS - COORDINATES AT WHICH STREAMLINES
C  **                                BEGIN...
C  **                   -- XV, YV - COORDINATES AT WHICH VELOCITY IS TO
C  **                                BE CALCULATED...
C  **                   -- X, Y - COORDINATES OF CORNER POINTS...
C  **
      READ  (5,160)  (TITLE(I), I = 1, 6), IK,   (TITLE(I), I = 7, 12)
      IF   (EOF(5) .NE. 0.0) CALL  KEOF  (5, 0, KNSUB, 101)
      READ  (5,170)   IND
      IF   (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 102)
      DO 10 I = 1, 15
      IF   (IND(I) .LT. 0) WRITE  (6,190)  I
```

```
10    CONTINUE                                                          XOVJ0570
C  **                                                                   XOVJ0580
C  **                                                                   XOVJ0590
      READ (5,170)  NPTS, NVOR, NSOR, MCPS, NSTLNS, NVLPTS              XOVJ0600
      IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 103)              XOVJ0610
      IF  ((NSTLNS .GE. 0) .AND. (NSTLNS .LE. 10)) GO  TO 20            XOVJ0620
      WRITE  (6,200)                                                    XOVJ0630
      CALL  EXIT                                                        XOVJ0640
20    CONTINUE                                                          XOVJ0650
      IF  ((NVLPTS .GE. 0) .AND. (NVLPTS .LE. 100)) GO  TO 30           XOVJ0660
      WRITE  (6,210)                                                    XOVJ0670
      CALL  EXIT                                                        XOVJ0680
30    CONTINUE                                                          XOVJ0690
      IF  ((IND(5) .LE. 1) .AND. (IND(6) .LE. 2)) GO  TO 40             XOVJ0700
      WRITE  (6,220)                                                    XOVJ0710
      CALL  EXIT                                                        XOVJ0720
40    CONTINUE                                                          XOVJ0730
      READ  (5,180)  XMBAR, YMBAR                                       XOVJ0740
      IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 104)              XOVJ0750
      READ (5, 180) USUBI, XLE, YLE, XTE, YTE                           XOVJ075
      READ (5, 180) XSLE, YSLE, XSTE, YSTE                              XOVJ0761
      IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 105)              XOVJ0770
      CALL  DSTSQD  (XTE,YTE,XLE,YLE,XDF,YDF,CRFSQ,CREF,1)              XOVJ0780
      IF  (CREF .GE. 1.0 E-06) GO  TO 50                                XOVJ0790
      WRITE  (6,230)                                                    XOVJ0800
      CALL  EXIT                                                        XOVJ0810
50    CONTINUE                                                          XOVJ0820
      READ (5,180)  ALPHA                                               XOVJ0830
      IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 106)              XOVJ0840
      NSING = 6                                                         XOVJ0850
      NMOD = NVOR + NSOR - NSING                                        XOVJ0860
      NVORT = NPTS + NVOR                                               XOVJ0870
      NSOR1 = NVORT + 1                                                 XOVJ0880
      NCPTS = NVORT + NSOR                                              XOVJ0890
      MCPF = MCPS + NMOD-1                                              XOVJ0900
      NB = NPTS - 1                                                     XOVJ0910
      NCTS = NPTS + 1                                                   XOVJ0920
      NPNTS = NCPTS + 1                                                 XOVJ0930
      IB = NCPTS - 1                                                    XOVJ0940
      IF  (NMOD .GT. 0) GO  TO 60                                       XOVJ0950
      IF  ((NSOR+NVOR) .LE. 0) GO  TO 60                                XOVJ0960
      WRITE  (6,240)                                                    XOVJ0970
      CALL  EXIT                                                        XOVJ0980
60    CONTINUE                                                          XOVJ0990
      IF  ((NCPTS .GT. 0) .AND. (NCPTS .LE. 200)) GO  TO 70             XOVJ1000
      WRITE  (6,250)                                                    XOVJ1010
      CALL  EXIT                                                        XOVJ1020
70    CONTINUE                                                          XOVJ1030
      IF  ((NPTS .GT. 0).AND.(NVOR .GE. 0).AND.(NSOR .GE. 0)) GO  TO 80 XOVJ1040
      WRITE  (6,260)                                                    XOVJ1050
      CALL  EXIT                                                        XOVJ1060
80    CONTINUE                                                          XOVJ1070
      IF  (IND(1) .EQ. 1) GO  TO 100                                    XOVJ1080
      IF  ((MCPS .GE. NCTS) .AND. (MCPF .LE. NPNTS)) GO  TO 90          XOVJ1090
      WRITE  (6,270)                                                    XOVJ1100
      CALL  EXIT                                                        XOVJ1110
```

```
90     CONTINUE                                                         XOVJ1120
       GO  TO 110                                                       XOVJ1130
100    CONTINUE                                                         XOVJ1140
       IF  ((MCPS .GT. 0) .AND. (MCPF .LE. NPTS)) GO  TO 110            XOVJ1150
       WRITE  (6,270)                                                   XOVJ1160
       CALL  EXIT                                                       XOVJ1170
110    CONTINUE                                                         XOVJ1180
       READ (5,180)  (X(I),Y(I),I=1,NPTS)                               XOVJ1190
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 107)             XOVJ1200
       IF  (IND(4) .GT. 0) GO  TO 120                                   XOVJ1210
       READ (5,180)  (X(I+NPTS),Y(I+NPTS),I=1,NVOR)                     XOVJ1220
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 108)             XOVJ1230
       READ (5,180)  (X(I),Y(I),I=NSOR1, NCPTS)                         XOVJ1240
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 109)             XOVJ1250
       READ  (5,180)  (CPMOD(I), I = 1, NMOD)                           XOVJ1260
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 110)             XOVJ1270
120    CONTINUE                                                         XOVJ1280
       IF  (USUBI.EQ.0) USUBI = 1.                                      XOVJ1290
       IF  (NSTLNS .EQ. 0) GO  TO 140                                   XOVJ1300
       READ  (5,180)  XSUBC,DLTLMN,DLTALN,DLTALM,TOLLMT,CRT             XOVJ1310
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 111)             XOVJ1320
       IF  ((DLTLMN .LE. DLTALN) .AND. (DLTALN .LE. DLTALM)) GO  TO 130 XOVJ1330
       WRITE  (6,280)                                                   XOVJ1340
       CALL  EXIT                                                       XOVJ1350
130    CONTINUE                                                         XOVJ1360
       READ  (5,180)  (XS(I), YS(I),  I = 1, NSTLNS)                    XOVJ1370
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 112)             XOVJ1380
140    CONTINUE                                                         XOVJ1390
       IF  (NVLPTS .EQ. 0) GO  TO 150                                   XOVJ1400
       READ  (5,180)  (XV(I), YV(I),  I = 1, NVLPTS )                   XOVJ1410
       IF  (EOF(5) .NE. 0.0) CALL  KEOF  (5, 1, KNSUB, 113)             XOVJ1420
150    CONTINUE                                                         XOVJ1430
       CALL  WRTE                                                       XOVJ1440
        RETURN                                                          XOVJ1450
C                                                                       XOVJ1460
160    FORMAT  (6A10, 4X, I1 / 6A10 )                                   XOVJ1470
170    FORMAT  (15I3)                                                   XOVJ1480
180    FORMAT  (6F10.0)                                                 XOVJ1490
190    FORMAT  (1H0, 48H*** IN SUBROUTINE RD IND(I) IS LESS THAN ZERO I=,XOVJ1500
     1         I3, 4H *** )                                             XOVJ1510
200    FORMAT  (1H0, 29H*** INPUT ERROR IN NSTLNS *** )                 XOVJ1520
210    FORMAT  (1H0, 29H*** INPUT ERROR IN NVLPTS *** )                 XOVJ1530
220    FORMAT  (1H0, 48H*** INPUT ERROR IN OPTION INDICATORS 5 AND 6 *** XOVJ1540
     1)                                                                 XOVJ1550
230    FORMAT  (1H0, 48H*** LEADING AND TRAILING EDGE ARE COINCIDENT *** XOVJ1560
     1)                                                                 XOVJ1570
240    FORMAT  (1H0, 58H*** INPUT ERROR IN NUMBER OF SOURCES AND / OR VORXOVJ1580
     1TICES *** )                                                       XOVJ1590
250    FORMAT  (1H0, 48H*** INPUT ERROR,NO. OF CONTROL PTS. IN ERROR *** XOVJ1600
     1)                                                                 XOVJ1610
260    FORMAT  (1H0, 65H*** INPUT ERROR IN NUMBER OF CORNER, VORTEX, OR SXOVJ1620
     1OURCE POINTS *** )                                                XOVJ1630
270    FORMAT  (1H0, 60H*** INPUT ERROR OF INDEXES OF PRESSURES TO BE SPEXOVJ1640
     1CIFIED  *** )                                                     XOVJ1650
280    FORMAT  (1H0, 44H*** INPUT ERROR IN STREAMLINE INPUT CARD *** )  XOVJ1660
       END                                                              XOVJ1670
```

```
      SUBROUTINE  RMTVTY                                                XOVX0010
C  **                                                                   XOVX0020
      COMMON /BLKA/ INC(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB                XOVX0030
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,XOVX0040
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ                      XOVX0050
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)   XOVX0060
      DIMENSION HD(11)                                                  XOVX0070
      DATA HD /  10H NO.       , 10H   XV      , 10H     YV             XOVX0080
     1        ,  10H   XV/CREF, 10H    YV/CRE, 10HF   MODULUS           XOVX0090
     2        ,  10H        MAG, 10H.        U, 10H                     XOVX0100
     3        ,  10HV         , 10H FLOW                               /XOVX0110
C  **                                                                   XOVX0120
C  **               SUBROUTINE 'RMTVTY' FINDS THE VELOCITY AT ALL       XOVX0130
C  **          REMOTE POINTS... THE COORDINATES, VELOCITY COMPONENTS,   XOVX0140
C  **          MODULUS AND MAGNITUDE ARE WRITTEN...                     XOVX0150
C  **                                                                   XOVX0160
      WRITE  (6,20)                                                     XOVX0170
      WRITE  (6,30)    HD                                               XOVX0180
      I = 0                                                             XOVX0190
      DO 10 J = 1, NVLPTS                                               XOVX0200
      CALL  VELOC (U0,V0,XV(J),YV(J),VL,-1)                             XOVX0210
      CALL  FLOCAL  (USUM, XV(J), YV(J))                                XOVX0220
      XL = (XV(J) - XLE) / CREF                                         XOVX0230
      YL = (YV(J) - YLE) / CREF                                         XOVX0240
      TN = ATAN(V0/U0) * 57.29578                                       XOVX0250
      WRITE  (6,40)  J, XV(J), YV(J), XL, YL, TN, VL, U0, V0, USUM      XOVX0260
      I = I + 1                                                         XOVX0270
      IF  (I .LT. 3) GO  TO 10                                          XOVX0280
      WRITE  (6,30)                                                     XOVX0290
      I = 0                                                             XOVX0300
10    CONTINUE                                                          XOVX0310
      RETURN                                                            XOVX0320
C                                                                       XOVX0330
20    FORMAT  (1H1, 53X, 25HVELOCITY AT REMOTE POINTS /  )              XOVX0340
30    FORMAT  (1H0, 11A10)                                              XOVX0350
40    FORMAT  (1X, I3, 3X, 2(F9.4, 3X), 6(F8.4, 3X), F10.5 )            XOVX0360
      END                                                               XOVX0370
```

```
      SUBROUTINE  SBFRCE  (CP, X, Y, XBP, YBP, COSSLP, SINSLP, NN
     1        , NNP1, CN2, CL1, CC2, CD1, CM2, CS, SN)
C  **
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
C  **
      DIMENSION HD(9), X(NNP1), Y(NNP1), XBP(NN), YBP(NN), COSSLP(NN)
     1     , SINSLP(NN), CP(NN)
      DATA HD / 10H          , 10H          , 10H
     1     ,     10H          , 10H NO.     , 10H     DC-
     2     ,     10HN         , 10H DC-C    , 10H     DC-M  /
C  **
C  **              SUBROUTINE SBFRCE COMPUTES THE FORCE COEFFICIENTS
C  **
      LMP1 = 1
      ICP1 = 1
      CPX = (CP(ICP1) + CP(ICP1 + NN - 1))/2.
      CN2 = 0.
      CC2 = 0.
      CM2 = 0.
      IF  (IND(6).EQ.1) GO  TO 10
      WRITE  (6,70)  HD
10    CONTINUE
      DO 60 J = 1, NN
      LM = LMP1
      LMP1 = LM + 1
      IC = ICP1
      ICP1 = IC + 1
      IF  (J.EQ.1) GO  TO 20
      CP1 = CP2
      S1 = S2
      GO  TO 30
20    CONTINUE
      CALL  DSTSQD (XBP(IC), YBP(IC), X(LM), Y(LM), XDF, YDF, S1, S1, 1)
      CP1 = CPX
30    CONTINUE
      A = 0.5 * (- S1)
      IF  (J.EQ.NN) GO  TO 40
      CALL  DSTSQD (XBP(ICP1), YBP(ICP1), X(LMP1), Y(LMP1), XDF, YDF,
     1              S2, S2, 1)
      CP2 = (S1 * CP(IC)  +  S2 * CP(ICP1))/(S1 + S2)
      GO  TO 50
40    CONTINUE
      CP2 = CPX
50    CONTINUE
      DF1 = A * (CP1 + CP(IC))
      DF2 = A * (CP2 + CP(IC))
      DFN1 = DF1 * COSSLP(IC)
      DFN2 = DF2 * COSSLP(IC)
      DFC1 = - DF1 * SINSLP(IC)
      DFC2 = - DF2 * SINSLP(IC)
      DFN = (DFN1 + DFN2)/CREF
      DFC = (DFC1 + DFC2)/CREF
      DM = (DFN1 * (XMBAR - (XBP(IC) + X(LM))/2.)
     1    - DFC1 * (YMBAR - (YBP(IC) + Y(LM))/2.)
```

```
      2     + DFN2 * (XMBAR - (XBP(IC) + X(LMP1))/2.)                        XOVV0570
      3     - DFC2 * (YMBAR - (YBP(IC) + Y(LMP1))/2.))/CRFSQ                 XOVV0580
       IF  (IND(6).EQ.0) WRITE  (6,80)    J, DFN, DFC, DM                    XOVV0590
       CN2 = CN2 + DFN                                                       XOVV0600
       CC2 = CC2 + DFC                                                       XOVV0610
       CM2 = CM2 + DM                                                        XOVV0620
 60    CONTINUE                                                              XOVV0630
       CL1= CN2 * CS - CC2 * SN                                              XOVV0640
       CD1= CN2 * SN + CC2 * CS                                              XOVV0650
       RETURN                                                                XOVV0660
 C                                                                           XOVV0670
 70    FORMAT  (1H0, 9A10)                                                   XOVV0680
 80    FORMAT  (41X, I3, 5X, 3(5X, F9.4))                                    XOVV0690
       END                                                                   XOVV0700
```

```
      SUBROUTINE  SHAPE                                                 XOVP0010
C  **                                                                   XOVP0011
C  **              SUBROUTINE SHAPE COMPUTES AND WRITES THE GECMETRY    XOVP0012
C  **          OF THE LEADING EDGE SLAT                                 XOVP0013
C  **                                                                   XOVP0014
C                                                                       XOVP0050
      COMMON /BLKA/ INC(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB                XOVP0060
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,XOVP0070
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ                      XOVP0080
      COMMON /BLKJ/ TITLE(12)                                           XOVP0090
      COMMON /BLKK/ SCOSLP(160), CPSLAT(160), SSINLP(160), XSLAT(161)   XOVP0100
     1    , XXBP(160), YSLAT(161), YYBP(160)                            XOVP0110
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE                              XOVP0120
      LOGICAL LOWER                                                     XOVP0130
      DIMENSION A(80), B(80), SX(81), SXH(80)                           XOVP0140
     1    , SY(81), SYH(80), U(80), UH(80), V(80), VH(80), CAB(80)      XOVP0150
     2    , G(3, 3), GA(3), DABUV(80), ANGAB(80), ANGUV(80)             XOVP0160
     3    , ANDAB(80), ANDUV(80), GF(3), SAVAB(80)                      XOVP0170
      EQUIVALENCE (GA(1), DLAM), (GA(2), DMU), (GA(3), DL)              XOVP0180
      DATA FIVE, RADCON / 5.0, 57.29578 /                               XOVP0190
      DATA EPS, ACEPT, ZERO, PTTWO, HALF, ONE, ONE25, TWO               XOVP0200
     1    /.00001, 0.0001  , 0.0, 0.2, 0.5, 1.0, 1.25, 2.0/             XOVP0210
      DATA IZERO, IONE, ITWO, ITEN       / 0, 1, 2, 10    /             XOVP0220
C                                                                       XOVP0230
C                                                                       XOVP0240
      SX(1) = XSLE                                                      XOVP0250
      SY(1) = YSLE                                                      XOVP0260
      DELX = XSLE - XSTE                                                XOVP0270
      DELY = YSLE - YSTE                                                XOVP0280
      DELX2 = DELX * DELX                                               XOVP0290
      DELY2 = DELY * DELY                                               XOVP0300
      DIST2 = DELX2 + DELY2                                             XOVP0310
      IF  (DIST2 .GE. EPS) GO  TO 10                                    XOVP0320
      WRITE  (6,430)                                                    XOVP0330
      CALL  EXIT                                                        XOVP0340
10    CONTINUE                                                          XOVP0350
      DIST = SQRT (DIST2)                                               XOVP0360
      XMT = DIST * HALF                                                 XOVP0370
      YSX = DELY / DELX                                                 XOVP0380
      THETA = ATAN (YSX)                                                XOVP0390
      IF  (ABS(YSX) .LT. EPS) YSX = ONE                                 XOVP0400
      DMUSAV = - (EPS * YSX)                                            XOVP0410
      COSETA = COS(THETA)                                               XOVP0420
      SINETA = SIN(THETA)                                               XOVP0430
      SU2 = ONE / (USUBI * USUBI)                                       XOVP0460
      N = ITEN                                                          XOVP0470
      NT2 = N / ITWO                                                    XOVP0480
      NT3 = NT2 + IONE                                                  XOVP0490
C                                                                       XOVP0500
      H = DIST * HALF                                                   XOVP0510
      LOWER = .FALSE.                                                   XOVP0520
      SIGN = ONE                                                        XOVP0530
C                                                                       XOVP0540
20    CONTINUE                                                          XOVP0550
C                                                                       XOVP0560
C                                                                       XOVP0570
```

```
C          DEFINITION OF INITIAL UPPER OR LOWER SLAT SURFACE
C
      ITER = IZERO
      IT = IZERO
      DLAM = EPS
      DMU = DMUSAV
      SN = N
      SN = ONE / SN
      DL = DIST * SN * SQRT(TWO)
      NM1 = N - IONE
      SX(N+1) = XSTE
      SY(N+1) = YSTE
      IF (IND(9) .EQ. 1) GO TO 171
      YMT = H * SIGN
      XMR = XMT * COSETA  - YMT * SINETA
      YMR = XMT * SINETA  + YMT * COSETA
      XM = XMR + XSLE
      YM = YMR + YSLE
      SN2 = SN + SN
      AA = XMR * SN2
      BB = YMR * SN2
      THETA = ATAN2(BB, AA)
      SUMA = FIVE * AA
      SUMB = FIVE * BB
C
      DO 30 I = 1, NT2
      A(I) = AA
      B(I) = BB
      ANGAB(I) = THETA
30    CONTINUE
C
      AA = (XSTE - XM) * SN2
      BB = (YSTE - YM) * SN2
      THETA = ATAN2(BB, AA)
      SUMA = SUMA + FIVE * AA
      SUMB = SUMB + FIVE * BB
      DO 40 I = NT3, N
      A(I) = AA
      B(I) = BB
      ANGAB(I) = THETA
40    CONTINUE
      GO  TO 140
50    CONTINUE
C  **
C          A RESTART WITH DOUBLE THE NUMBER OF PREVICUS SEGMENTS
C          UNTIL N = 80, WITH N = 80 ITERATE UNTIL CCNVERGENCE
C
C  **
      DL = (DL + SAVDL) * HALF
      SUMA = ZERO
      SUMB = ZERO
      DO 60 I = 1, N
      ANGAB(I) = (ANGAB(I) + SAVAB(I)) * HALF
      A(I) = COS (ANGAB(I)) * DL
      SUMA = SUMA + A(I)
      B(I) = SIN (ANGAB(I)) * DL
```

```
      SUMB = SUMB + B(I)                                                XOVP1130
60    CONTINUE                                                          XOVP1140
      DNORMA = - DELX / SUMA                                            XOVP1150
      IF ((ABS(DELY).GT. ACEPT).AND.(ABS(SUMB).GT. ACEPT)) GO  TO 70    XOVP1160
      DNORMB = ONE                                                      XOVP1170
      GO  TO 80                                                         XOVP1180
70    CONTINUE                                                          XOVP1190
      DNORMB = - DELY / SUMB                                            XOVP1200
80    CONTINUE                                                          XOVP1210
      IF  (IND(8) .EQ. IZERO) GO  TO 90                                 XOVP1220
      WRITE  (6,440)  DNORMA, DNORMB                                    XOVP1230
90    CONTINUE                                                          XOVP1240
      DORMDL =  (DNORMA + DNORMB) * HALF                                XOVP1250
      DL = DL * DORMDL                                                  XOVP1260
      SUMA = SUMA * DORMDL                                              XOVP1270
      SUMB = SUMB * DORMDL                                              XOVP1280
      DO 100 I = IONE, N                                                XOVP1290
      A(I) = A(I) * DORMDL                                              XOVP1300
      B(I) = B(I) * DORMDL                                              XOVP1310
100   CONTINUE                                                          XOVP1320
      DMU = (DMU + SAVMU) * HALF                                        XOVP1330
      DLAM = (DLAM + SAVLA) * HALF                                      XOVP1340
      IF  (N .LE. 40) GO  TO 120                                        XOVP1350
      IT = IT + IONE                                                    XOVP1360
      DLDIF = DL - SAVDL                                                XOVP1370
      IF  (SAVDL .GE. EPS) GO  TO 110                                   XOVP1380
      WRITE  (6,450)                                                    XOVP1390
      CALL  EXIT                                                        XOVP1400
110   CONTINUE                                                          XOVP1410
      DLDIF = ABS (DLDIF / SAVDL)                                       XOVP1420
      WRITE  (6,460)  IT, DLDIF                                         XOVP1430
      IF  (DLDIF .LT. ACEPT) GO  TO 390                                 XOVP1440
      IF  (IT .GT. ITEN) GO  TO 380                                     XOVP1450
      GO  TO 140                                                        XOVP1460
120   CONTINUE                                                          XOVP1470
      JL = N + N                                                        XOVP1480
      JS = N                                                            XOVP1490
      DO 130 I = 1, N                                                   XOVP1500
      A(JL) = A(JS) * HALF                                              XOVP1510
      B(JL) = B(JS) * HALF                                              XOVP1520
      ANGAB(JL) = ANGAB(JS)                                             XOVP1530
      JL = JL - IONE                                                    XOVP1540
      A(JL) = A(JL+1)                                                   XOVP1550
      B(JL) = B(JL+1)                                                   XOVP1560
      ANGAB(JL) = ANGAB(JS)                                             XOVP1570
      JL = JL - IONE                                                    XOVP1580
      JS = JS - IONE                                                    XOVP1590
130   CONTINUE                                                          XOVP1600
      DL = DL * HALF                                                    XOVP1610
      ITER = IZERO                                                      XOVP1620
      N = N + N                                                         XOVP1630
      SN = N                                                            XOVP1640
      SN = ONE / SN                                                     XOVP1650
      NM1 = N - IONE                                                    XOVP1660
      SX(N+1) = XSTE                                                    XOVP1670
      SY(N+1) = YSTE                                                    XOVP1680
```

```
C  **
140    CONTINUE
C  **
       IF  (N .LT. 80) GO  TO 160
       SAVDL = DL
       SAVMU = DMU
       SAVLA = DLAM
       DO 150 I = 1, N
       SAVAB(I) = ANGAB(I)
150    CONTINUE
160    CONTINUE
C
C          CALCULATE THE SEGMENTS AND THE VELOCITIES AT THE MIDPOINTS
C
       DO 170 I = 1, NM1
       SX(I+1) = SX(I) + A(I)
       SY(I+1) = SY(I) + B(I)
       SXH(I) = (SX(I+1) + SX(I)) * HALF
       SYH(I) = (SY(I+1) + SY(I)) * HALF
       CALL  VELOC  (UH(I), VH(I), SXH(I), SYH(I), W, IZERO)
       U(I) = UH(I) / W
       V(I) = VH(I) / W
170    CONTINUE
       GO TO 179
171    CONTINUE
       READ (5, 172) (SX(I), SY(I), I = ITWO, ITEN)
172    FORMAT (6F10.0)
       DL = ZERO
       DO 174     I = IONE, ITEN
       A(I) = SX(I+1) - SX(I)
       B(I) = SY(I+1) - SY(I)
       ANGAB(I) = ATAN2(B(I), A(I))
       DL  = DL  + SQRT  (A(I) * A(I) + B(I) * B(I))
174    CONTINUE
       DL  = DL * 0.1
       DO 176     I = IONE, 9
       SXH(I) = (SX(I+1) + SX(I)) * HALF
       SYH(I) = (SY(I+1) + SY(I)) * HALF
       CALL VELOC (UH(I), VH(I), SXH(I), SYH(I), W, IZERO)
       U(I) = UH(I) / W
       V(I) = VH(I) / W
176    CONTINUE
179    CONTINUE
       SXH(N) = (SX(N) + SX(N+1)) * HALF
       SYH(N) = (SY(N) + SY(N+1)) * HALF
       CALL  VELOC  (UH(N), VH(N), SXH(N), SYH(N), W, IZERO)
       U(N) = UH(N) / W
       V(N) = VH(N) / W
C
C          CALCULATE THE TOTAL ANGULAR DEVIATIONS BETWEEN LOCAL
C          STREAMLINE SLOPE AND SEGMENTED LINE SLOPE
C
       ITER = ITER + IONE
       ERR2 = ZERO
       DO 180 I = 1, N
       ANGUV(I) = ATAN2 (V(I), U(I))
```

```
      ANDUV(I) = ANGUV(I) * RADCON
      ANDAB(I) = ANGAB(I) * RADCON
      CAB(I) = ANGAB(I) - ANGUV(I)
      DABUV(I) = CAB(I) * RADCON
      ERR2 = ERR2 + ABS(DABUV(I))
180   CONTINUE
C
      IF  (IND(8) .EQ. IZERO) GO  TO 190
C
      WRITE  (6,470)  DLAM, DMU
      WRITE  (6,480)  N, DL, ERR2
      WRITE  (6,490)  (I, SX(I), SY(I), U(I), V(I), ANDUV(I), ANDAB(I)
     1       , DABUV(I), I = 1, N)
C
190   CONTINUE
C
      DO 350 M = IONE, ITEN
C          CALCULATE NEW ANGLES OF SEGMENTED LINE
C
      DO 230 I = IONE, N
      DO 210 M1 = IONE, ITEN
      ABSIN = SIN(ANGAB(I)) * DL
      ABCOS = COS(ANGAB(I)) * DL
      SUM = TWO + DMU * ABCOS + DLAM * ABSIN
      IF  (ABS(SUM) .GE. EPS) GO  TO 200
      WRITE  (6,500)
      CALL  EXIT
200   CONTINUE
      CAB(I) = ANGAB(I) - ANGUV(I)
      SUM = (CAB(I) + CAB(I) + DMU * ABSIN - DLAM * ABCOS) / SUM
      ANGAB(I) = ANGAB(I) - SUM
      IF  (ABS(SUM) .LT. ACEPT) GO  TO 220
210   CONTINUE
220   CONTINUE
230   CONTINUE
      SUMA = ZERO
      SUMB = ZERO
      DO 240 I = 1, N
      A(I) = COS (ANGAB(I)) * DL
      SUMA = SUMA + A(I)
      B(I) = SIN (ANGAB(I)) * DL
      SUMB = SUMB + B(I)
240   CONTINUE
      IF  (ABS(SUMA + DELX) .LT. ACEPT) GO  TO 360
      SUMLM = ZERO
      SDL = ONE / DL
      SUMCO = SUMA * SDL
      SUMSI = SUMB * SDL
      G1MU = ZERO
      G2MU = ZERO
      G3MU = ZERO
      G1LAM = ZERO
      G2LAM = ZERO
      G3LAM = ZERO
      DO 260 I = 1, N
      DENCM = TWO + DLAM * B(I) + DMU * A(I)
```

```
      IF  (ABS(DENOM) .GE. EPS) GO  TO 250                              XOVP2610
      WRITE  (6,510)                                                    XOVP2620
      CALL  EXIT                                                        XOVP2630
250   CONTINUE                                                          XOVP2640
      DENOM = ONE / DENOM                                               XOVP2650
      DDDLAM = A(I) * DENOM                                             XOVP2660
      DDDMU = - B(I) * DENOM                                            XOVP2670
      DDDL = (DLAM * A(I) - DMU * B(I)) * DENOM * SDL                   XOVP2680
      G1MU = G1MU + B(I) * DDDLAM                                       XOVP2690
      G2MU = G2MU + B(I) * DDDMU                                        XOVP2700
      G3MU = G3MU + B(I) * DDDL                                         XOVP2710
      G1LAM = G1LAM + A(I) * DDDLAM                                     XOVP2720
      G2LAM = G2LAM + A(I) * DDDMU                                      XOVP2730
      G3LAM = G3LAM + A(I) * DDDL                                       XOVP2740
260   CONTINUE                                                          XOVP2750
      G(1, 1) = SUMSI + (DLAM * G1LAM - DMU * G1MU) * SDL               XOVP2760
      G(1, 2) = SUMCO + (DLAM * G2LAM - DMU * G2MU) * SDL               XOVP2770
      G(1, 3) = (DLAM * G3LAM - DMU * G3MU) * SDL                       XOVP2780
      G(2, 1) = - G1LAM                                                 XOVP2790
      G(2, 2) = - G2LAM                                                 XOVP2800
      G(2, 3) = (- SUMSI) - G3LAM                                       XOVP2810
      G(3, 1) = G1MU                                                    XOVP2820
      G(3, 2) = G2MU                                                    XOVP2830
      G(3, 3) = (- SUMCO) + G3MU                                        XOVP2840
C                                                                       XOVP2850
C            CALCULATE THE INVERSE OF G MATRIX                          XOVP2860
C                                                                       XOVP2870
      DO 310 I = 1, 3                                                   XOVP2880
      IF  (ABS(G(I,I)) .GE. 1.0E-10) GO  TO 270                         XOVP2890
      WRITE  (6,520)                                                    XOVP2900
      CALL  EXIT                                                        XOVP2910
270   CONTINUE                                                          XOVP2920
      TE = ONE / G(I,I)                                                 XOVP2930
      G(I,I) = ONE                                                      XOVP2940
      DO 280 K = 1, 3                                                   XOVP2950
      G(I,K) = G(I,K) * TE                                              XOVP2960
280   CONTINUE                                                          XOVP2970
      DO 300 J = 1, 3                                                   XOVP2980
      IF  (J .EQ. I) GO  TO 300                                         XOVP2990
      TE = G(J,I)                                                       XOVP3000
      G(J,I) = ZERO                                                     XOVP3010
      DO 290 K = 1, 3                                                   XOVP3020
      G(J,K) = G(J,K) - G(I,K) * TE                                     XOVP3030
290   CONTINUE                                                          XOVP3040
300   CONTINUE                                                          XOVP3050
310   CONTINUE                                                          XOVP3060
C            CALCULATE NEW DLAM, DMU, DL                                XOVP3070
      DO 330 I = 1, 3                                                   XOVP3080
      GF(1) =  GA(1) * SUMSI + GA(2) * SUMCO                            XOVP3090
      GF(2) = (-DELY) - GA(3) * SUMSI                                   XOVP3100
      GF(3) = (-DELX) - GA(3) * SUMCO                                   XOVP3110
      SUM = ZERO                                                        XOVP3120
      DO 320 J = 1, 3                                                   XOVP3130
      SUM = SUM + G(I,J) * GF(J)                                        XOVP3140
320   CONTINUE                                                          XOVP3150
      GA(I) = GA(I) - SUM                                               XOVP3160
```

```
      SUMLM = SUMLM + ABS(SUM)
330   CONTINUE
      IF  (IND(8) .EQ. IZERO) GO  TO 340
      WRITE  (6,530)  SUMLM
340   CONTINUE
350   CONTINUE
360   CONTINUE
      IF  (N .EQ. 80) GO  TO 50
      IF  (ITER .GT. IONE) GO  TO 50
      DO 370 I = 1, N
      SAVAB(I) = ANGAB(I)
370   CONTINUE
      SAVDL = DL
      SAVMU = DMU
      SAVLA = DLAM
      GO  TO 140
380   CONTINUE
      WRITE  (6,540)
C
390   CONTINUE
C
      SDL = ONE / DL
      IF  (LOWER) GO  TO 410
      LOWER = .TRUE.
      SIGN = -  ONE
      N = ITEN
C         SAVE THE CORNER AND BOUNDARY POINTS FOR THE UPPER SURFACE
      J = 81
      DO 400 I = 1, 80
      XSLAT(I) = SX(J)
      YSLAT(I) = SY(J)
      XXBP(I) = SXH(J-1)
      YYBP(I) = SYH(J-1)
      CPSLAT(I) = ONE - (UH(J-1) * UH(J-1) + VH(J-1) * VH(J-1)) * SU2
      SCOSLP(I) = A(J-1) * SDL
      SSINLP(I) = B(J-1) * SDL
      J = J - 1
400   CONTINUE
      GO  TO 20
C  **
410   CONTINUE
C  **
      J = 81
      DO 420 I = 1, 80
      XSLAT(J) = SX(I)
      YSLAT(J) = SY(I)
      XXBP(J) = SXH(I)
      YYBP(J) = SYH(I)
      CPSLAT(J) = ONE - (UH(I) * UH(I) + VH(I) * VH(I)) * SU2
      SCOSLP(J) = - A(I) * SDL
      SSINLP(J) = - B(I) * SDL
      J = J + 1
420   CONTINUE
      XSLAT(J) = XSTE
      YSLAT(J) = YSTE
      WRITE  (6,550)
```

```
      WRITE  (6,560)
      WRITE  (6,570)
      WRITE  (6,580)  (J, XSLAT(J), YSLAT(J), XXBP(J), YYBP(J), CPSLAT(J
     1)       , J = 1, 40)
      WRITE  (6,550)
      WRITE  (6,560)
      WRITE  (6,570)
      WRITE  (6,580)  (J, XSLAT(J), YSLAT(J), XXBP(J), YYBP(J), CPSLAT(J
     1)       , J = 41, 80 )
      WRITE  (6,550)
      WRITE  (6,590)
      WRITE  (6,570)
      WRITE  (6,580)  (J, XSLAT(J), YSLAT(J), XXBP(J), YYBP(J), CPSLAT(J
     1)       , J = 81, 120 )
      WRITE  (6,550)
      WRITE  (6,590)
      WRITE  (6,570)
      WRITE  (6,580)  (J, XSLAT(J), YSLAT(J), XXBP(J), YYBP(J), CPSLAT(J
     1)       , J = 121, 160)
      J = 161
      WRITE   (6,580)   J, XSTE, YSTE
      RETURN
C
  430 FORMAT  (1H0, 57H*** LEADING AND TRAILING EDGE OF SLAT ARE COINCID
     1ENT  *** )
  440 FORMAT  (1H , 28HNORMALIZATION FACTOR FOR A =,E15.8, 5H  B =,E15.8
     1)
  450 FORMAT  (1H0, 41H*** IN SUBROUTINE SHAPE SAVOL IS ZERO *** )
  460 FORMAT  (1H0,  6HAFTER , I3, 53H ITERATIONS , THE RELATIVE CHANGE
     1IN SEGMENT LENGTH = , E14.5 )
  470 FORMAT  (1H0, 7HDLAM = , E14.5, 11X, 6HDMU = , E14.5 )
  480 FORMAT  (1H0, 21HNUMBER OF SEGMENTS = , I3, 3X, 17HSEGMENT LENGTH
     1= , E14.5, 3X,  8HERROR = , E14.5/1H0, 5X, 1HI, 9X, 4HX(I), 11X,
     2   4HY(I), 11X, 4HU(I), 11X, 4HV(I), 11X, 8HTHETA UV, 7X,
     3   8HTHETA AB, 7X, 6HERR(I) //)
  490 FORMAT (4X, I3, 7F15.6)
  500 FORMAT  (1H0, 41H*** IN SUBROUTINE SHAPE A SUM IS ZERO *** )
  510 FORMAT  (1H0, 49H*** IN SUBROUTINE SHAPE A DENOMINATOR IS ZERO ***
     1)
  520 FORMAT  (1H0, 54H*** IN SUBROUTINE SHAPE PIVOT ELEMENT IS TOO SMAL
     1L *** )
  530 FORMAT  (1H0, 52HABSOLUTE SUM OF CORRECTIONS TO DLAM, DMU, AND DL
     1IS , E14.5)
  540 FORMAT  (1H0, 46HCONVERGENCE HAS NOT OCCURRED WITH  80 SEGMENTS )
  550 FORMAT  (1H1, 43HSLAT COORDINATES AND PRESSURE DISTRIBUTION )
  560 FORMAT  (1H+, 43X, 21HFOR UPPER SURFACE *** )
  570 FORMAT  (1H0, 3X, 1HI, 6X, 4HX(I), 8X, 4HY(I), 6X, 6HXBP(I), 6X
     1      , 6HYBP(I), 7X, 5HCP(I) )
  580 FORMAT  (1X, I3, 5F12.5)
  590 FORMAT  (1H+, 43X, 21HFOR LOWER SURFACE *** )
      END
```

```
      SUBROUTINE  SLNR
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKG/ GAMMA(20C)
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT
      DIMENSION HD(11), A(200,201)
      DIMENSION IPIVO(200)
      DATA HD / 10H   PT      , 10H  G        , 10H        PT
     1     ,     10H       G   , 10H            , 10H PT
     2     ,     10HG          , 10H     PT    , 10H    G
     3     ,     10H          P, 10HT      G   /
      DATA EPSLN, IONE, IZERO, ONE, ZERO / 1.0E-10, 1, 0, 1.0, 0.0 /
      DATA KNSUB /4HSLNR /
C  **
C  **               SUBROUTINE  SLNR  PRODUCES THE GAMMAS REQUIRED FOR
C  **          THE SOLUTION OF THE CURRENT PROBLEM...
C  **
C  **
      DO 10 I = 1, NCPTS
C  **
      READ  (9)  (A(I, J), J = 1, NPNTS )
      IF  (EOF(9).NE. 0.C) CALL  KEOF (9, 1, KNSUB, 115)
10    CONTINUE
      REWIND  9
C  **
C         ZERO THE PIVOT INDEX ARRAY
C  **
      DO 20 I = 1, NCPTS
      IPIVO(I) = IZERO
20    CONTINUE
C
      DO 110 I = 1, NCPTS
      T = ZERO
C
C         THE FOLLOWING SELECTS THE PIVOT ELEMENT WHICH HAS THE
C                        LARGEST MAGNITUDE .
C
      DO 40 J = 1, NCPTS
      IF  (IPIVO(J) .NE. IZERO) GO  TO 40
      DO 30 K = 1, NCPTS
      IF  (IPIVO(K) .NE. IZERO) GO  TO 30
      TE = ABS (A(J,K))
      IF  (TE  .LE. T ) GO  TO 30
      T = TE
      JROW = J
      KCOL = K
30    CONTINUE
40    CONTINUE
C
C         IF A SELECTED PIVOT ELEMENT IS LESS THAN THE SINGULARITY
C                     CRITERION, CALL ERROR(112)
C
      IF  (T .GE. EPSLN) GO  TO 50
      WRITE  (6,140)
50    CONTINUE
C
      IPIVO(KCOL) = IONE
```

```
      IF  (JROW .EQ. KCOL) GO  TO 70
C
C          PUT  A(JROW, KCOL) ON DIAGONAL
C
      DO 60 J = 1, NPNTS
      TE = A (KCOL, J)
      A(KCOL, J) = A(JROW, J)
      A(JROW, J) = TE
60    CONTINUE
70    CONTINUE
C
C          DIVISION DOWN THE PIVOT ROW BY ITS MAIN DIAGONAL ELEMENT
C
      TE = ONE / A(KCOL, KCOL)
      A(KCOL, KCOL) = ONE
      DO 80 K = 1, NPNTS
      A(KCOL, K) = A(KCOL, K) * TE
80    CONTINUE
C
C          REPLACE EACH ROW BY LINEAR COMBINATION WITH PIVOT ROW
C
      DO 100 J = 1, NCPTS
      IF  (J .EQ. KCOL) GO  TO 100
      TE = A(J, KCOL)
      A(J, KCOL) = ZERO
      DO 90 K = 1, NPNTS
      A(J, K) = A(J, K) - A(KCOL, K) * TE
90    CONTINUE
100   CONTINUE
110   CONTINUE
C
C
      DO 120 I = 1, NCPTS
      GAMMA(I) = A(I, NPNTS)
120   CONTINUE
      WRITE  (6,150)
      WRITE  (6,160)  HD
      WRITE  (6,170) (J, GAMMA(J), J = 1, NVORT)
      IF  (NSOR .EQ. 0) GO  TO 130
      WRITE  (6,180)
      WRITE  (6,170)  ( J, GAMMA(J), J = NSOR1, NCPTS )
130   CONTINUE
      RETURN
C
140   FORMAT  (1H0, 62H*** IN SUBROUTINE SLNR SELECTED PIVOT ELEMENT IS
     1TOO SMALL *** )
150   FORMAT  (1H1, 46X, 18HGAMMA DISTRIBUTION )
160   FORMAT  (1H0, 11A10)
170   FORMAT  (5(2X, I3, 1H., F15.5, 3X))
180   FORMAT  (1H0, 45X, 19HSOURCE DISTRIBUTION)
      END
```

```
      SUBROUTINE  SLPCLR (SLP, S, T)                                    XOVM0010
C  **                                                                   XOVM0011
C  **              SUBFOUTINE SLPCLR CALCULATES AN ANGLE BY THE         XOVM0012
C  **         ARCTANGENT FUNCTION                                       XOVM0013
C  **                                                                   XOVM0014
      LOGICAL SFLG, TFLG                                                XOVM0020
      SFLG = S.EQ.0.                                                    XOVM0030
      TFLG = T.FQ.0.                                                    XOVM0040
      IF  (SFLG .AND. TFLG) WRITE  (6,30)                               XOVM0050
      IF  (SFLG) GO  TO 10                                              XOVM0060
      IF  (TFLG) GO  TO 20                                              XOVM0070
      SLP = ATAN(T/S) * 57.29578                                        XOVM0080
      RETURN                                                            XOVM0090
10    CONTINUE                                                          XOVM0100
      SLP = 90.                                                         XOVM0110
      IF  (T.LT.0.) SLP = - SLP                                         XOVM0120
      RETURN                                                            XOVM0130
20    CONTINUE                                                          XOVM0140
      SLP = 0.                                                          XOVM0150
      RETURN                                                            XOVM0160
C                                                                       XOVM0170
30    FORMAT  (1H0, 47H*** IN SUBROUTINE SLPCLR S AND T EQUAL ZERO ***) XOVM0180
      END                                                               XOVM0190
```

```
      SUBROUTINE  STLNS
C  **
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)
C  **
C  **                 SUBROUTINE 'STLNS' CALCULATES
C  **         FOR THE GIVEN AIRFOIL SYSTEM.  IF THE ITERATIONS REQUIRED
C  **         FOR A GIVEN STREAMLINE EXCEED 'CRT' BEFORE THE CUTOFF
C  **         VALUE FOR 'X' IS ACHIEVED, THE NEXT STREAMLINE IS TO BE
C  **         ATTEMPTED...
C  **
      DO 10 J = 1, NSTLNS
      XA = XS(J)
      YA = YS(J)
      WRITE  (6,20)  J
      CALL  STREAM (XSUBC,DLTLMN,DLTALN,DLTALM,TOLLMT,CRT,XA,YA)
10    CONTINUE
      RETURN
C
20    FORMAT  (1H1, 8X, 10HSTREAMLINE , I2 )
      END
```

```
      SUBROUTINE  STREAM (XSUBC,DLTLMN,SAVE   ,DLTALM,TOLLMT,CRT,XA,YA)  XOVQ0010
C   **                                                                  XOVQ0011
C   **            SUBROUTINE STREAM CALCULATES THE STREAMLINES FOR THE  XOVQ0012
C   **         AIRFOIL SYSTEM                                           XOVQ0013
C   **                                                                  XOVQ0014
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,DUM(7)       XOVQ0020
C   **                                                                  XOVQ0030
      LOGICAL BLAP                                                      XOVQ0040
      DIMENSION HD (12),XX(5,1),YY(5,1),ERR1(5),ERR2(5),SLP(5),CP(5)    XOVQ0050
     1     , PSI(5)                                                     XOVQ0060
      DATA HD / 10H          , 10H X        , 10H   Y                   XOVQ0070
     1    ,    10H     X    , 10H       Y  , 10H          X             XOVQ0080
     2    ,    10H          , 10HY         , 10H    X                   XOVQ0090
     3    ,    10H     Y    , 10H        X , 10H          Y  /          XOVQ0100
      DATA BLANK / 1H /, ERR1 / 1H1, 1H2, 1H3, 1H4, 1H5 /               XOVQ0110
      IK = 0                                                            XOVQ0120
      BLAP = .TRUE.                                                     XOVQ0130
      IL = 0                                                            XOVQ0140
      DLTALN = SAVE                                                     XOVQ0150
      WRITE  (6,80)  HD                                                 XOVQ0160
      CALL  VELOC (UA,VA,XA,YA,DA,-1)                                   XOVQ0170
10    CONTINUE                                                          XOVQ0180
      SNA = VA/DA                                                       XOVQ0190
      CSA = UA/DA                                                       XOVQ0200
20    CONTINUE                                                          XOVQ0210
      VL1 = DLTALN * CSA                                                XOVQ0220
      VL2 = DLTALN * SNA                                                XOVQ0230
      XB = XA + VL1                                                     XOVQ0240
      YB = YA + VL2                                                     XOVQ0250
      XC = XB + VL1                                                     XOVQ0260
      YC = YB + VL2                                                     XOVQ0270
      CALL  VELOC (UB,VB,XB,YB,DB,-1)                                   XOVQ0280
      CT = DLTALN/DB                                                    XOVQ0290
      XD = XB + UB * CT                                                 XOVQ0300
      YD = YB + VB * CT                                                 XOVQ0310
      CALL  DSTSQD (XC, YC, XD, YD, XE, YE, S1, S2, 1)                  XOVQ0320
      IF  (S2.GE.TOLLMT) GO  TO 50                                      XOVQ0330
      IF  (BLAP) DLTALN = 1.25 * DLTALN                                 XOVQ0340
      BLAP = .TRUE.                                                     XOVQ0350
      IF  (DLTALN.GT.DLTALM) DLTALN = DLTALM                            XOVQ0360
30    CONTINUE                                                          XOVQ0370
      IK = IK + 1                                                       XOVQ0380
      IL = IL + 1                                                       XOVQ0390
      ERR2(IK) = BLANK                                                  XOVQ0400
      IF  (.NOT.BLAP) ERR2(IK) = ERR1(IK)                               XOVQ0410
      XX(IK, 1) = XB                                                    XOVQ0420
      YY(IK, 1) = YB                                                    XOVQ0430
      SLP(IK) = 57.29578*ATAN2(VB,UB)                                   XOVQ0440
      CP(IK) = 1. - (UB*UB + VB*VB)/USUBI/USUBI                         XOVQ0450
      CALL  FLOCAL  (PSI(IK), XB, YB)                                   XOVQ0460
      IF  (IK.NE.5) GO  TO 40                                           XOVQ0470
      IK = 0                                                            XOVQ0480
      WRITE  (6,90)  (XX(IA,1),YY(IA,1),IA=1,5),(SLP(IA),CP(IA),IA=1,5) XOVQ0490
      WRITE  (6,100)    ERR2                                            XOVQ0500
      WRITE  (6,110)     PSI                                            XOVQ0510
40    CONTINUE                                                          XOVQ0520
```

```
      FL = IL                                                           XOVQ0530
      IF ((FL.GT.CRT).OR.(DLTALN.GT.0..AND.XB.GT.XSUBC).OR.             XOVQ0540
     1    (DLTALN.LT.0..AND.XB.LT.XSUBC)) GO  TO 60                     XOVQ0550
      XA = XB                                                           XOVQ0560
      YA = YB                                                           XOVQ0570
      UA = UB                                                           XOVQ0580
      VA = VB                                                           XOVQ0590
      DA = DB                                                           XOVQ0600
      GO  TO 10                                                         XOVQ0610
50    CONTINUE                                                          XOVQ0620
      IF  (DLTALN.EQ.DLTLMN) GO  TO 30                                  XOVQ0630
      BLAP = .TRUE.                                                     XOVQ0640
      DLTALN = 0.75 * DLTALN                                            XOVQ0650
      IF  (DLTALN.GT.DLTLMN) GO  TO 20                                  XOVQ0660
      BLAP = .FALSE.                                                    XOVQ0670
      DLTALN = DLTLMN                                                   XOVQ0680
      GO  TO 20                                                         XOVQ0690
60    CONTINUE                                                          XOVQ0700
      IF  (IK.EQ.0) GO  TO 70                                           XOVQ0710
      WRITE  (6,90)  (XX(IA,1),YY(IA,1),IA=1,IK)                        XOVQ0720
      WRITE  (6,90)  (SLP(IA),CP(IA),IA=1,IK)                           XOVQ0730
      WRITE  (6,100)   (ERR2(IA), IA = 1, IK)                           XOVQ0740
      WRITE  (6,110)   (PSI(IA), IA = 1, IK)                            XOVQ0750
70    CONTINUE                                                          XOVQ0760
      RETURN                                                            XOVQ0770
C                                                                       XOVQ0780
80    FORMAT  (1H0, 12A10)                                              XOVQ0790
90    FORMAT  (4X, 5(4X, F9.4, 2X, F9.4) / 4X, 5(4X, F9.4, 2X, F9.4))   XOVQ0800
100   FORMAT  (1H+, 125X, 5A1)                                          XOVQ0810
110   FORMAT  (4X, 5(15X, F9.4) / )                                     XOVQ0820
      END                                                               XOVQ0830
```

```
      SUBROUTINE  VELOC (U0,V0,XBP,YBP,VM,IZ)                           XOVD0010
C  **                                                                   XOVD0011
C  **              SUBROUTINE VELOC  COMPUTES THE VELOCITIES INDUCED AT XOVD0012
C  **         A POINT BY SINGULARITIES AT THE CONTROL POINTS            XOVD0013
C  **                                                                   XOVD0014
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB                XOVD0020
      COMMON /BLKB/ BUM(6),USUBI,CDUM(8)                                XOVD0030
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)   XOVD0040
      COMMON /BLKD/ CS, SN, DUM(800)                                    XOVD0050
      COMMON /BLKE/ U(200), V(200), CPUP(200), CPLR(200), CPM(200)      XOVD0060
      COMMON /BLKG/ GAMMA(200)                                          XOVD0070
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT   XOVD0080
      U0 = USUBI * CS                                                   XOVD0110
      V0 = USUBI * SN                                                   XOVD0120
      IF  (IZ .LE. 0) GO  TO 100                                        XOVD0130
      DO 90 IY=1,NCPTS                                                  XOVD0140
      IYM1=IY-1                                                         XOVD0150
      IYP1=IY+1                                                         XOVD0160
      CALL  DSTSQD  (X(IY), Y(IY), XBP, YBP, X1, Y1, A, A1, 1)          XOVD0170
      IF ((IY.EQ.1).OR.(IY.EQ.NSOR1).OR.(IY.EQ.NCTS)) GO TO 10          XOVD0180
      IF ((IY.EQ.NCPTS).OR.(IY.EQ.NPTS).OR.(IY.EQ.NVORT)) GO TO 20      XOVD0190
      IF  (IZ.EQ.IY) GO  TO 30                                          XOVD0200
      IF  (IZ.EQ.IYM1) GO  TO 40                                        XOVD0210
C **  STANDARD POINT                                                    XOVD0220
      CALL  DSTRBD  (X(IY), Y(IY), X(IYP1), Y(IYP1), U1, V1, A, XBP, YBPXOVD0230
     1)                                                                 XOVD0240
      CALL  DSTRBD  (X(IY), Y(IY), X(IYM1), Y(IYM1), U2, V2, A, XBP, YBPXOVD0250
     1)                                                                 XOVD0260
      GO  TO 60                                                         XOVD0270
C  **                                                                   XOVD0280
10    CONTINUE                                                          XOVD0290
      U2 = 0.                                                           XOVD0300
      V2 = 0.                                                           XOVD0310
      IF  (IZ.EQ.IY) GO  TO 50                                          XOVD0320
      U1 = 0.                                                           XOVD0330
      V1 = 0.                                                           XOVD0340
      CALL  DSTRBD  (X(IY), Y(IY), X(IYP1), Y(IYP1), U2, V2, A, XBP, YBPXOVD0350
     1)                                                                 XOVD0360
      GO  TO 60                                                         XOVD0370
C  **                                                                   XOVD0380
20    CONTINUE                                                          XOVD0390
      U2 = 0.                                                           XOVD0400
      V2 = 0.                                                           XOVD0410
      IF  (IZ.EQ.IYM1) GO  TO 50                                        XOVD0420
      U1 = 0.                                                           XOVD0430
      V1 = 0.                                                           XOVD0440
      CALL  DSTRBD  (X(IY), Y(IY), X(IYM1), Y(IYM1), U2, V2, A, XBP, YBPXOVD0450
     1)                                                                 XOVD0460
      GO  TO 60                                                         XOVD0470
C  **                                                                   XOVD0480
30    CONTINUE                                                          XOVD0490
      CALL  DSTRBD  (X(IY), Y(IY), X(IYM1), Y(IYM1), U2, V2, A, XBP, YBPXOVD0500
     1)                                                                 XOVD0510
      GO  TO 50                                                         XOVD0520
C  **                                                                   XOVD0530
40    CONTINUE                                                          XOVD0540
```

```
      CALL  DSTRBD  (X(IY), Y(IY), X(IYP1), Y(IYP1), U2, V2, A, XBP, YBP
     1)
50    CONTINUE
      B1 = 6.283184 * A1
      U1 = Y1/B1
      V1 = - X1/B1
60    CONTINUE
      IF (IY.GE.NSOR1) GO TO 70
      UT = U1 + U2
      VT = V1 + V2
      GO  TO 80
70    CONTINUE
      UT = V1 + V2
      VT =-U1 - U2
80    CONTINUE
      U(IY) = UT
      V(IY) = VT
90    CONTINUE
      RETURN
C  **
100   CONTINUE
C  **
      CALL  DSTSQD  ( X(1     ), Y(1     ), XBP, YBP, X1, Y1, A, A1, 1)
      CALL  DSTRBD  (X(1), Y(1), X(2), Y(2), U2, V2, A, XBP, YBP)
      U0 = U0 + U2 * GAMMA(1)
      V0 = V0 + V2 * GAMMA(1)
C  **
      DO 110 IY = 2, NB
      CALL  DSTSQD  ( X(IY    ), Y(IY    ), XBP, YBP, X1, Y1, A, A1, 1)
      CALL  DSTRBD  (X(IY), Y(IY), X(IY+1), Y(IY+1), U1, V1, A, XBP, YBP
     1)
      CALL  DSTRBD  (X(IY), Y(IY), X(IY-1), Y(IY-1), U2, V2, A, XBP, YBP
     1)
      UT = U1 + U2
      VT = V1 + V2
      U0 = U0 + UT * GAMMA(IY)
      V0 = V0 + VT * GAMMA(IY)
110   CONTINUE
C  **
C           IY = NUMBER OF CORNER POINTS ON AIRFOIL = NPTS, CONTRIBUTION
C  **
      CALL  DSTSQD  ( X(NPTS ), Y(NPTS ), XBP, YBP, X1, Y1, A, A1, 1)
      CALL  DSTRBD  (X(NPTS), Y(NPTS), X(NB), Y(NB), U2, V2, A, XBP, YBP
     1)
      U0 = U0 + U2 * GAMMA(NPTS)
      V0 = V0 + V2 * GAMMA(NPTS)
C  **
C           IY = NCTS  TO NVORT
C  **
      IF  (NVOR  .EQ. 0) GO  TO 140
      IY = NCTS + 1
      CALL  DSTSQD  ( X(NCTS ), Y(NCTS ), XBP, YBP, X1, Y1, A, A1, 1)
      CALL  DSTRBD  (X(NCTS), Y(NCTS), X(IY), Y(IY), U2, V2, A, XBP, YBP
     1)
      U0 = U0 + U2 * GAMMA(NCTS)
      V0 = V0 + V2 * GAMMA(NCTS)
```

```
C   **
        IF  (IY .EQ. NSOR1) GO  TO 140
        NVORM1 = NVORT - 1
        IF  (NVOR .EQ. 2) GO  TO 130
C   **
        DO 120 I = IY, NVORM1
        CALL  DSTSQD  ( X(I      ), Y(I      ), XBP, YBP, X1, Y1, A, A1, 1)
        CALL  DSTRBD  (X(I), Y(I), X(I+1), Y(I+1), U1, V1, A, XBP, YBP)
        CALL  DSTRBD  (X(I), Y(I), X(I-1), Y(I-1), U2, V2, A, XBP, YBP)
        UT = U1 + U2
        VT = V1 + V2
        U0 = U0 + UT * GAMMA(I)
        V0 = V0 + VT * GAMMA(I)
120     CONTINUE
130     CONTINUE
        CALL  DSTSQD  ( X(NVORT), Y(NVORT), XBP, YBP, X1, Y1, A, A1, 1)
        CALL  DSTRBD  (X(NVORT), Y(NVORT), X(NVORM1),  Y(NVORM1),  U2, V2
       1        , A, XBP, YBP)
        U0 = U0 + U2 * GAMMA(NVORT)
        V0 = V0 + V2 * GAMMA(NVORT)
140     CONTINUE
C   **
C              IY = NSOR1 TO NCPTS
C   **
        IF  (NSOR .EQ. 0) GO  TO 170
        IY = NSOR1 + 1
        CALL  DSTSQD  ( X(NSOR1), Y(NSOR1), XBP, YBP, X1, Y1, A, A1, 1)
        CALL  DSTRBD  (X(NSOR1), Y(NSOR1), X(IY), Y(IY), U2, V2, A, XBP,YB
       1P)
        U0 = U0 + V2 * GAMMA(NSOR1)
        V0 = V0 - U2 * GAMMA(NSOR1)
C   **
        IF  (NSOR1 .EQ. NCPTS) GO  TO 170
        IF  (NSOR .EQ. 2) GO  TO 160
C   **
        DO 150 I = IY, IB
        CALL  DSTSQD  ( X(I      ), Y(I      ), XBP, YBP, X1, Y1, A, A1, 1)
        CALL  DSTRBD  (X(I), Y(I ), X(I+1), Y(I+1), U1, V1, A, XBP, YBP)
        CALL  DSTRBD  (X(I), Y(I ), X(I-1), Y(I-1), U2, V2, A, XBP, YBP)
        UT = V1 + V2
        VT = -U1 - U2
        U0 = U0 + UT * GAMMA(I)
        V0 = V0 + VT * GAMMA(I)
150     CONTINUE
160     CONTINUE
        CALL  DSTSQD  ( X(NCPTS), Y(NCPTS), XBP, YBP, X1, Y1, A, A1, 1)
        CALL  DSTRBD  (X(NCPTS), Y(NCPTS), X(IB), Y(IB), U2, V2, A, XBP,YB
       1P)
        U0 = U0 + V2 * GAMMA(NCPTS)
        V0 = V0 - U2 * GAMMA(NCPTS)
170     CONTINUE
        VM = SQRT (U0*U0 + V0*V0)
        RETURN
        END
```

```
      SUBROUTINE  WRTE                                                  XOVK0010
C   **                                                                  XOVK0011
C   **              SUBROUTINE WRTE WRITES THE INPUT DATA               XOVK0012
C   **                                                                  XOVK0013
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB                XOVK0020
      COMMON /BLKB/ XLE,YLE,XTE,YTE,XMBAR,YMBAR,USUBI,CREF,XSUBC,DLTLMN,XOVK0030
     1              DLTALN,DLTALM,TOLLMT,CRT,CRFSQ                      XOVK0040
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)   XOVK0050
      COMMON /BLKJ/ TITLE(12)                                           XOVK0060
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVCRT   XOVK0070
      COMMON /BLK2/ CPMOD(50)                                           XOVK0080
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE                              XOVK0090
      WRITE  (6,80)                                                     XOVK0100
      WRITE  (6,90)  TITLE                                              XOVK0110
      IF  (NMOD .GT. 0) WRITE  (6,100)  NMOD, MCPS, MCPF                XOVK0120
      IF  (NVLPTS.NE.0) WRITE  (6,110)  NVLPTS                          XOVK0130
      IF  (IND(5).NE.0) GO TO 10                                        XOVK0140
      WRITE  (6,120)                                                    XOVK0150
      GO TO 20                                                          XOVK0160
10    CONTINUE                                                          XOVK0170
      WRITE (6,130)                                                     XOVK0180
20    CONTINUE                                                          XOVK0190
      IF  (IND(6).EQ.2) GO TO 30                                        XOVK0200
      WRITE  (6,140)                                                    XOVK0210
      GO TO 40                                                          XOVK0220
30    CONTINUE                                                          XOVK0230
      WRITE  (6,150)                                                    XOVK0240
40    CONTINUE                                                          XOVK0250
      IF  (NSTLNS .EQ. 0) GO  TO 50                                     XOVK0260
      WRITE  (6,160)  NSTLNS                                            XOVK0270
      GO TO 60                                                          XOVK0280
50    CONTINUE                                                          XOVK0290
      WRITE  (6,170)                                                    XOVK0300
60    CONTINUE                                                          XOVK0310
      WRITE  (6,180)  ALPHA                                             XOVK0320
      WRITE  (6,190)  XMBAR,YMBAR                                       XOVK0330
      IF  (IND(4) .EQ. 0) WRITE  (6,200)  NVOR, NSOR                    XOVK0340
      WRITE  (6,210)  XSLE,YSLE,XSTE,YSTE,XLE,YLE,XTE,YTE,CREF          XOVK0350
      IF  (NSTLNS.EQ.0) GO TO 70                                        XOVK0360
      WRITE (6,220)  (I,XS(I),YS(I),I=1,NSTLNS)                         XOVK0370
70    CONTINUE                                                          XOVK0380
      IF  (NVLPTS.NE.0) WRITE  (6,230)  (I,XV(I),YV(I),I=1,NVLPTS)      XOVK0390
      IF  (IND(4) .EQ. 1) RETURN                                        XOVK0400
      WRITE (6,240)  (I,X(I),Y(I),I=NCTS,NVCRT)                         XOVK0410
      WRITE (6,250)  (I,X(I),Y(I),I=NSOR1,NCPTS)                        XOVK0420
      WRITE (6,260)  (I,X(I),Y(I),CPMOD(I-MCPS+1),I=MCPS,MCPF)          XOVK0430
      RETURN                                                            XOVK0440
C                                                                       XOVK0450
80    FORMAT  (1H1,24X,70HTWO-DIMENSIONAL SLAT DESIGN BY THE METHOD OF DXOVK0460
     1ISTRIBUTED SINGULARITIES//)                                       XOVK0470
90    FORMAT  (1X, 12A10 // )                                           XOVK0480
100   FORMAT  (16X,24H-- PRESSURE SPECIFIED AT,I3,29H POINTS, BEGINNING XOVK0490
     1WITH POINT,I4,18H ENDING WITH POINT,I4)                           XOVK0500
110   FORMAT  (16X,25H-- VELOCITY CALCULATED AT,I4,17H REMOTE POINTS...)XOVK0510
120   FORMAT  (16X,26H-- PRESSURES CALCULATED...)                       XOVK0520
130   FORMAT (16X,30H-- NO PRESSURE CALCULATIONS...)                    XOVK0530
```

```
140    FORMAT (16X,23H-- FORCES CALCULATED...)                          XOVK0540
150    FORMAT (16X,27H-- NO FORCE CALCULATIONS...)                      XOVK0550
160    FORMAT (16X, 2H--,I3,26H STREAMLINES CALCULATED...)              XOVK0560
170    FORMAT  (16X,29H-- NO STREAMLINE CALCULATIONS)                   XOVK0570
180    FORMAT (16X,32H-- ANGLE OF ATTACK OF SYSTEM IS ,F8.4,8H DEGREES) XOVK0580
190    FORMAT (16X,34H-- MOMENTS TAKEN ABOUT THE POINT (F8.4,1H,,F8.4,1H)XOVK0590
      1)                                                                XOVK0600
200    FORMAT (16X,47H-- SLAT VORTICITY DISTRIBUTION TO BE DEFINED AT,I3,XOVK0610
      1  9HPOINTS.../16X,44H-- SLAT SOURCE DISTRIBUTION TO BE DEFINED AT,XOVK0620
      2  I3,9HPOINTS...)                                                XOVK0630
210    FORMAT (16X,30H-- SLAT LEADING EDGE        X =,F8.4,5H  Y =,F8.4/ XOVK0640
      1          16X,30H-- SLAT TRAILING EDGE       X =,F8.4,5H  Y =,F8.4/ XOVK0650
      2          16X,30H-- AIRFOIL LEADING EDGE     X =,F8.4,5H  Y =,F8.4/ XOVK0660
      3          16X,30H-- AIRFOIL TRAILING EDGE    X =,F8.4,5H  Y =,F8.4/ XOVK0670
      4          16X,30H-- REFERENCE CHORD          C =,F8.4 )           XOVK0680
220    FORMAT (/16X,34H-- STREAMLINE STARTING COORDINATES/              XOVK0690
      1 (16X,3(I6, 8H.    X = F8.4,6H   Y = F8.4)))                     XOVK0700
230    FORMAT (/16X,27H-- REMOTE POINT COORDINATES/                     XOVK0710
      1 (16X,3(I6, 8H.    X = F8.4,6H   Y = F8.4)))                     XOVK0720
240    FORMAT (/16X,29H-- SLAT VORTEX CONTROL POINTS/                   XOVK0730
      1 (16X,3(I6, 8H.    X = F8.4,6H   Y = F8.4)))                     XOVK0740
250    FORMAT (/16X,29H-- SLAT SOURCE CONTROL POINTS/                   XOVK0750
      1 (16X,3(I6, 8H.    X = F8.4,6H   Y = F8.4)))                     XOVK0760
260    FORMAT (/16X,22H-- SPECIFIED PRESSURES/                          XOVK0770
      1 (16X,2(I6,8H.    X = F8.4,6H   Y = F8.4,7H  CP = F8.4)))        XOVK0780
       END                                                              XOVK0790
```

```
      PROGRAM XOVOO( INPUT=512, OUTPUT=512, TAPE5=INPUT, TAPE6=OUTPUT      XOVA001
     1        , TAPE1=2048, TAPE9=1024 )                                   XOVA002
C    **                                                                    XOVA0021
C    **                PROGRAM XOVOO DIRECTS THE CALLING OF THE OVERLAYS   XOVA0022
C    **                                                                    XOVA0023
      COMMON /BLKA/ IND(30),NCPTS,NSTLNS,NVLPTS,NPNTS,IB                   XOVA0030
      COMMON /BLKB/ XLE, YLE, XTE, YTE, XMBAR, YMBAR, USUBI, CREF, XSUBCXOVA0040
     1        , DLTLMN, DLTALN, DLTALM, TOLLMT, CRT, CRFSQ                 XOVA0050
      COMMON /BLKC/ ALPHA,XS(10),YS(10),XV(100),YV(100),X(200),Y(200)      XOVA0060
      COMMON /BLKD/ CS,SN,XBP(200),YBP(200),COSSLP(200),SINSLP(200)        XOVA0070
      COMMON /BLKE/ U(200), V(200), CPUP(200), CPLR(200), CPM(200)         XOVA0080
      COMMON /BLKG/ GAMMA(200)                                             XOVA0090
      COMMON /BLKJ/ TITLE(12)                                              XOVA0100
      COMMON /BLK1/ NVOR,NSOR,MCPS,MCPF,NMOD,NPTS,NB,NCTS,NSOR1,NVORT      XOVA0110
      COMMON /BLK3/ XSLE, YSLE, XSTE, YSTE                                 XOVA0120
C    **                TWO-DIMENSIONAL POTENTIAL FLOW ANALYSIS BY THE METHODXOVA0130
C    **         OF DISTRIBUTED SINGULARITIES...                            XOVA0140
110   CONTINUE                                                             XOVA0150
      REWIND  1                                                            XOVA0160
      CALL  OVERLAY  (3HDUM, 1, 0)                                         XOVA0170
      CALL  OVERLAY  (3HDUM, 1, 1)                                         XOVA0180
      CALL  OVERLAY (3HDUM, 2, 0)                                          XOVA0190
      CALL  OVERLAY (3HDUM, 1, 0)                                          XOVA0200
      CALL  OVERLAY  (3HDUM, 1, 2)                                         XOVA0210
      GO  TO 110                                                           XOVA0220
      END                                                                  XOVA0230
```

```
      PROGRAM XOV10                                                     XOVC001
C   **                                                                  XOVC0011
C   **                PROGRAM XOV10 PERMITS SUBROUTINES TO BE IN        XOVC0012
C   **          OVERLAY (DUM, 1, 0)                                     XOVC0013
C   **                                                                  XOVC0014
      G = 0.0                                                           XOVC0020
      END                                                               XOVC0030
```

```
      PROGRAM XOVI1                                                       XOVI0001
C  **                                                                     XOVI0011
C  **              PROGRAM XOVI1 DIRECTS THE CALLING OF SUBROUTINES       XOVI0012
C  **          RD, GEOM, MTRXGN                                           XOVI0013
C  **                                                                     XOVI0014
      CALL  RD                                                            XOVI0020
      CALL  GEOM                                                          XOVI0030
      CALL  MTRXGN                                                        XOVI0040
      END                                                                 XOVI0050
```

```
      PROGRAM XOV12                                                     XOV0001
C   **                                                                  XOV0002
C   **              PROGRAM XCV12 DIRECTS THE CALLING CF SUBROUTINES    XOV0003
C   **         STLNS, RMTVTY, SHAPE, PRSS, FRCE                         XOV0004
C   **                                                                  XOV0005
      COMMON /BLKA / IND(30), NCPTS, NSTLNS, NVLPTS, NPNTS, IB          XOV00060
      IF  (NSTLNS .GT. 0) CALL  STLNS                                   XOV00070
      IF  (NVLPTS .NE. 0) CALL  RMTVTY                                  XOV00080
      IF ((IND(2) .EQ. 0) .AND. (IND(4) .EQ. 0)) CALL  SHAPE            XOV00090
      IF  (IND(5) .EQ. 0) CALL  PRSS                                    XOV00100
      IF  (IND(6) .NE. 2) CALL  FRCE                                    XOV00110
      END                                                               XOV00120
```

```
      PROGRAM XCV20                                                     XOVY001
C  **                                                                   XOVY0011
C  **             PROGRAM XCV20 DIRECTS THE CALLING OF SUBROUTINE SLNR  XOVY0012
C  **                                                                   XOVY0013
      CALL  SLNR                                                        XOVY0020
      END                                                               XOVY0030
```

REFERENCES

1. Peebles, G. H., "A Method for Calculating Airfoil Sections from Specifications on the Pressure Distributions," Journal of the Aeronautical Sciences, Vol. 14, No. 8 (August 1947), 451-456.

2. Eppler, R., "Direct Calculation of Wing Profiles from the Pressure Distribution (Direkte Berechnung von Tragflugel profilen aus der Druck verteilung)," Ingenieur Archiv, Vol. 25 (1957), 32-59.

3. Liebeck, R. H., and Ormsbee, A. I., "Optimization of Airfoils for Maximum Lift," AIAA Paper No. 69-739, July 1969.

4. Lighthill, M. J., A New Method of Two-Dimensional Aerodynamic Design, Aeronautical Research Council R&M No. 2112, 1945.

5. Chen, A. W., The Determination of the Geometries of Multiple-Element Airfoils Optimized for Maximum Lift Coefficient, Thesis, University of Illinois at Champaign-Urbana, 1971.

6. O'Pray, J. E., A Semi-Inverse Design Technique for Leading-Edge Slats, Thesis, California Institute of Technology, 1970.

7. Bhateley, I. C., and McWhirter, J. W., Development of Theoretical Method for Two-Dimensional Multi-element Airfoil Analysis and Design, Part I: Viscous Flow Analysis Method, AFFDL-TR-72-96, August 1972.

8. Hess, J. L., and Smith, A. M. O., Calculation of Non-lifting Potential Flow about Arbitrary Three-dimensional Bodies, McDonnell-Douglas Report No. E.S. 40622, 15 March 1962.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Convair Aerospace Division General Dynamics Corporation Fort Worth, Texas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
DEVELOPMENT OF THEORETICAL METHOD FOR TWO-DIMENSIONAL MULTI-ELEMENT AIRFOIL ANALYSIS AND DESIGN
Part II: Leading-Edge Slat Design Method

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Report — 24 May 1971 through 12 June 1972

5. AUTHOR(S) *(First name, middle initial, last name)*
O. Wayne McGregor
Jack W. McWhirter

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1972	172	8

8a. CONTRACT OR GRANT NO. F33615-71-C-1597 b. PROJECT NO. 1366 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFFDL-TR-72-96, Part II 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	AFFDL/FXM

13. ABSTRACT

A method has been developed for the design of leading-edge slats that produce a specified pressure distribution on the main airfoil. The method of distributed singularities is applied in a unique manner. The airfoil is represented in the conventional manner by a vortex sheet having the same shape as the airfoil. The slat is represented by a vortex sheet and a source line. The source line provides the slat thickness; the vortex sheet provides the camber. A closed slat shape is guaranteed by requiring that the net mass added to the system be zero and that the stream function at the slat leading-edge stagnation point have the same value as at the trailing edge. It was found that valid solutions are possible only when the source line at least approximates a streamline generated by the airfoil without the slat. The slat shape is computed by locating the body streamline of the slat. A constrained least-square analysis provides this definition. Several sample designs are discussed. Detailed instructions for application of the method are provided. Details regarding the associated computer program are included.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
High-lift devices						
Inverse design						
Leading-edge slats						
Method of distributed singularities						
Potential flow						

Unclassified
Security Classification